Multi-Objective System-Level Management of Modern Green Data Centers

Thèse N° 9457

Présentée le 23 septembre 2019

à la Faculté des sciences et techniques de l'ingénieur
Laboratoire des systèmes embarqués
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Ali PAHLEVAN

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Prof. D. Atienza Alonso, directeur de thèse
Prof. T. Š. Rosing, rapporteuse
Prof. E. Macii, rapporteur
Prof. A. P. Burg, rapporteur

2019

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

The only true wisdom is to know that
you know nothing.
— Socrates

To my grandparents.
To my parents.
To my family...

Acknowledgements

I would not have been able to carry out the work presented in this thesis without the excellent guidance and support given by my advisor, *Prof. David Atienza*. Back in 2014, I arrived to EPFL as a trainee, and after six months he gave me an opportunity to join his group as a PhD student. During all this time, he has made me understand what research is, how to collaborate in a team and why critical thinking, patience and hard work are the foundations to do a good job. His enthusiasm, kindness, commitment, and faith in my abilities have been extremely helpful and motivating to carry out my research work under the best possible conditions. For all of these reasons, I am deeply grateful to him. The following paragraph, I will express my sincerest gratitude to those who helped me to produce this manuscript.

First of all, I greatly thank my thesis jury members *Prof. Pascal Frossard* (jury president), *Prof. Tajana Simunic Rsoing* (external examiner), *Prof. Enrico Macii* (external examiner) and *Prof. Andreas Burg* (internal examiner), for taking their valuable time out of their busy schedule to review this manuscript and providing me with their insightful comments. Furthermore, I would like to express my heartfelt thanks to *Prof. Luca Benini, Prof. Babak Falsafi, Prof. Ayse K. Coskun, Prof. Davide Brunelli, Andrea, Davide, Javier, Arash*, and *Maurizio* for their priceless guidance throughout our fruitful collaboration. Then, I am grateful to the postdoctoral researchers *Dr. Marina Zapater* and *Dr. Pablo Garcia del Valle*, who co-advised me, carefully oversaw my work and helped me face the avalanche of difficulties encountered during the elaboration of our various publications.

Additionally, I would like to seize this opportunity to acknowledge all my colleagues at ESL, with whom I interacted on a daily basis, making memorable moments, from time-to-time coffee breaks and birthday parties, to all exciting ESL outdoors activities. Especially, I would like to thank *Homeira* and *Francine*, our secretaries, for organizing lab events and my trips to different conferences in many places; *Rodolphe*, our IT technician, for all his beyond-the-imagination supports and effors in all IT-related issues; *Martino, Jungsoo, Mohamed, Hossein, Karim, Amir, Miguel, Ruben, Alexandre, Tomas,* and *Adriana,* our former and current post-doctoral fellows, with whom I worked indirectly during my PhD; *Artem* and *Halima* for the amazing moments we shared in our office; *Soumya* and *Loris*; and *Yasir, Kawsar,* and *Xiaou*, not only as my good friends, but also because of our productive joint research. More particularly, I would like to express my especial gratitude to *Dr. Amir Aminifar*, for being such a great friend and all his insightful comments that led me enhance my research quality; and *Arman* who was not only a great colleague and friend, but a wonderful collaborator and travel companion. Also, I would like to thank all other ESL members, who made such a friendly

and peaceful environment during the past years: *Farnaz, Fabio, Eli, Benoit, William, Lara, Szabolcs, Wellington, Damián , Renato, Luis, Gregoire,* and *Dionisije*. In addition, during these years of living in Switzerland, there were many great Iranian friends who made life with all its ups-and-downs sweet and enjoyable for me. Hereby, I would like to thank them all (*Vahid, Peyman, Reza, Hossein, Hesam, Farnood, Morteza, Amir, Ehsan, Abolfazl, Mohammad, Ashkan, Hassan, Saleh, Ahmad, Omid, Nastaran, Farzaneh, Fatemeh, Zhaleh, Aida,* and *Farnaz*). I would not like to finish these acknowledgements without mentioning the best friends that I met during my life, particularly *Mahdi, Majid, Reza, Navid,* and *Masood*.
The last but here the most, I would like express my heartfelt gratitude to my entire family, and particularly, my grandparents, parents, and sister, without whom it was definitely impossible.

Lausanne, 2 April 2019 Ali Pahlevan.

Abstract

In our modern society, the average citizen has turned into a daily cloud user. Despite being virtually transparent for the user, internet services such as web search, e-mail, video streaming, data analytics, etc., require a heavy infrastructure behind the scenes. Data centers with thousands of servers and communication equipment burn several megawatts of power to serve users on a 24/7 basis, and their electricity bill is a major fraction of their costs, reaching 1,120 GWh and $67 million annually for major players such as Google. Hence, modern cloud data centers need to tackle efficiently the increasing demand for computing resources while at the same time addressing the energy efficiency challenge. This is a complex optimization problem that worsens as more constraints and objectives are added, especially since power and guaranteed Quality-of-Service (QoS) (i.e., response time or throughput) indicate different directions for optimization. In public clouds, such as Amazon Web Services or Google Cloud Platform, virtualization transforms a data center into a flexible cloud infrastructure in which Virtual Machines (VMs) behave as separate entities that share physical hardware resources among each other. Therefore, it is essential to develop resource provisioning policies that are aware of VMs characteristics (e.g., CPU utilization and data communication) and applicable in dynamic scenarios. Due to the size of the problem, state-of-the-art techniques for VM allocation in data centers only consider a subset of the VMs characteristics, depending on the metrics they want to optimize; thus yielding sub-optimal solutions for multi-objective optimizations.

To address the previous challenges, this thesis first presents heuristic and Machine Learning (ML)-based VM allocation methods and assesses them in terms of energy, QoS, network traffic, number of migrations, and scalability for various data center scenarios. Then, a novel hyper-heuristic algorithm is proposed that exploits the benefits of both methods by dynamically finding the best one according to a user-defined metric. For optimality assessment, I formulate an Integer Linear Programming (ILP)-based VM allocation method to minimize energy consumption and data communication, which obtains optimal results, but is impractical at run-time. The results demonstrate that the ML approach provides up to 24% server-to-server network traffic improvement and reduces execution time by up to 480x for large-scale scenarios. On the contrary, the heuristic outperforms the ML method in terms of energy and network traffic for reduced scenarios. I also show that the heuristic and ML approaches have up to 6% energy consumption overhead compared to ILP-based optimal solution.

However, optimizing the energy and cost of a single data center powered by the grid is not enough in today's cloud computing context, where multiple data centers, built in different ge-

ographical locations, are used to deploy online services, and use renewable energy sources to reduce their carbon footprint. For instance, major players such as Yahoo currently uses 56.4% green energy to power its data centers. This thesis also presents a two-phase multi-objective VM placement, clustering and allocation algorithm, along with a dynamic migration technique, for geo-distributed data centers coupled with renewable and Electrical Energy Storage (EES) sources. The proposed technique exploits the holistic knowledge of VMs characteristics to tackle the challenges of operational cost (i.e., electricity bill) optimization and energy-performance trade-offs. Experimental results show that the proposed method provides up to 54% operational cost savings, 14% energy consumption, and 10% performance (response time) improvements compared to state-of-the-art schemes. Furthermore, in order to efficiently minimize cost, power market operators have recently introduced emerging demand-response programs, in which electricity consumers regulate their power usage following providers' requests. Among different programs, Regulation Service (RS) reserves are particularly promising for data centers due to the high credit gain possibilities and data centers' flexibility in regulating their power consumption. Therefore, it is essential to develop bidding strategies for data centers to participate in emerging power markets together with power management policies that are aware of power market requirements at run-time. In this thesis I also propose a holistic strategy to jointly optimize the data center RS provision problem and VM allocation that satisfies the hour-ahead power market constraints in the presence of renewable and EES energy. The results show up to 71%, 48%, and 28% monetary cost, renewable, and EES utilization improvements, respectively, compared to other approaches.
Even if novel data center resource management techniques can efficiently tackle the dramatic increase in the number of servers, each computing server remains power limited due to effect of post-Dennard scaling. Therefore, techniques such as Near-Threshold Computing (NTC) need to complement novel system-level approaches to improve data centers' energy efficiency. NTC increases energy efficiency by lowering the operating voltage to a value slightly higher than the transistor threshold. For this purpose, I first use an accurate power modeling characterization for a new server architecture based on the Fully Depleted Silicon On Insulator (FD-SOI) process technology that enables NTC features. Then, I explore the new energy-performance trade-offs brought by next-generation NTC-based data centers when executing virtualized applications with different CPU utilization and memory footprint characteristics. Finally, based on this analysis, I propose a novel energy proportionality-aware dynamic VM allocation method at data center level that exploits the knowledge of VMs characteristics together with the accurate power model presented for NTC servers. As a result, the proposed approach increases the energy proportionality of NTC-based data centers, providing up to 45% energy savings compared to the latest consolidation techniques, while guaranteeing QoS requirements.

Keywords: *Geo-distributed data centers, cloud computing, energy efficiency, heuristic, machine learning (ML), hyper-heuristic, renewable energies, electrical energy storage (EES), operational costs, network traffic, quality-of-service (QoS), scalability, virtual machines (VMs), near-threshold computing (NTC), fully depleted silicon on insulator (FD-SOI)*

Résumé

Dans notre société moderne, le citoyen moyen est un utilisateur quotidien du Cloud. Bien que ce soit transparent à l'utilisateur, les services Internet tels que la recherche Web, la messagerie électronique ou le multimédia en continu nécessitent une lourde infrastructure. Les centres de données avec des milliers de serveurs et d'équipements de communication consomment plusieurs mégawatts pour desservir les utilisateurs en continu. Leur facture d'électricité représente une fraction importante de leurs dépenses, atteignant 1120 GWh et 67 millions de dollars par an pour des géants comme Google. Ainsi, les centres de données modernes doivent faire face à la demande croissante en ressources tout en visant l'efficacité énergétique. Ce problème complexe d'optimisation grandit avec l'ajout de nouvelles contraintes et objectifs, d'autant plus que la puissance et la qualité de service nécessitent des optimisations différentes. Dans les Clouds publics comme Amazon Web Services ou Google Cloud Platform, la virtualisation transforme le centre de données en une infrastructure flexible dans laquelle les machines virtuelles (VMs) sont des entités distinctes qui partagent les ressources matérielles. Il est donc nécessaire de développer des stratégies d'approvisionnement en ressources dynamiquement applicables, tenant compte des caractéristiques des VMs (utilisation des processeurs ou communication de données). En raison de la taille du problème, les techniques actuelles d'allocation de VMs dans les centres de données ne prennent en compte qu'une partie des caractéristiques des VMs dépendamment des métriques qu'ils souhaitent optimiser, résultant ainsi à des solutions sous-optimales.

Pour adresser les défis précédents, cette thèse présente tout d'abord des méthodes heuristiques d'allocation de VMs ainsi que des méthodes basées sur l'apprentissage automatique (ML), et les évalue en termes d'énergie, de qualité de service, de trafic de réseau, de migrations et d'évolutivité, et ce pour différents scénarios. Ensuite, un nouvel algorithme hyper-heuristique est proposé. Il exploite les avantages des deux méthodes en recherchant dynamiquement la meilleure, selon une métrique définie. Pour évaluer l'optimalité de l'algorithme, j'ai formulé une méthode d'allocation de VMs basée sur la programmation linéaire en nombres entiers (ILP). Elle minimise la consommation d'énergie et la communication de données et permet des résultats optimaux. Cependant, elle n'est pas pratique pendant l'exécution. Les résultats démontrent que l'approche ML améliore le trafic et réduit le temps d'exécution pour des scénarios à grande échelle. Cependant, La méthode heuristique surpasse la méthode ML pour les scénarios réduits.

Cependant, l'optimisation de l'énergie et du coût d'un seul centre de données n'est pas suffisante dans le contexte de l'informatique en Cloud, où plusieurs centres de données dans

différentes locations géographiques sont utilisés pour déployer des services en ligne, et utilisent des sources d'énergie renouvelables pour réduire leur empreinte carbone. Par exemple, un géant tel que Yahoo utilise 56,4% d'énergie verte pour alimenter ses centres de données. Cette thèse présente un algorithme en deux phases et multi-objectifs de placement, regroupement et allocation de VMs, ainsi qu'une technique de migration dynamique pour les centres de données distribués, couplés à des sources d'énergie renouvelable et des sources de stockage d'énergie électrique (EES). La technique proposée exploite la connaissance globale des caractéristiques des VMs pour optimiser les coûts opérationnels et la performance énergétique. En outre, afin de réduire efficacement leurs coûts, les opérateurs du marché de l'énergie ont récemment introduit de nouveaux programmes de réponse à la demande, dans lesquels les utilisateurs régulent leur consommation selon la demande des fournisseurs. Parmi les différents programmes, les réserves du service de régulation (RS) sont particulièrement prometteuses pour les centres de données en raison des possibilités de gain de crédit élevées et la flexibilité de régulation de leur consommation d'énergie. Il est donc essentiel de développer des stratégies d'appel d'offres permettant aux centres de données de participer aux marchés émergeants, ainsi que des politiques de gestion qui tiennent compte des besoins en temps d'exécution. Dans cette thèse, je propose une stratégie globale de co-optimisation du problème d'approvisionnement RS et d'allocation de VMs qui satisfasse les contraintes d'une heure à l'avance, en présence d'énergie renouvelable et EES. Les résultats montrent des améliorations jusqu'à 71%, 48% et 28% des coûts, de l'utilisation d'énergies renouvelables et de la SEE respectivement.
Même si les nouvelles techniques de gestion de ressources des centres de données font face efficacement à l'augmentation importante du nombre de serveurs, la puissance de chaque serveur est limitée à cause des effets de la réduction des dimensions post-Dennard. Par conséquent, des techniques comme l'électronique proche du seuil (NTC) doivent complémenter les approches d'amélioration de l'efficacité énergétique globale des centres de données. Ainsi, j'utilise d'abord une caractérisation précise de la consommation électrique de nouveaux serveurs avec la technologie du FD-SOI permettant le fonctionnement NTC. Ensuite, j'explore de nouveaux compromis entre performance et consommation énergétique des centres de données de nouvelle génération, lors de l'exécution d'applications virtualisées avec différentes caractéristiques d'utilisation du processeur et de la mémoire. Enfin, je propose une nouvelle méthode d'allocation dynamique de VMs qui exploite la connaissance des caractéristiques des VMs ainsi que le modèle de puissance précis des serveurs NTC. Cette approche augmente la proportionnalité énergétique des centres de données basés sur le NTC, permettant des économies d'énergie allant jusqu'à 45% par rapport aux dernières techniques de consolidation, tout en garantissant la qualité de service exigée.

Mots clés : *Centres de données géo-distribués, informatique en nuage, efficacité énergétique, heuristique, apprentissage automatique (ML), hyper-heuristique, énergies renouvelables, stockage de l'énergie électrique (EES), coûts opérationnels, trafic réseau, qualité de service, qualité évolutive , machines virtuelles (VM), électronique proche du seuil (NTC), silicium sur isolant totalement déserté (FD-SOI)*

Contents

List of Figures

List of Tables

Acronyms

AC Alternating Current

ARIMA Autoregressive Integrated Moving Average

BER Bit Error Rate

BFD Best-Fit-Decreasing

CRAC Computer Room Air Conditioning

CTI Charge Transfer Interconnect

DAG Directed Acyclic Graph

DC Direct Current

DP Dynamic Programming

DSO Distribution System Operator

DVFS Dynamic Voltage and Frequency Scaling

DoD Depth-of-Discharge

EES Electrical Energy Storage

FBB Forward Body Biasing

FD-SOI Fully Depleted Silicon On Insulator

FinFET Fin Field-Effect Transistor

HES Hybrid Electric Systems

Acronyms

HPC High Performance Computing

ILP Integer Linear Programming

IoT Internet of Things

ISA Instruction Set Architecture

ISO Independent System Operator

IT Information Technology

LLC Last-Level Cache

MAPE Mean Average Percentage Error

MILP Mixed Integer Linear Programming

ML Machine Learning

MPPT Maximum Power Point Tracking

NOCT Nominal Operating Cell Temperature

NTC Near-Threshold Computing

OoO Out-of-Order

PCP Peak Clustering-based Placement

PDU Power Distribution Unit

PUE Power Usage Effectiveness

PV Photovoltaic

QoS Quality-of-Service

RAPL Running Average Power Limit

RBB Reverse Body Biasing

RL Reinforcement Learning

RS Regulation Service

SLA Service-Level Agreement

STC Standard Test Conditions

SoC State-of-Charge

SoH State-of-Health

TDP Thermal Design Power

ToR Top-of-Rack

UIPC User Instructions Per Cycle

UIPS User Instructions Per Second

UPS Uninterruptible Power Supply

UTBB Ultra-Thin Body and Buried Oxide

VDC Virtual Data Center

VM Virtual Machine

WFM Wait-For-Memory

1 Introduction

1.1 Fundamentals and Trends of Data Centers

Cloud computing has recently been brought into focus in both academia and industry due to the increase of applications and services, such as the web search, mail, video, apps, storage, etc. that require a heavy infrastructure for processing, like data centers with thousands of servers and communication equipment. Hence, ever-increasing demands for computing and growing number of clusters and servers in data centers have ramped up the power consumption costs as an undesirable effect [1]. Typically, large-scale data centers dissipate several megawatts of power and the corresponding annual electricity bills are in the order of tens of millions of dollars, such as Google with over 1,120 GWh and $67 million, and Microsoft with over 600 GWh and $36 million [2]. Reportedly, data centers electricity consumption will reach 8% of the worldwide electricity production by 2020 [3]. This implies a significant amount of carbon emission rate from fossil fuel combustion.

Traditional fossil fuel concerns, i.e., carbon emissions and global warming, impose the introduction of sustainable energy sources [4], since 10% of the world's electricity generation has been estimated to be consumed by Information Technology (IT) infrastructures [5]. In this context, the current trend from data center providers is to use green geo-distributed data centers, which are multiple data centers built in different geographical locations, connected through the network, and coupled with renewable energy sources [4], as shown in Fig. 1.1. By employing renewable energy sources (e.g., solar, wind, etc.), they reduce their carbon footprint (the need of electricity from the grid) and, by exploiting the geographical price diversity and the varying green energy supply in the different locations, the cost of energy can be minimized. According to Greenpeace's report [6], both Google (39.4% clean energy) and Yahoo (56.4% clean energy) are active in supporting policies to power data centers with green energy. In contrast, many large IT companies, such as Amazon, Apple, and Microsoft, rapidly expand their cloud business without adequate attention to the electricity source, and rely heavily on brown energy to power their clouds [7]. Even worse, there are numerous small- and medium-sized data centers that consume the majority of energy with less attention to energy efficiency.

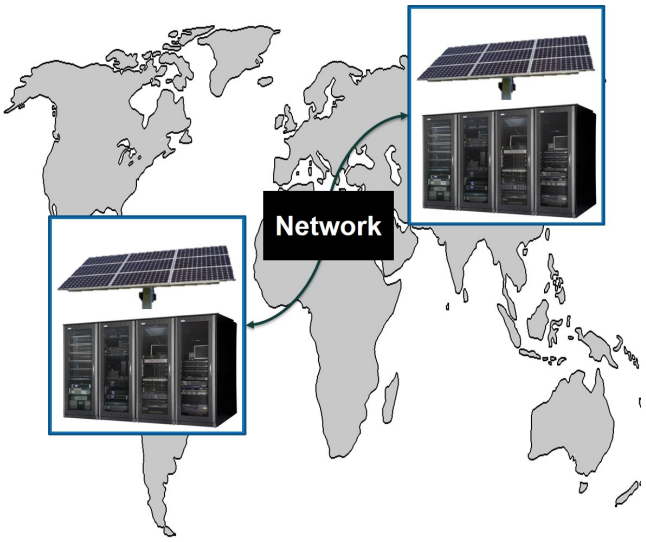

Figure 1.1 – Structure of the green geo-distributed data centers.

In order to efficiently manage the operation of a data center, advanced optimization techniques, at architecture and system levels, are required to minimize data center energy consumption and operational cost. At system-level management, the goal is to route workloads to locations with cheaper electricity price while maximizing green energy utilization. This is an NP-hard problem that is worsened as more constraints and objectives are added, since most of them indicate different directions for the optimization, e.g., power, performance, or a guaranteed Quality-of-Service (QoS) (e.g., response time or throughput).

1.1.1 Green Energy Sources - Renewable Energy and Electrical Energy Storage (EES) Systems

Large-scale data centers use renewable energy to reduce their dependency on costly and brown energy (fossil fuel) from the power grid [8]. In recent years, all the big energy consumers in the IT market (Google, Rackspace, etc.) have introduced renewable energy sources in their supply chain, locating their infrastructures in suitable geographical locations around the world. The penetration of renewable and green energy sources is almost non-existent for company-owned data centers, i.e., IT infrastructures located in the same corporate building where the business is run, mostly in urban environments.

Among the renewables, solar energy is the most effective renewable source employed in green data centers since Photovoltaic (PV) modules can be easily located close by the data center and the converted energy can be immediately used without distribution. Moreover, it is the most suitable for small- to medium-scale data centers (up to few hundreds of kWs of IT power) located in urban environments where wind turbines and water storage infrastructures cannot be built, given the space required for such infrastructures. Nevertheless, renewable energy sources are not constant over the time; their intensity depends on weather, geographical

position of the plant and seasons. Hence, a maximum in the energy intake rarely corresponds with a maximum in the demand. However, estimating their short-term trend (one day ahead) with small error (i.e., Mean Average Percentage Error (MAPE) close to 10%) is possible, as it has been demonstrated in previous research [9]. Similar results can be expected when dealing with electricity demand prediction at the building scale (few tens of kWs) [10].

To tackle the imbalance between energy intake (e.g., renewable energy) and demand, efficient techniques are required to optimize the usage of Electrical Energy Storage (EES), as an energy storage that collects the surplus of green energy for future needs. Therefore, the management of EES (i.e., real-time decision on charging/discharging of the battery) plays a major role in lifetime and operation of battery bank. To address this challenge, different battery technologies (i.e., Hybrid Electric Systems (HES)) can be used to compensate the drawbacks of each other (e.g., life cycles, cost, etc.).

1.1.2 Cloud Application Characteristics for Multi-Objective Data Center Optimization

To optimize the operation of a data center as well as real-time systems [11], it is crucial to minimize both IT and cooling energy consumptions. In public clouds, such as Amazon Web Services or Google Cloud Platform, virtualization transforms a data center into a flexible cloud infrastructure in which Virtual Machines (VMs) behave as separate entities (i.e., using isolated operating systems as shown in Fig. 1.2) that share physical hardware resources among each other. Existing algorithms typically consider a particular subset of VM characteristics that is relevant to optimize the main goal, and keep the rest of metrics either unchanged [12] or under a predefined threshold [13]. The process of placing a set of VMs on a server requires not only to check that the total size of VMs' load does not exceed the servers' capacity [14], but also to analyze other factors that influence how suitable they are for co-location. Regarding the VMs characteristics, data centers universally host heterogeneous applications with high variability in CPU utilization and heavy data communication between VMs (e.g., bank applications with transactions between any two customers). Both characteristics are

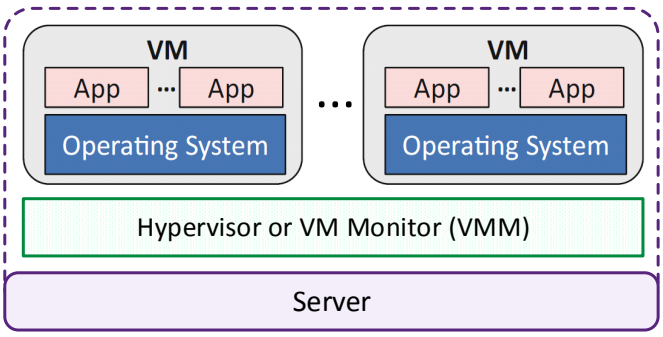

Figure 1.2 – Virtualized servers data centers including VMs and hypervisor.

important since high variability in the CPU utilization of VMs provides significant potential for power savings by efficiently consolidating VMs. On the other hand, the data communication requires to exchange data between VMs through the network, which directly impacts response time.

Based on the users' demands, very different computation and communication patterns can be found in virtualized data centers [15, 16]. For instance, scale-out applications [17] (e.g., web search, web serving, data analytics, etc.) exhibit very different characteristics compared to traditional High Performance Computing (HPC) services. They are user-interactive, sensitive to latency, and present higher variability in CPU utilization (fast-changing loads) due to their dependency on the number of clients/queries. Additionally, they also have high data communication. For example, a web search query requires parallel communication among VMs to return the most relevant results. In particular, in MapReduce, the output data of map phase should be transferred before proceeding to the reduce step [18]. Likewise, banking applications, Google Docs, and Microsoft Office Online, as well as social networking applications such as Facebook, LinkedIn, and Twitter also require data communication among the VMs [19]. To this end, two key factors to consider as workload characteristics are CPU-load and data correlation. CPU-load correlation indicates whether their CPU utilization peaks coincide during a certain time interval [20], as shown in Fig. 1.3. Data correlation refers to the dependency between each two VMs due to the amount of data that they need to exchange [21], as shown in Fig. 1.4.

From the CPU-load correlation side, studies [20, 22] design energy-efficient algorithms that separate CPU-load-correlated VMs in order to minimize energy consumption. The prior study [20] demonstrates that having detailed information about the characteristics of the running applications, as opposed to using stationary CPU-load values for VMs (e.g. peak or average values), gives the opportunity to further reduce the power consumption. On the other hand, data transfer among VMs (i.e., data correlation) is an important aspect severely missing from many previous studies. Communication among VMs has already been taken

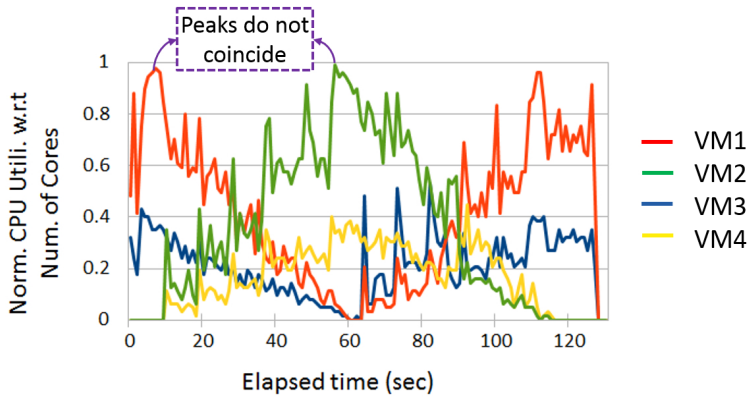

Figure 1.3 – CPU-load correlation - less correlation when the peaks do not coincide.

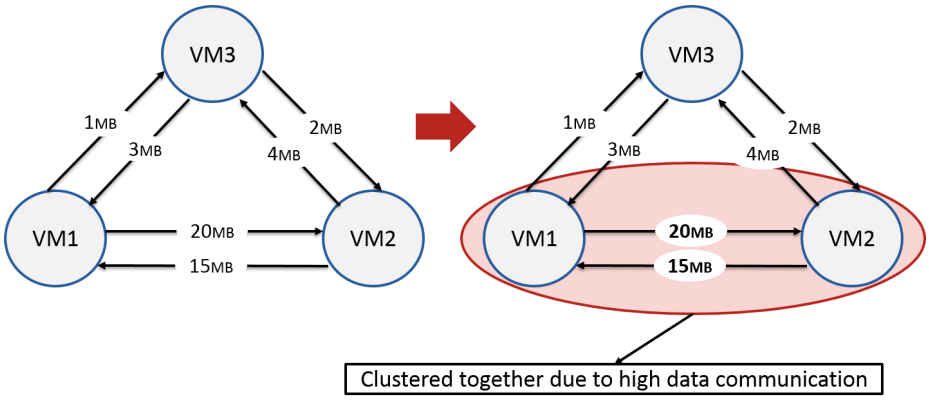

Figure 1.4 – Data correlation - amount of data exchanged among the VMs.

into account in several works [19, 21, 23] to minimize network traffic and response time. However, the communication they consider is only unidirectional while, in practice, two VMs regularly exchange different amounts of data in both directions (bidirectional data correlation), and these amounts change at run-time depending on real-time information. In this case, one of the data directions between each two VMs can lead to network congestion and the increase of response time especially when two VMs are in different data centers. Therefore, these correlation constraints indicate opposed goals, as highly CPU-load-correlated VMs should be placed apart, while highly data-correlated VMs should be clustered together. It is, thus, challenging for data center providers to conduct an efficient management taking into consideration the trade-offs between energy and performance (response time) in the presence of CPU-load and data correlations, as studied in a recent survey [24]. Especially in modern data centers, these two correlations directly impact the main provider objectives including operational costs, data center energy consumption, renewable usage, battery utilization, network traffic and response time.

In addition to a high-variability in the CPU utilization traces of applications, a daily periodicity is also observed, as shown in Fig. 1.5. This provides great opportunities to accurately predict the workloads, which increases the workload management efficiency for a time-ahead decision.

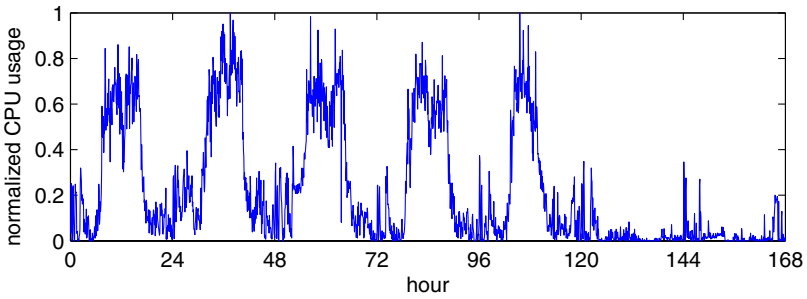

Figure 1.5 – One week workload traces with a daily pattern [25].

Basically, quality degradation is observed due to the miss-predictions, especially during abrupt workload changes. In this thesis, I target banking applications which are virtualized batch jobs and defined in terms of resource requirements, arrival and total time (life time). A batch job can run at anytime on different servers with relaxed QoS constraints.

1.2 State-of-the-art on Workload Allocation Techniques

In recent years, several techniques have been proposed on allocation techniques in single and geo-distributed data centers based on different objectives. I classified previous studies and adopted methods into two main topics: i) single data center, and ii) geo-distributed data center approaches. I also provide the resulting challenges raised by these optimizations in each category.

1.2.1 Single Data Centers and Challenges

Energy-Aware VM Allocation and Provisioning: Among all energy reduction techniques, VM consolidation [12] is one of the most widely used methods, as it packs VMs into the minimal number of active servers [14], and consolidating multiple turned on servers into the minimal number of racks [26]. For this purpose, techniques for dynamic consolidation of VMs by means of migration have been presented which minimizes the power consumption and ensures Service-Level Agreements (SLAs) [27]. A common approach for migration is to use adaptive utilization thresholds, setting a lower and upper bounds for the server's utilization. For instance, when the server utilization becomes either less or more than the thresholds, it is decided to migrate the server's workloads. For performing the migration actions, there are short-term (i.e., every few seconds) and long-term decisions process (i.e., every few hours). The main drawback of short-term decision is its high VM migration overhead (migrating VMs every few seconds). Thus, as opposed to short-term decision, a long-term VM consolidation mechanism (migrating VMs every few hour for the purpose of better consolidation) is used such that the total demand of co-located VMs nearly reaches their host capacity during a long time.

In general, when deciding to place a set of VMs on a server, many works only check that the total size of the VMs does not exceed the server's capacity [14]. Hence, various server consolidation solutions are proposed based on per-VM workload characteristics, i.e., the peak, off-peak, and average utilization of VMs in a time series [12, 13]. However, there are a few studies [20, 22, 28, 29] to consider also other attributes of the VMs like CPU-load correlation that impacts how suitable they are for co-location to achieve further power savings. One approach is to simplify time-varying VMs' CPU utilizations, representing a single value to all CPU utilizations [22]. Another approach is to pair two uncorrelated VMs into a super-VM [28, 29]. However, both approaches do not work well in the presence of high variability in the VMs CPU utilization. As opposed to using stationary CPU-load values for correlations, there are few heuristic methods to separate CPU-load-correlated VMs with high variability in their

utilizations [20].

Network-Aware VM Allocation: To provide better network resource usage and, thereby, to improve the performance of applications, Virtual Data Centers (VDCs) are introduced instead of VM only [30]. Compared to VM-only offerings, VDCs are defined as a collection of VMs, switches, and routers that are interconnected through virtual links, and where each link is characterized by its bandwidth capacity and propagation delay [31]. In this case, VDCs provide a better isolation for network resources. This leads to a new challenge for data center providers to map virtual resources (e.g., VMs, switches, routers) onto the physical infrastructure.

Recently, some works addressed this problem presenting architectures and systems, like SecondNet and Oktopus, to allocate VDCs components requirement given by service providers to data center's equipment. SecondNet [32], a network virtualization architecture, demonstrates a greedy algorithm for bandwidth optimization, guaranteed to allocate resources to VDCs which in turn leads to high network utilization and low time complexity. Oktopus [33] implements two virtual network abstractions based on costs and provider revenue using greedy algorithms for mapping virtual resources to a tree-like physical topology. Furthermore, Zhani *et al.* [34] presented VDC Planner, a dynamic VDC framework for data centers that manages dynamic VM migration to achieve high revenue while minimizing the total energy cost over time. Their framework also supports a VDC consolidation algorithm to minimize the number of active physical servers during low-demand periods. These types of works neglect to adopt an energy-efficient method considering workload characteristics to further reduce energy consumption and maximize data center provider's profit. Also, the impact of green energy sources (renewables and EES) on energy savings are missing from these works.

Green Data Centers: Nowadays, many modern data centers are powered by brown energy generated from the power grid (e.g., fuel fossil and oil), which directly harms the environment. In this context, the importance of data centers energy consumption has been investigated based on increasing electricity bills and carbon dioxide footprints [35]. Moving toward data centers which are entirely or partially powered by renewable energy is a solution to reduce the carbon footprint and data centers operational cost. To address this challenge, allocation methods have been presented to adjust the available solar energy to computational workloads in a data center [36]. For solving this problem, several optimization methods (e.g., convex-mathematical model) have been used to maximize the total profit with respect to the stochastic nature of workload and SLAs requirements [37]. However, the use of such a strategy turns the optimization problem into an NP-hard problem. Therefore, this problem is not applicable at run-time and especially for large-scale data centers.

Other approaches utilize EES for a server activation strategy to maximize the usage of green energy using battery based on the workload [38]. For this type of solutions, it is crucial to consider both time-changing trends of renewable energy sources and power loss in battery bank due to aging and charging sequences. For instance, Goiri *et al.* [39, 40] first introduced Parasol, a prototype green data center, and then described GreenSwitch to dynamically sched-

ule workloads. GreenSwitch aimed at minimizing the overall cost of electricity and battery lifetime constraints while managing workloads and energy sources during grid outages.

1.2.2 Geo-Distributed Data Centers and Challenges

Operational Costs: One of the promising solutions for operational cost minimization is to adjust the number of active servers in different data centers, to which all user requests are distributed [41]. Nevertheless, the use of geo-distributed data centers allows designers to minimize the electricity cost (i.e., the main cost for data center providers) by exploiting dynamic workload allocation based on the temporal and regional diversity of electricity prices [42, 43, 44, 45]. Moreover, the advantage of Dynamic Voltage and Frequency Scaling (DVFS) can be investigated to further lower the electricity cost of geo-distributed data centers [46]. This optimization problem can be formulated as a Mixed Integer Linear Programming (MILP) problem, and then solved by a computation-efficient heuristic algorithm [41].

Following the same rationale, another approach consists of the dynamic pricing problem with respect to workloads delay constraints on their processing time [47]. Also, to use dynamic pricing in multiple data centers via migration, network latency delay plays a major role to distribute workloads across data centers according to data centers distance, potentially leading to an increase in response time [3]. For instance, Zhang *et al.* [48] utilized a model predictive control framework to determine the locations of service applications based on price and demand fluctuations. In this framework, services can dynamically be migrated over time while performance requirements (response time) are ensured. However, these approaches generally ignore the advantages of using renewable and battery energy sources as green energies to further minimize operational costs and carbon footprints.

Data Communication-Aware Allocation Schemes: Wide-area data transmission is the major contributor to the network costs [19]. Reportedly, inter-data center network cost accounts for 15% of the total infrastructure cost, which is much higher than the intra-data center network cost [49]. In this context, one key objective is to minimize the network traffic within the data centers, placing high-data communicating VMs in the same data center.

In the multiple data centers problem, data communication among VMs is an important aspect severely missing from many previous studies. A common approach to optimize network traffic and response time are presented with the assumption that data dependencies are given in the form of a Directed Acyclic Graph (DAG) [21, 23]. In this graph, the vertices represent the VMs and the edges depict data transfer and dependencies at each time. Differently, in practice, there are often cyclic communication scenarios, e.g., two VMs regularly exchanging information in both directions with different amounts. Moreover, these amounts change at run-time depending on real-time information. The complexity of this problem dramatically increases in geo-distributed data centers, where inter-data center VMs migration should be considered.

For instance, Agarwal *et al.* [50] defined a system called Volley to automatically migrate data across data centers. This solution uses an iterative optimization algorithm based on weighted spherical means considering the data locality, bandwidth costs, and storage capacity. The goal of this placement is to minimize user-perceived latency. Xin *et al.* [51] introduced an algorithm to split a request into partitions supporting virtual networking technology. This work has only focused on workload balancing. Cordeschi *et al.* [52] developed an optimal minimum-energy scheduler for the adaptive joint allocation of the task sizes, computing rates, communication rates and communication power in geo-distributed VDCs that operate under hard delay constraints. The goal is to minimize the overall communication and computing energy consumption by dividing the problem into two simpler sub-problems. However, these approaches neglect most of the main providers' objectives including operational costs, energy consumption, renewable and energy storage usage. Therefore, it is challenging for data center providers to conduct an efficient management to meet the trade-off between energy and network traffic considering the workload characteristics.

Green Data Centers: To address the environmental impact of electricity usage, power usage may be increased during the low electricity price periods, leading to higher carbon emission [53]. To mitigate the harmful effects of carbon footprint, data centers are equipped with renewable energy sources [4]. In order to avoid energy usage from the grid, techniques such as VM allocation and migration can be presented to utilize local green energy sources efficiently [54, 55, 56]. For instance, Ghamkhari *et al.* [56] introduced a QoS-aware workload distribution method to minimize the energy cost and carbon footprint, benefiting from the different types of renewable energy sources. For this purpose, they migrate the waiting workloads in each data center queue to another data center for execution.

Beside electricity cost optimization, another important aspect in geo-distributed data centers is to consider a delay cost for migrating workloads across data centers, which in turn affects the demand response time. Another approach to maximize renewables usage and reduce carbon footprint is to exploit EES, efficiently tackling the demand peak during the high-price periods [57, 58, 59]. Moreover, the complexity of the problem increases in multiple green data centers with heterogeneous physical servers [60], where trade-offs between the energy efficiency and performance of workloads must also be evaluated.

1.3 Emerging Power Markets and Challenges

With the increase in power usage, the electricity cost of data centers doubles every five years [61]. As mentioned in previous section (i.e., Section 1.2), the latest generation of data centers tend to use on-site (demand-side) EES systems and renewable energy sources (green data centers) to reduce costs, carbon emissions, and their dependency on energy from the power grid [4, 8]. However, due to the instability and high variability of renewable energy production (i.e., solar and wind), matching the demand-side renewable production and load in a green data center is a challenging task, which forces data centers to be connected to the power grid.

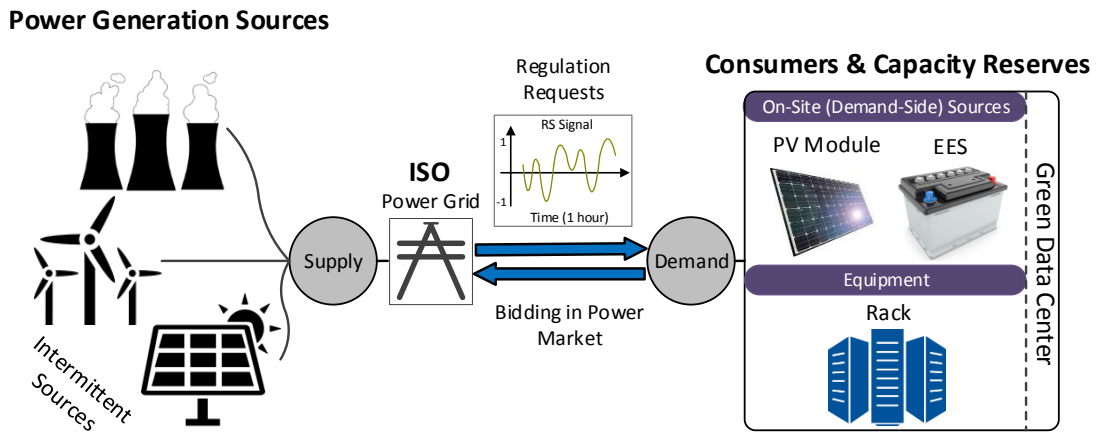

Figure 1.6 – Structure of the supply-side and demand-side (consumers) as capacity reserves.

Renewable energy sources are also being integrated on the supply-side. In fact, the European Union (EU) aims to integrate over 20% share of renewables in gross energy production for carbon emission reduction by 2020 [62], and a growth of 52% is expected in the US by 2040 [63]. However, with growing integration of renewables into the grid, the volatility and intermittency of renewable generation provide higher uncertainty to Independent System Operators (ISOs), where need to match supply and demand in the power grid in real-time. One potential solution in emerging power markets that provide competitive prices and services for the consumers is to use demand-side capacity reserves [64, 65]. That is, the ISO requests consumers to adapt their power consumption depending on its requirements (supply-demand matching). As data centers are among the fastest growing electricity consumers, they are highly promising candidates to provide demand-side capacity reserves and reduce their electricity costs.

Among the various types of capacity reserves, Regulation Service (RS) reserves [65] are particularly interesting for green data centers due to the relatively high value of such reserves and capabilities of data centers for providing high flexibility in their power consumption. In RS reserves provision, the demand-side (i.e., green data center) must dynamically modulate its power consumption to follow an RS signal broadcasted by the ISO every few seconds. In this scenario (depicted in Fig. 1.6), the demand-side acts as a capacity reserve that stabilizes the ISO power from the intermittency of renewable energies, and benefits from the power market rewards. However, the demand-side itself is also affected by the instability of on-site renewables. This poses an important challenge on the electricity cost and power minimization of green data centers.

Recently, several studies have evaluated the capabilities and benefits of RS reserves provision in data centers [66, 67, 68]. However, most of these studies disregard the use of demand-side renewables and EES when computing average power and reserve values in emerging power market bidding. In addition, an online policy that is aware of data center energy sources and workload constraints is required to track the RS signal. The prior works [69, 70] present an online tracking policy that exploits different server power modes to regulate server and data

center power consumption. However, they do not consider the demand-side renewable and EES usage, as well as optimizing the number of active servers and workload co-allocation in one problem. Thus, they cannot provide the best solution when computing the average power and reserve values.

A major challenge in this context is that the computation of the best power consumption and reserve values (bidding), and the RS tracking problem are largely impacted by the availability of demand-side renewable and EES, incoming workload, efficient server selection, and VM allocation policies. Therefore, to achieve the highest savings, a low-overhead method that incorporates all these aspects is required.

1.4 Next-Generation Servers and Data Centers

Even if data center resource management techniques, mentioned in Section 1.2 and 1.3, can efficiently tackle the dramatic increase in the number of servers, each computing server remains power limited due to effect of post-Dennard scaling. In order to maximize energy efficiency (i.e., performance per watt), customized server architectures can be used to increase throughput by identifying and eliminating the bottlenecks in conventional server processors. However, energy reduction in deep sub-micron technologies has lagged behind, resulting in power-limited servers, and chip underutilization [71].

A promising approach to overcome the power bottlenecks is Near-Threshold Computing (NTC). NTC takes advantage of the quadratic dependency between the supply voltage and dynamic power consumption, by lowering the operating voltage to a value slightly higher than the transistor threshold voltage, increasing energy efficiency at the expense of reduced performance. However, for current cloud computing applications, NTC allows adjusting supply voltage to optimize the trade-off between performance and power, emerging as a promising approach to overcome the power-wall [72].

From a technology viewpoint, the Ultra-Thin Body and Buried Oxide (UTBB) Fully Depleted Silicon On Insulator (FD-SOI) technology has demonstrated its suitability for NTC. In contrast to traditional bulk technology, FD-SOI features a significantly increased voltage range and even higher performance for the same energy thanks to the better behavior of transistors at low voltage [73]. Moreover, the extended body bias range enabled by this technology allows further reduction in the supply voltage for a given frequency target, and further minimization of dynamic power consumption [73]. The 28nm FD-SOI technology process is currently employed for mass production by Samsung and ST Microelectronics; the 20nm technology is being produced by GlobalFoundries while the 12nm node is on the strategic roadmap [74]. As shown in Fig. 1.7, with respect to Fin Field-Effect Transistor (FinFET) technology, FD-SOI provides a cost-sensitive solution for low-power (both active and leakage) systems without increasing die cost [75], but at the expense of performance. These features make FD-SOI a suitable solution for next generation near-threshold servers.

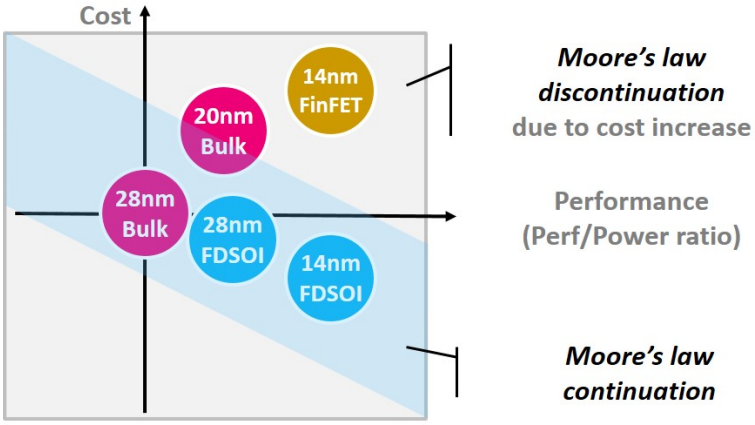

Figure 1.7 – FD-SOI versus FinFET technology suitability for NTC [76].

The new trade-offs brought by the FD-SOI technology and NTC servers, and the analysis of its impact on data center level energy-aware policies, remains an open challenge. VM consolidation [14] has represented for years the most widely used technique to minimize energy consumption. However, the emergence of energy-proportional NTC servers, with drastically reduced static power, together with the advent of applications able to work at reduced frequencies, changes the underlying assumptions that made consolidation the best choice for energy efficiency.

1.5 Thesis Contributions

The main goal of this thesis is to develop a set of system-level techniques to improve the efficiency of servers and cloud data centers. In particular, the contributions of my work can be grouped as follows:

1.5.1 Efficient Workload Allocation

Due to the highly dynamic nature of workloads and data communication, data center providers face significant challenges in managing application performance while minimizing energy-related costs. This context motivates the adoption of efficient multi-objective management techniques for data centers. In this regard, I first propose and assess two different approaches to tackle the VM allocation problem: i) a two-phase greedy heuristic, and ii) a Machine Learning (ML)-based approach. Both approaches exploit CPU-load and data correlations, together with information about data center network topology. The strategies consolidate VMs into the minimum number of servers and racks, and set DVFS appropriately. Then, I present a novel hyper-heuristic method that integrates the strengths of both heuristic and ML methods. The approaches are evaluated in terms of energy consumption, QoS degradation, network traffic, number of migrations and scalability, and are compared to an Integer Linear

Programming (ILP)-based optimal solution and other state-of-the-art methods.

In particular, the main contributions are as follows:

- I propose a multi-objective hyper-heuristic method to determine dynamically which method among heuristic and ML is to be used at each time.
- I propose two energy- and network-aware VM allocation methods: i) a two-phase greedy heuristic, and ii) a low-complexity ML-based approach that uses the value iteration algorithm to assign VMs to servers.
- I provide an evaluation of the flexibility, scalability, benefits and drawbacks of heuristic versus ML methods for the highly dynamic and complex VM allocation problem.
- I compare the proposed solutions with an ILP-based method, that provides an optimal solution, and also with two methods in the state-of-the-art.

1.5.2 Multi-Objective Optimization for Green Data Centers

1.5.2.1 Green Data Centers Framework

In this thesis, I introduce and propose a multi-level and multi-objective framework for the optimization of green virtualized data centers, to jointly minimize the energy consumption and the carbon footprint, exploiting renewable energy sources, VM allocation schemes and HES. With HES, I refer to EES where different battery technologies are employed together, allowing to compensate for the inherent drawbacks of each other (e.g., life cycle, capacity, cost, charge/discharge speed, etc.). The framework consists of two modules running concurrently, a data center energy controller that manages the energy consumption of data center and shares the real energy consumption data with green energy controller; and a green energy controller that manages renewable sources and HES, providing feedback to the data center energy controller.

In current data centers, insufficient efforts have been dedicated to implement adaptive energy reduction techniques and real-time resource scheduling to efficiently manage IT equipment and renewable energy sources. The framework consists of an HES architecture to replace standard Uninterruptible Power Supply (UPS) systems, which allows an active management and the full exploitation of the energy buffers for the locally-generated renewable energy. I also designed a dedicated control loop which connects the VMs allocation scheme to the HES manager and optimizes the resources in real-time. In the following part of my research, I explore the efficiency of proposed VM allocation method for green data centers implemented on this framework, considering the battery limitations and lifetime, to simulate a realistic scenario.

1.5.2.2 VM Allocation Method for Green Geo-Distributed Data Centers

I present a two-phase multi-objective VM placement (i.e., clustering and allocation) algorithm along with a dynamic migration technique for geo-distributed data centers. The first phase, i.e., global controller, clusters VMs for each data center exploiting both CPU-load and data correlations based on data centers status (current electricity price, battery information, renewable energy forecast and VMs utilization prediction). Meanwhile, during the cluster creation process, the VMs need to be migrated across data centers while avoiding QoS degradation by defining a hard time constraint. The second phase, i.e., local controller which is used in each data center, allocates the VMs of each data center cluster to servers considering CPU-load correlation. The proposed algorithm optimizes the operational costs, data center energy consumption, network traffic and response time while maximizing the renewable energy and battery usage. Due to the usage of VMs CPU and memory utilization prediction and renewable energy forecast, an online green controller is also considered to manage battery, renewable, and grid energy sources based on real renewable energy and VMs utilization rates. In particular, I address dynamic arrivals of VMs into the system, as well as different VMs completion times. Briefly, this work is the first to propose a multi-objective VM placement for green geo-distributed data centers exploiting CPU-load and bidirectional data correlations (i.e., different amounts of data exchanged among VMs in both directions) in one problem.

Compared to previous studies, the contributions are as follows:

- I jointly incorporate CPU-load and data correlations to address the energy-performance trade-off in multiple data centers. Since in practice, each pair of VMs regularly exchange information in both directions, I consider bidirectional data correlations which change at run-time.

- I propose a two-phase controller along with a migration technique that splits the complex VM placement problem into clustering and allocation phases. In the first phase, the global controller exploits the CPU-load and data correlations to cluster the VMs for the data centers. In the second phase, the local controllers distributed into the data centers, allocate the VMs of each data center cluster to servers exploiting the CPU-load correlation.

- I define and formulate this problem for geo-distributed data centers connected through a network topology to address the energy-performance trade-off and operational cost minimization while maximizing the use of renewable and battery energies.

- I optimize the whole problem to find the best solution based on load and renewable forecast information. Therefore, I am able to adopt a low-complexity rule-based green controller to compensate the difference between real and forecasted information.

1.5.3 Data Center Cost Optimization in Emerging Power Markets

Todays, despite using demand-side renewable energy sources to minimize cost, power market operators have introduced interesting demand-response offers and programs for the electricity consumers. In these programs, electricity consumers regulate their power usage following provider requests. Among different programs, RS reserves are interesting offers for data centers due to the high credit gain possibilities and data centers' flexibility in regulating their power consumption. Hence, it is essential to develop bidding strategies for data centers to participate in emerging power markets together with power management policies that are aware of power market requirements at runtime.

In order to minimize the monetary cost of data centers in emerging power markets, I first optimize the power market bidding parameters together with the VMs allocation in a fast analytical way with respect to the available demand-side EES and renewable power that satisfies the power market constraints. Then, I propose an online tracking policy that considers VMs CPU resource limits and efficiently utilizes EES and renewable power, while guaranteeing QoS requirements.

In particular, the main contributions are as follows:

- I introduce ECOGreen, a new Electricity Cost Optimization strategy for Green data centers that computes the best average power and reserve bidding problem considering the renewable and EES energy for RS reserves provision in emerging power markets, along with determining the number of active servers.
- I jointly manage VM allocation with the use of demand-side renewable and EES in RS reserves provision. To this end, I consider both time-changing trends of renewable energy sources and power loss in battery bank due to aging and charging sequences.
- I develop an online policy that enables a green data center to regulate its power and track the RS signal broadcasted every few seconds accurately, while also guaranteeing QoS constraints.

1.5.4 Energy Proportionality in Near-Threshold Computing (NTC) Servers and Cloud Data Centers

Despite novel data center resource management techniques can efficiently tackle the dramatic increase in the number of servers, each computing server is still power-limited due to effect of post-Dennard scaling. Therefore, techniques such as NTC need to complement novel system-level approaches to improve data centers' energy efficiency. NTC increases energy efficiency by lowering the operating voltage to a value slightly higher than the transistor threshold.

In this thesis, I evaluate the impact of VM consolidation on new architectures for NTC servers using accurate power models. I demonstrate the stagnation of consolidation-based and

server turn-off techniques for NTC-based data centers. In this respect, I propose a new policy able to provide significant energy savings when compared to state-of-the-art consolidation approaches.

In particular, the contributions of the work are as follows:

- I show how the energy-proportionality of NTC servers, enabled by the new FD-SOI technology, results in a paradigm shift in which traditional VM consolidation and server turn-off strategies no longer yield optimal results in terms of energy consumption.
- I propose the Energy Proportionality-Aware dynamiC allocaTion (EPACT) method, a novel data center workload allocation policy for NTC servers, which also selects the best DVFS setup. The approach increases the energy proportionality of NTC-based data centers, outperforming latest consolidation techniques, while guaranteeing QoS requirements.
- I assess the performance and efficiency of virtualized workloads on three architectures: i) x86, ii) ARM-based Cavium ThunderX, and iii) a new architecture for NTC server, which modifies and improves the efficiency of the ThunderX architecture.

Finally, as a future work, I would like to cover future trends on server processors and technologies, in particular benefiting from the other features of FD-SOI technology (e.g., Forward Body Biasing (FBB)) to increase the performance of applications.

1.6 Thesis Organization

The remainder of this thesis follows the same structure than the one detailed in Section 1.5. Each chapter will provide the necessary background and a separate review of the related works. In particular, the content is organized as follows:

Chapter 2 presents a heuristic and a ML-based VM allocation method for various data center scenarios. I also present a novel hyper-heuristic algorithm that exploits the benefits of both methods by dynamically finding the best algorithm, according to a user-defined metric. For optimality assessment, I formulate an ILP-based VM allocation method to minimize energy consumption and data communication in a single data center, which obtains optimal results. I first review related work and mark the differences. Then, I describe the system model and used application. Finally, I provide an overview of the problem description, followed by the proposed methods and the experimental results.

Chapter 3 first introduces the overall system modeling used for green data centers. Second, I present a framework (in-house simulator tool) for the optimization of green virtualized data centers, to jointly minimize the energy consumption and the carbon footprint, exploiting renewable energy sources, VMs allocation schemes and HES, in different scenarios. Third,

for green geo-distributed data centers, I introduce the proposed optimization scheme implemented in the framework, followed by the experimental results. Finally, I provide an overview of the problem description and target optimization scenario for the participation of green data centers in emerging power markets. For this problem, I first solve the RS bidding problem, and then provide a run-time policy that dynamically regulates the green data center power to track the RS signal in real time.

Chapter 4 first provides the server architecture and power models for NTC servers based on FD-SOI technology. Then, I describe the applications along with exploring the existing energy versus performance trade-offs when virtualized applications with different CPU utilization and memory footprint characteristics are executed. Finally, based on this analysis, I propose a novel dynamic energy-efficient VM allocation method that exploits the knowledge of VMs characteristics together with the accurate server power model for next-generation NTC-based data centers.

Chapter 5 finally concludes the thesis by summarizing the key contributions and providing important items for future research in the same direction.

2 Efficient Workload Allocation in Single Data Center

2.1 Introduction

The cloud offers a wide variety of services from scale-out applications (where individual servers dictate overall performance) to traditional virtualized applications and batching (e.g., banking applications), as well as various deployment models. The deployment model ranges from publicly available clouds where services are available on a pay-as-you-go basis, to private clouds internally owned by an organization. Both the particular application requirements and the deployment model greatly affect the techniques that can be applied to leverage energy efficiency and performance. In this chapter, I focus on privately-owned data centers, running an heterogeneous set of virtualized services with uneven CPU-load and data communication patterns. As a case study, I consider banking applications executing batches of tasks. Therefore, to address the energy efficiency challenge in such scenarios, it is essential to develop resource provisioning policies that are aware of Virtual Machine (VM) characteristics, such as CPU-load and data communication, while at the same time being applicable in real scenarios. CPU-load and data correlations indicate opposed goals, as highly CPU-load correlated VMs (i.e., VMs with highly similar utilization traces and whose peaks coincide) should be placed apart to reduce the amount of active servers for energy efficiency, while highly data-correlated VMs (i.e., VMs that exchange large amount of information) should be clustered together to minimize network traffic and response time. Thus, there is an interesting trade-off for the efficient energy-performance management of these workloads. In this context, several greedy heuristic-based works either address CPU-load correlation, consolidating VMs when their peak utilizations do not coincide in time, or take data correlation (data exchange across VMs) into account. Nonetheless, jointly incorporating CPU-load and data correlations is an important aspect, which dramatically increases the complexity of the VM allocation problem.

Because of the above mentioned complex nature of the VM allocation problem and the large number of constraints, finding an optimal solution is unfeasible at run-time due to its high computational overhead. Therefore, heuristic methods are generally used to speed up the problem solving. However, heuristic algorithms are problem-specific and less sensitive to

dynamic environments, and thus their benefits become more limited when CPU utilization and data communication among VMs dynamically change every time. In addition, heuristics fall short in terms of flexibility and applicability for large-scale VM allocation scenarios. Thus, when tackling dynamic problems with large state and/or action spaces (solution space), Machine Learning (ML), and in particular Reinforcement Learning (RL), is the most promising solution [77]. However, each technique has its own benefits and drawbacks. In order to balance the trade-offs across different metrics, or dynamically change the optimization goals during run-time, a deep assessment is required to integrate the strengths of different methods. Within this context, hyper-heuristics [78] are *"heuristics that choose heuristics"* and allow to leverage the benefits of VM allocation approaches, determining which method can be used depending on the current data center status to provide better trade-offs than when using the methods separately.

2.1.1 Contributions

In this chapter, I propose and compare two different approaches to tackle the VM allocation problem: i) a greedy heuristic and ii) a ML-based approach. Both approaches exploit CPU-load and data correlations, together with information about data center network topology. The strategies aim at consolidating the VMs into the minimum number of servers and racks, turning off the unused ones, and setting Dynamic Voltage and Frequency Scaling (DVFS) appropriately. The approaches are evaluated in terms of scalability, energy consumption, performance (network traffic), and Quality-of-Service (QoS) degradation, and they are compared to an Integer Linear Programming (ILP)-based optimization method (optimal solution). Moreover, I propose a hyper-heuristic method to find better solutions for the VM allocation problem. The proposed algorithm dynamically determines which method among the heuristic and the ML is to be used at each time. In this case, better candidate solutions can be found based on trade-offs between different considered objectives. Thus, the strengths of the two approaches are combined together in one problem.

2.2 State-of-the-art and Comparison of VM Allocation Methods

Research on VM allocation can be generally categorized in energy- and network-aware methods.

2.2.1 Energy-Aware VM Allocation

Regarding energy-aware methods, when deciding the allocation of VMs to physical servers, several works only check that the total size of VMs' load does not exceed the server's capacity [13, 14, 79, 80, 81, 82]. Hence, consolidation solutions are proposed based on per-VM workload characteristics, i.e., the peak, off-peak, and average utilization of VMs [12, 13, 83]. Ahvar *et al.* [84] present a cost and carbon emission-efficient VM placement method and

optimize network between data centers, using fuzzy sets. A few studies [20, 22, 28, 29, 85, 86] consider other VM's attributes, like CPU-load correlation, to achieve further energy savings. Among the latter, Verma *et al.* [22] define VMs' CPU utilization in a time series as a binary sequence where the value becomes '1' when CPU utilization is higher than a threshold. However, this aggressive quantization alters the original behavior and is only applicable when VM envelops are stationary. Meng *et al.* [28] propose a VM sizing technique that pairs two uncorrelated VMs into a super-VM by predicting the workloads. Nevertheless, once the super-VMs are formed, this solution does not consider dynamic changes, which limits further energy savings. Kim *et al.* [20] present a CPU-load correlation-aware solution based on the First-Fit-Decreasing heuristic to separate CPU-load correlated VMs. The main drawback of this approach is that it cannot be used for online management at large-scale data centers due to its high computational overhead. Lin *et al.* [85] utilize the peak workload characteristics to measure the similarity of VMs' workload. This method achieves better results for VMs whose workload follows a Gaussian distribution. Ruan *et al.* [87] propose a dynamic migration-based VM allocation method to achieve the optimal balance between server utilization and energy consumption such that all servers operate at the highest performance-to-power levels. Wang *et al.* [88] also address a matching-based VM consolidation mechanism using migration such that active servers can operate close to a desirable utilization threshold.

ML is a recently used technique for energy-aware VM allocation in data centers. Farahnakian *et al.* [89] and Masoumzadeh *et al.* [90] present a cooperative multi-agent learning management to minimize the number of active servers managing the overutilized and underutilized servers. Masoumzadeh *et al.* [91] introduce a VM selection task using a fuzzy Q-learning technique to make decisions for migration. Ravi *et al.* [92] also present an energy-efficient Q-learning based technique to decide on VM migrations. The main drawback of those approaches is their high VM migration overhead. Thus, as opposed to short-term decision, Chen *et al.* [93] propose a long-term VM consolidation mechanism such that the total demand of co-located VMs nearly reaches their host capacity during their lifetime period. This algorithm first detects the utilization pattern of each VM based on the four types of simple pulse functions. Then, a heuristic algorithm is used to place all VMs in as few servers as possible. They show a significant reduction in the number of migrations, i.e., only 4% of the total number of VMs, compared to dynamic short-term decision-based methods. Nonetheless, this work ignores the original utilization pattern of the VMs, which is usually a combination of those simple types of functions, achieving lower energy savings. Moreover, none of these approaches consider the data communication between VMs in the allocation process.

2.2.2 Network-Aware VM Allocation

To provide better network resource usage and improve the performance of applications, certain algorithms [21, 23, 84] take into account the communication among VMs in the data center. However, some works assume that data dependencies are given in the form of a Directed Acyclic Graph (DAG). Differently, in practice there are often cyclic communication scenarios,

where two VMs regularly exchange information in both directions. As a result, Biran *et al.* [16] propose two new heuristic algorithms to address bidirectional data communication under time-varying traffic demands. However, without the consideration of CPU-load correlation, both approaches are sub-optimal to minimize energy consumption.

2.3 Data Center Model and Application Description

In this section I first define the considered data center configuration and describe the used power model. Then, I detail the type of applications tackled in this chapter. For the sake of clarity, Table 2.1 summarizes the main parameters and notations used throughout this chapter.

2.3.1 Data Center Configuration

A data center typically encompasses two main structural components of the interest: i) computing elements –Information Technology (IT)– comprising servers and network switches, and ii) cooling systems. A typical raised-floor air-cooled data center [26] is considered with 8 racks arranged in a hot-cold aisle topology, as depicted in Fig. 2.1. Each rack contains 10 to 16 servers, each server with its dedicated power unit, fans and disks. Each rack has one Top-of-Rack (ToR) switch to which all servers in the rack are physically attached, providing a bandwidth B_{tor}. The ToR switch is the lowest layer of a three-layer tree network topology [79]. Several ToR switches connect to an aggregate switch, which consolidates traffic into a higher speed link with bandwidth B_{agr}. The Aggregation-layer switches connect to a core router that redirects incoming requests to servers, and tracks and routes VM migrations from one server to another server or data center in another location. The Core router operates in network

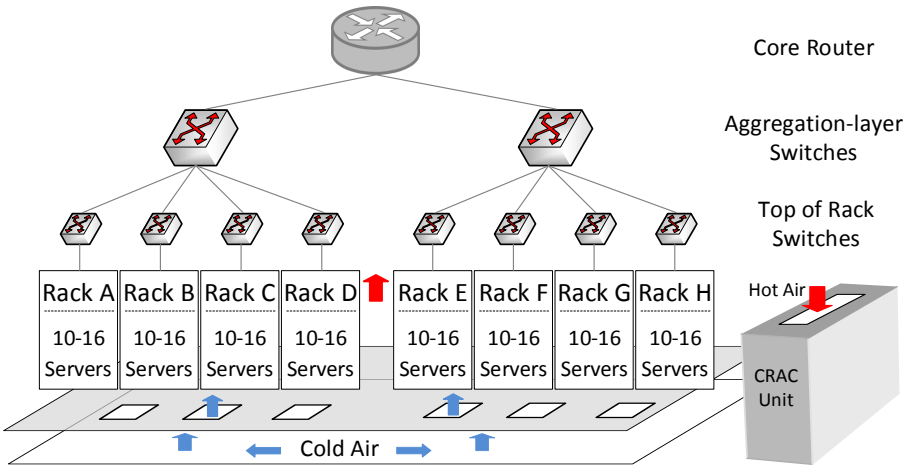

Figure 2.1 – Considered data center configuration: location of servers, cooling system and multi-layer network topology.

Table 2.1 – Overview of the used notation

General Parameters and Variables			
N_s	Total number of servers in data center	$VMcpu$	VMs CPU utilization traces
N_{VM}	Total number of VMs in data center	$VMmem$	VMs memory footprint traces
N_t	Number of samples per time slot	$VMdata$	Data communication demands between VMs
f^{max}	Maximum frequency level of each server	$Freq_j^T$	Selected frequency level of j^{th} server in time slot T
B_{tor}	ToR switch bandwidth	$U^T_{cpu_j,n}$	j^{th} server CPU utilization at n^{th} sample in time slot T
B_{agr}	Aggregation-layer switch bandwidth	$U^T_{mem_j,n}$	j^{th} server memory utilization at n^{th} sample in time slot T
B_{cr}	Core router bandwidth	t_f	Servers' frequency update time during one time slot
C^s	Maximum CPU capacity of each server	t_c	Cooling system update time during one time slot
C^m	Maximum memory capacity of each server		
General Power Model Parameters and Variables			
P_{DC}	Data center total power consumption	P_s	Total server power consumption
P_c	Cooling power consumption	P_j	j^{th} server power consumption
P_{IT}	IT power consumption	P_{net}	Total network power consumption
ILP-Based Method Parameters and Variables			
X_j^T	j^{th} server is on ($X_j^T = 1$) or off ($X_j^T = 0$) in T	$D^T_{j,n}$	j^{th} server data communication at n^{th} sample in T
$Place^T_{j,k}$	Whether k^{th} VM is placed on j^{th} server in T	$VMstatus_{j,k,l}$	Placement status of any pair of VMs on j^{th} server
e_j^T	Number of placed VMs on j^{th} server	$BinVMstatus_{j,k,l}$	Whether k^{th} and l^{th} VMs have been allocated to j^{th} server
D^T_{total}	Total data communication amongst servers		
Heuristic Method Parameters and Variables			
$\hat{N}_{server}$	Minimum number of servers to accommodate all VMs	NT^r_{tor}	ToR switch traffic of r^{th} rack
G_{data}	Inter-cluster data communication graph	NT^h_{agr}	Aggregation-layer switch traffic of h^{th} group of racks
ML Method Parameters and Variables			
$\phi^T_{k,l}$	Similarity score between VM_k and VM_l in T	$\phi^{class}_{avg,T}$	Average similarity score among all classes per T
$\rho^T_{k,l}$	Pearson correlation similarity on any pair of VMs	N^{server}_{VM}	Maximum number of VMs can be allocated to each server
$Dist^T_{k,l}$	Euclidean distance between k^{th} and l^{th} VM features	R_j	Total reward value per server
K	Number of classes	λ	Weighting factor to keep reward factors in the same range
N_ω	Number of VMs available in class ω	$\hat{U}^T_{cpu_j}$	Maximum utilization of j^{th} server among samples in T
$\phi^{class}_{\omega,T}$	Per-class (ω) similarity score		
Hyper-Heuristic Method Parameters and Variables			
$\mathbb{O}$	Selected objectives set	$Cost_i$	Cost value per method i
α **and** β	User-defined weighting factors to objectives	Num_i	Number of times that i^{th} method is selected
$\mathbb{M}$	Pool of candidate methods		

backbone with the highest speed and bandwidth B_{cr}.

The data center is cooled using one Computer Room Air Conditioning (CRAC) unit [26, 94, 95] that circulates hot and cold air to keep the servers temperatures below a certain limit. Cold air flows through the perforated tiles located on the raised floor, and then is intaken by the server fans from the cold aisles. The hot air exits the back side of servers toward hot aisles and then leaves the data center room through the hot air intakes located on the the ceiling above hot aisles.

2.3.2 Data Center Power Model

The total data center power consumption (P_{DC}) is modeled as the sum of: i) IT power (P_{IT}), including total server (P_s) and network (P_{net}) power, and ii) the cooling power (P_c):

$$\begin{aligned} P_{DC} &= P_{IT} + P_c \\ P_{IT} &= P_s + P_{net} \end{aligned} \tag{2.1}$$

Server power can be further extended as: $P_s = \sum_{j=1}^{N_s} P_j$, where P_j and N_s specify the j^{th} server power and the total number of servers in the data center, respectively. Following the same methodology as in previous research in the area [96, 97], the major contributors to power consumption in servers are considered to be the CPU, memory, fans, and disks. Among these, the CPU has the largest effect on power, and previous research shows that the power-frequency relation is linear for a given CPU-intensive workload [98]. Hence, server power can be calculated as [96]:

$$\begin{aligned} P_j &= P_{j_{static}} + P_{j_{dyn}} \\ \begin{cases} P_{j_{static}} = P_{disk} + P_{fan} + P_{cpu}^{leak} + P_{cpu}^{idle} + P_{mem}^{idle} \\ P_{j_{dyn}} = P_{cpu}^{dyn} \cdot (U_{cpu_j}/100) + P_{mem}^{dyn} \cdot (U_{mem_j}/100) \end{cases} \end{aligned} \tag{2.2}$$

where $P_{j_{static}}$ indicates all the contributions to power that are workload-independent. P_{disk} and P_{fan} are considered constants for the particular workload, and respectively account for the power consumption of disks and fans. P_{cpu}^{leak} refers to temperature-dependent leakage power. A high fan speed and a low inlet temperature are taken into account to reduce the effect of temperature-dependent leakage power, considering it as a worst-case constant. P_{cpu}^{idle} and P_{mem}^{idle} are constants that show the idle power consumption of CPU and memory, respectively. $P_{j_{dyn}}$ accounts for server dynamic power, and is proportional to workload. P_{cpu}^{dyn} and P_{mem}^{dyn} are the fitted constants of the linear model for dynamic CPU and memory power, respectively. Finally, U_{cpu_j} and U_{mem_j} represent CPU and memory utilization of the server, and vary between 0 and 100. The fitted values of P_{cpu}^{dyn} and P_{mem}^{dyn} have been obtained from the models proposed in previous work [96, 97], assuming that for the workloads (i.e., virtualized banking applications) memory utilization is proportional to the amount of memory accesses.

Network power (P_{net}) is the summation of power of all turned-on switches in each layer of network topology, as [79, 99]:

$$P_{net} = P_{tor} + P_{agr} + P_{cr} \tag{2.3}$$

where P_{tor}, P_{agr} and P_{cr} denote power consumption of ToR and aggregation-layer switches, and core router respectively.

Finally, to account for cooling power consumption, P_c, a time-varying Power Usage Effectiveness (PUE) model is used, proportional to power consumed by IT components; i.e. $P_c = (PUE - 1) \cdot P_{IT}$, as presented in prior research [100]. In this model, PUE mainly depends on the data center room and outside temperature.

2.3.3 Applications Description

Due to the tight dependency between the nature of the applications and the techniques to be applied, I consider business-critical applications [101] and, in particular, virtualized banking applications executing batches of tasks. To characterize the power and performance of these applications, I make use of synthetic workloads which are representative of real banking applications according to our industry partners. As realistic CPU usage and memory footprint traces, I use the publicly available traces from Bitbrains, a service provider that provides service to banks such as ING [101]. Bitbrains traces provide data every 5 minutes. Half of the VMs have low variance on CPU usage. However, around 20% of VMs have a very unstable CPU usage. Concerning memory footprint, 80% of VMs use less than 1GB of memory, and in most of them maximum is below 8GB [101].

The Bitbrains traces do not provide any information on the amount of data being exchanged across VMs. Thus, the amount of data communicated between each pair of VMs is synthetically generated using a non-uniform distribution. Basically, data communication between a pair of VMs is modeled by a log-normal distribution [102] as 80% of the VMs have 800 kB/min traffic among each other while 4% of the VMs have 10 times higher traffic [103].

2.4 Problem Description

In this section, I present an overview of the problem description, including the objectives, inputs and outputs. The goal of all the methods proposed in this chapter is to minimize the overall server (P_s) and accordingly data center (P_{DC}) power consumption, and network traffic (D_{total}) by means of efficient consolidation-based VM allocation. All the proposed approaches examined consist of two consecutive steps: i) VM characterization and ii) VM allocation, as shown in Fig. 2.2.

The VM characterization step is used to determine the data communication patterns between VMs, the CPU utilization, and the memory requirements for the next time slot. For the heuristic

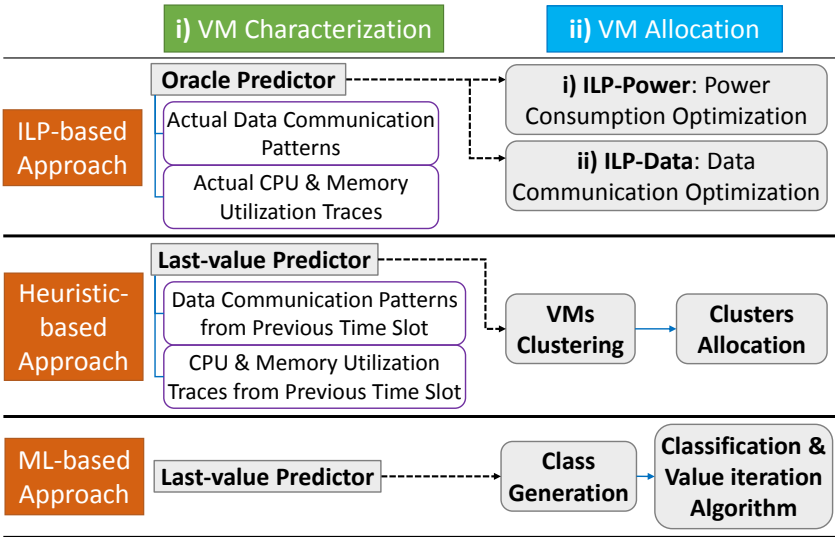

Figure 2.2 – Overall diagram of the proposed scenario.

and ML approach, a last-value predictor is used to estimate these parameters. The last-value predictor considers that the CPU and memory utilization traces of the current time slot (e.g., a time series of n samples, each sample gathered every 5 minutes) are exactly the same as on the previous time slot, as shown in Fig. 2.3. The specified areas in the figure indicate miss-predictions that can potentially lead to server overutilization and violations, when the predicted CPU utilization is lower than the real one. For the ILP-based method, it is assumed that, at the beginning of each time slot, all the VM characteristics for the time slot are known (oracle predictor).

The VM Allocation step takes the input VMs CPU, memory, and data communication requirements from the previous step, and the data center network topology. Every time slot, the VM allocation method re-allocates the existing VMs, migrating them if needed to the minimum number of servers such that highly data-correlated VMs are placed together, while highly CPU-load correlated VMs are placed apart. By lowering the number of active servers and racks, unused computing equipment (i.e., idle servers and network switches) can be turned off during that time slot to increase energy efficiency. Turning on/off IT equipment can be applicable to such applications when time slot duration is long enough to prevent significant performance degradation caused by the long transition latency between power modes and changes of resource demands.

After allocating all VMs to the minimum number of servers, the minimum frequency level ($Freq_j^T$) among all the samples in time slot T for each turned-on server is computed as:

$$Freq_j^T \geq (U_{cpu_j,n}^T/100) \cdot f^{max}, \quad \forall n \tag{2.4}$$

where $U_{cpu_j,n}^T$ indicates the total CPU utilization of the j^{th} server at n^{th} sample in time slot T. In addition, a homogeneous data center is considered with all servers of the same type.

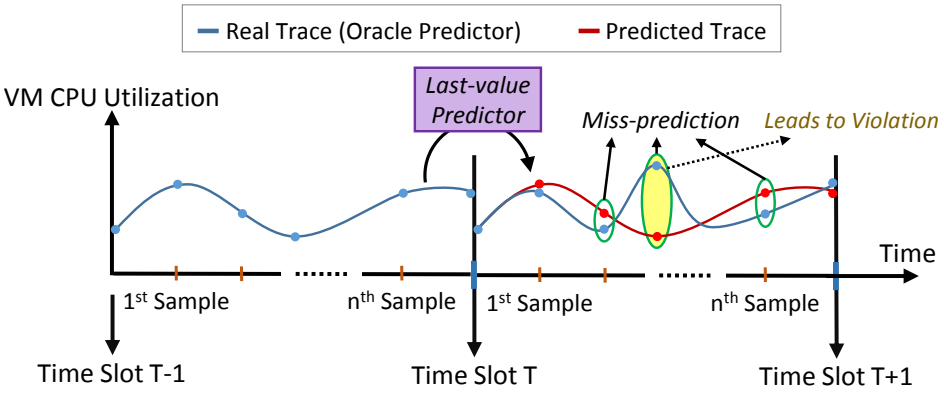

Figure 2.3 – Time slot and sample description.

Therefore, f^{max} is equal for all servers and determines the maximum frequency level. This equation guarantees that the selected frequency is sufficient to avoid server overutilization for all the samples in time slot T. Thus, in the proposed scenario, violations can only occur due to miss-predictions on the VM usage (e.g., for the case highlighted in Fig. 2.3), when the server utilization needs a higher frequency than the one selected. In this sense, the ILP method will never exhibit violations, as it uses an oracle predictor. After allocation, for all the proposed methods during one time slot, the servers frequency is updated every t_f by computing the maximum server utilization occurred in the previous t_f period. Finally, after computing the total IT power consumption, the cooling system power is updated and computed every t_c during one time slot, based on the data center room and outside temperatures.

2.5 Proposed Integer Linear Programming (ILP)-Based Optimization Method

The ILP method can be divided in two minimization problems: i) power consumption (ILP-Power), and ii) data communication (ILP-Data). Due to the varying nature of data center workloads (i.e., VMs' utilization patterns change over time), there is an optimal VM allocation for each time slot and, thus, a need to invoke the ILP-based method. The time slot duration is a parameter that can be adjusted by the data center operator depending on the granularity of the traces used, to increase accuracy. The ILP formulations are defined in a generic form regardless of the network topology constraints. Hence, an optimal server-to-server data communication is obtained that represents the total network traffic. In the following subsections, I describe these two optimization goals in detail.

2.5.1 Power Consumption Optimization

The proposed ILP-Power VM allocation aims at minimizing overall server power based on the Eq. 3.19. The minimization objective is given by Eq. 2.5, where P_s^T and $P_{j,n}^T$ denote overall

and j^{th} server power consumption at the n^{th} sample of the T^{th} time slot, respectively. N_s and N_t are the number of servers in the data center and number of samples in one time slot, respectively. The binary variable X_j^T is defined to indicate whether the j^{th} server is on ($X_j^T = 1$) or off ($X_j^T = 0$) in the T^{th} time slot.

The binary variable $Place_{j,k}^T$ is used to indicate whether k^{th} VM ($k \in 1,2,...,N_{VM}$) is placed on j^{th} server in T^{th} time slot. N_{VM} is the total number of VMs available in the data center. Matrices $VMcpu_{k,n}^T$ and $VMmem_{k,n}^T$ contain the k^{th} VM's CPU utilization and memory footprint at n^{th} sample, respectively, during the T^{th} time slot. Similarly, $U_{mem_j,n}^T$ indicates the total memory utilization of the j^{th} server.

$$\begin{aligned} \min \; P_s^T &= \sum_{j=1}^{N_s} \sum_{n=1}^{N_t} X_j^T \cdot P_{j,n}^T \\ &= \sum_{j=1}^{N_s} \sum_{n=1}^{N_t} (X_j^T \cdot P_{j_{static}} + P_{j_{dyn},n}^T) \\ &= \sum_{j=1}^{N_s} \sum_{n=1}^{N_t} (X_j^T \cdot P_{j_{static}} + P_{cpu}^{dyn} \cdot U_{cpu_j,n}^T + P_{mem}^{dyn} \cdot U_{mem_j,n}^T) \end{aligned} \tag{2.5}$$

where

$$U_{cpu_j,n}^T = \sum_{k=1}^{N_{VM}} Place_{j,k}^T . VMcpu_{k,n}^T \tag{2.6}$$

$$U_{mem_j,n}^T = \sum_{k=1}^{N_{VM}} Place_{j,k}^T . VMmem_{k,n}^T \tag{2.7}$$

subject to the following Constraints:

1. $\sum_{j=1}^{N_s} Place_{j,k}^T = 1$ (2.8)
2. $U_{cpu_j,n}^T \leq C^s$ (2.9)
3. $U_{mem_j,n}^T \leq C^m$ (2.10)
4. $e_j^T = \sum_{k=1}^{N_{VM}} Place_{j,k}^T$ (2.11)
5. $X_j^T \leq e_j^T \leq X_j^T N_{VM}$ (2.12)

The minimization problem is subject to the constraints given in Eq. 2.8 to 2.12. Constraint 1 forces each VM to be placed only in one server. Constraints 2 and 3 enforce that aggregated CPU and memory utilizations of the VMs in j^{th} server do not exceed the maximum CPU and memory capacities, i.e., C^s and C^m, respectively. In Constraint 4, the integer variable e_j^T is used to specify the number of placed VMs on the j^{th} server. This value is upper-bounded by N_{VM}. Constraint 5 guarantees that if no running VMs are placed on j^{th} server in the T^{th} time slot ($e_j^T = 0$), this server can be turned off ($X_j^T = 0$).

Server power in the objective function (Eq. 2.5) should be written as:

$$P_j^T = X_j^T \cdot (P_{j_{static}}^T + P_{j_{dyn}}^T) \tag{2.13}$$

however, this would introduce a non-linearity in the ILP problem (due to the product of variables $X_j^T \cdot U_{cpu_j}^T$ and $X_j^T \cdot U_{mem_j}^T$ in $P_{j_{dyn}}^T$). Constraint 5 avoids this issue as, when the number of VMs on j^{th} server (e_j^T) as well as the server CPU and memory utilization is zero, $X_j^T = 0$. On the contrary, if $1 \leq e_j \leq N_{VM}$, then $X_j^T = 1$. Therefore, it can be written: $X_j^T \cdot U_{cpu_j,n}^T = U_{cpu_j,n}^T$ and $X_j^T \cdot U_{mem_j,n}^T = U_{mem_j,n}^T$.

2.5.2 Data Communication Optimization

The amount of data exchanged between VMs directly impacts network traffic and response time. In practice, two VMs regularly exchange a varying amount of data. The goal is to minimize total data communication (network traffic, D_{total}^T) amongst the servers. In the formulation, $D_{j,n}^T$ represents the j^{th} server data communication; i.e., the amount of data transferred by a server at the n^{th} sample of the T^{th} time slot.

To express $D_{j,n}^T$, the binary variable $BinVMstatus_{j,k,l}^T$ indicates whether both k^{th} and l^{th} VMs have been allocated to j^{th} server ($BinVMstatus_{j,k,l}^T = 0$); otherwise, $BinVMstatus_{j,k,l}^T = 1$ in the T^{th} time slot. The matrix $VMdata_{k,l,n}^T$ contains the amount of data transferred from the k^{th} to l^{th} VM at the n^{th} sample during T^{th} time slot.

$$\min \quad D_{total}^T = \sum_{j=1}^{N_s} \sum_{n=1}^{N_t} D_{j,n}^T \tag{2.14}$$

where

$$D_{j,n}^T = \sum_{k=1}^{N_{VM}} \sum_{\substack{l=1 \\ l \neq k}}^{N_{VM}} [Place_{j,k}^T - (1 - BinVMstatus_{j,k,l}^T)] \cdot VMdata_{k,l,n}^T \tag{2.15}$$

subject to the following Constraints:

$$1.\ \sum_{j=1}^{N_s} Place_{j,k}^T = 1 \tag{2.16}$$

$$2.\ U_{cpu_j,n}^T \leq C^s \tag{2.17}$$

$$3.\ U_{mem_j,n}^T \leq C^m \tag{2.18}$$

$$4.\ VMstatus_{j,k,l}^T = (1 - Place_{j,k}^T) + (1 - Place_{j,l}^T) \tag{2.19}$$

$$5.\ BinVMstatus_{j,k,l}^T \leq VMstatus_{j,k,l}^T \leq 2 \cdot BinVMstatus_{j,k,l}^T \tag{2.20}$$

The constraints of the problem are formulated in Eq. 2.16 to 2.20, as follows. Constraints 1, 2 and 3 are the same as those of ILP-Power. Constraint 4 determines the status of the k^{th} and l^{th} VMs on the j^{th} server. The variable $VMstatus_{j,k,l}^T$ is used to specify the status of any pair of VMs based on the status of each VM on j^{th} server. As this variable is the sum of two binary variables, it can take only three different values (i.e., 0, 1, and 2): i) the k^{th} and l^{th} VMs are

allocated to the j^{th} server ($Place^T_{j,k} = 1$ & $Place^T_{j,l} = 1$), then $VMstatus^T_{j,k,l} = 0$; or ii) either the k^{th} or the l^{th} VM is allocated to the j^{th} server ($Place^T_{j,k} = 1$ & $Place^T_{j,l} = 0$ or vice versa), then $VMstatus^T_{j,k,l} = 1$; or iii) neither the k^{th} and the l^{th} VMs are allocated to the j^{th} server ($Place^T_{j,k} = 0$ & $Place^T_{j,l} = 0$), $VMstatus^T_{j,k,l} = 2$.

The original data communication objective is written as:

$$D_{j,n} = \sum_{k=1}^{N_{VM}} Place_{j,k} \cdot \sum_{l=1,\ l\neq k}^{N_{VM}} (1 - Place_{j,l}) \cdot VMdata_{k,l,n} \tag{2.21}$$

To remove the non-linearity in the equation, constraint 5 is used to compute per-server data communication and demonstrates that, if $VMstatus_{j,k,l} = 0$, then $BinVMstatus_{j,k,l} = 0$, and '1' otherwise. In other words, if both VMs are allocated to the same server, then $BinVMstatus_{j,k,l} = 0$.

2.6 Proposed Two-Phase Greedy Heuristic Method

To address the defined problem, I propose a two-phase greedy heuristic algorithm (Heuristic, in what follows) that, at each time slot T, jointly minimizes power consumption –P^T_s– and network traffic –D^T_{total}– (Phase 1), and then allocates resulting traffic in a network topology-aware fashion (Phase 2).

2.6.1 Phase 1 - VM Clustering

I split this phase in two steps and use a method similar to the one presented in [104]. First, at time slot T, all VMs available in the system are represented as points in a two dimensional (2D) plane. Based on the data and CPU-load correlation properties, as highly data-correlated VMs should be clustered together while highly CPU-load correlated VMs should be placed apart, a function is defined to calculate attraction and repulsion forces between each two VMs. Nonetheless, differently from the original algorithm [104], the attraction force is calculated as a worst-case peak bidirectional data exchanged between each two VMs during the time slot. Similarly, the repulsion force is computed as a worst-case peak CPU utilization when the peaks of two VMs coincide during the last time slot. As a result, the points are remapped in the 2D plane with new coordinates based on the computed forces.

In the second step, after finding the final position of the VMs, the minimum number of clusters (i.e., servers), $\hat{N}_{server}$ is determined, as follows:

$$\begin{aligned} \hat{N}_{server} &= max\{\hat{N}^{cpu}_{server}, \hat{N}^{mem}_{server}\} \\ \begin{cases} \hat{N}^{cpu}_{server} = max_n(\sum_{k=1}^{N_{VM}} VMcpu^{T-1}_{k,n}/C^s) \\ \hat{N}^{mem}_{server} = max_n(\sum_{k=1}^{N_{VM}} VMmem^{T-1}_{k,n}/C^m) \end{cases} \end{aligned} \tag{2.22}$$

where $\hat{N}_{server}^{cpu}$ and $\hat{N}_{server}^{mem}$ denote the minimum number of servers needed to comply with the VMs' CPU and memory utilization requirements, respectively. Hence, $\hat{N}_{server}$ is equal to the minimum number of servers to accommodate all the VMs while satisfying both the VMs CPU and memory utilization.

Then, a modified version of the k-means algorithm [104] is utilized to cluster VMs with respect to the distance between two VMs obtained from the repulsion and attraction phase in the 2D plane. Differently from the original k-means algorithm, I define a cap per cluster (i.e., C^s and C^m) when considering the VMs' CPU and memory utilization. Moreover, the initial centroid of clusters are not set randomly, but instead calculated based on the last position of points available in the previous time slot. I start with the minimum number of clusters, $\hat{N}_{server}$, to allocate the VMs to the clusters with shortest distance. If unfeasible, I increment the number of clusters by one. The process is iterated until all VMs are allocated to the minimum number of possible clusters. The proposed method guarantees that the total load of each cluster at each sample does not exceed C^s and C^m during the time slot. However, violation occurs due to miss-predictions, leading to delays in workload execution and, eventually, to their execution in the next time slot (with 100% prediction accuracy, no violation occurs).

2.6.2 Phase 2 - Clusters Allocation

In this phase, the clusters are allocated to the appropriate servers considering the data center network structure as described in Algorithm 1. This algorithm fills up the racks one by one, reducing the number of active switches, and minimizing network power, while keeping highly-communicating servers close to each other.

The output of the modified k-means creates an edge-weighted data communication graph ($G_{data} = \{V, E\}$), where set V and $W(E)$ represent the clusters and the amount of data transferred across clusters, respectively. The algorithm first selects the maximum edge weight ($Edge^{max}$) and initializes the amount of traffic transferred through all aggregation-layer switches (NT_{agr}^{h}) for different groups of racks (lines 1 and 2). Then, I select the first rack and try to fill it up. For the selected rack (r^{th} rack), I first initialize its ToR switch traffic (NT_{tor}^{r}) and the number of unused servers with the total number of servers available in r^{th} rack (lines 4 and 5). Then, for clusters related to the selected edge (lines 6 ~ 25) if either: i) two clusters have not been allocated yet and the number of unallocated clusters is less than the unused servers in the rack, ii) NT_{tor}^{r} of the new selected clusters is less than the B_{tor}, and iii) NT_{agr}^{h} related to the selected rack is less than B_{agr}, I allocate clusters to the servers in that rack. I also update the NT_{tor}^{r}, NT_{agr}^{h}, and the number of unused servers of the r^{th} rack (lines 7 ~ 17). After allocation, I combine clusters V_w and V_z and update $W(E)$ (lines 18 and 19). It is repeated until violating those conditions for the rack, and then the next rack is selected. This algorithm iterates until all clusters are allocated to physical servers, which guarantees that the bandwidth of switches is not exceeded.

Algorithm 1 Cluster Allocation

Input: Network topology and data communication graph ($G_{data} = \{V, E\}$)
V = clusters & $W(E)$ = data communication between clusters
Output: Cluster allocation
$Edge^{max} \leftarrow$ Max. edge ($W(E_{w,z}) + W(E_{z,w})$) between any V_w and V_z
$NT^{h}_{agr} \leftarrow 0$ Initial network traffic of h^{th} agr. switch
for $r = 1$: Total racks **do**
  $NT^{r}_{tor} \leftarrow 0$ Initial ToR switch network traffic of r^{th} rack
  Unused servers of r^{th} rack $\leftarrow$ Total servers of r^{th} rack
  while (Selected clusters $\leq$ Unused servers of r^{th} rack) & ($NT^{r}_{tor} \leq B_{tor}$) & ($NT^{h}_{agr} \leq B_{agr}$) **do**
    if V_w and V_z have not been allocated **then**
      Allocate clusters w and z to two servers in r^{th} rack
      $NT^{r}_{tor} \leftarrow$ Update traffic of servers of r^{th} rack
      $NT^{h}_{agr} \leftarrow$ Update traffic of racks of h^{th} group
      Update unused servers in r^{th} rack
    else if V_w or V_z has not been allocated **then**
      Allocate w or z to one server in r^{th} rack
      $NT^{r}_{tor} \leftarrow$ Update traffic of servers of r^{th} rack
      $NT^{h}_{agr} \leftarrow$ Update traffic of racks of h^{th} group
      Update unused servers in r^{th} rack
    end if
    Combine V_w and V_z
    Update $W(E)$
    if All clusters allocated **then**
      Terminate
    end if
    $Edge^{max} \leftarrow$ Find maximum edge weight
    Find number of selected clusters ('1' or '2') when both clusters have not been allocated
  end while
end for

2.7 Proposed Machine Learning (ML) Method

This section describes a two-step ML approach to allocate VMs to servers. First, I generate offline different classes using k-means according to the features extracted from the VMs' CPU utilization traces. To decide the appropriate number of classes (K), a heuristic-based process is used, as explained in Section 2.7.1. Second, at run-time, I classify VMs into classes by determining the shortest euclidean distance to each class centroid, and then I use the *value iteration* algorithm, amongst the various RL methods, to allocate the VMs to physical servers. RL is particularly useful in problems with large state and/or action spaces that change dynamically over time and depend on the environment [77, 105]. I use the first week of Bitbrains traces for class generation and for the exploration phase of the ML approach, and the second week of traces for run-time VMs classification and exploitation phase.

2.7.1 Class Generation – Offline Pattern Detection

Class generation significantly simplifies the process of VM allocation, reducing the complexity of the value iteration algorithm. In the Bitbrains traces, a high-variability is observed but also a daily periodicity in the CPU utilization traces, making them suitable for classification. As memory resources are more over-provisioned than CPU resources, and less critical, the class generation only takes into account the CPU utilization traces to generate classes. In this step, based on the time slot duration (i.e., 1 hour), each VM's CPU utilization trace per time slot is considered as an individual pattern composed of 12 samples (1 every 5 minutes).

Feature extraction is a key point that greatly affects classification results. Hence, a list of features is selected as:

- Maximum and minimum CPU utilization, to represent the range of variation and the absolute value of the traces.
- Time at which the maximum utilization happens, to enable CPU-load correlation techniques.
- Variance, as it shows the trace variability.
- Median, to account for typical values.
- Skewness, which is a measure of the trace asymmetry.
- Kurtosis, which provides an insight on the trace shape.

Then, the k-means method is used to classify CPU utilization patterns [106]. As k-means does not decide on the number of classes (K), the following heuristic is proposed.

2.7.1.1 Heuristic-Based Process for Determining The Appropriate Number of Classes (K)

First, I define a similarity score ($\phi^T_{k,l}$ in Eq. 2.23) that expresses the similarity between any pair of VMs, VM_k and VM_l during time slot T. To define this metric, the Pearson Correlation ($\rho^T_{k,l}$) is used, which is effective to judge the similarity on the shape of the traces. However, as the Pearson Correlation cannot reflect the absolute CPU utilization value, the euclidean distance ($Dist^T_{k,l}$) over all the samples is incorporated into the metric. As a result, Eq. 2.23 demonstrates that $\phi^T_{k,l}$ is high when two traces have both the same shape and CPU utilization absolute value. Since two VM traces may be totally the same, $(Dist^T_{k,l}+1)$ is considered in the denominator to avoid having infinite value when $Dist^T_{k,l}=0$, and the values are normalized to $[0,1]$.

$$\begin{aligned} \phi^T_{k,l} &= \frac{\rho^T_{k,l}}{Dist^T_{k,l}+1} \\ Dist^T_{k,l} &= \|VMcpu^T_k - VMcpu^T_l\|_2 \end{aligned} \tag{2.23}$$

Second, for each time slot in one week, the VMs are classified into K different classes according to their euclidean distance (exhaustively testing different values of K). After classification, I compute the per-class similarity score during each time slot ($\phi^{class}_{\omega,T}$), as calculated by Eq. 2.24, where N_ω is the number of VMs available in class ω, obtaining K different scores (i.e., as many as classes). I average these K scores to obtain an average similarity score ($\phi^{class}_{avg,T}$) per T.

$$\phi^{class}_{\omega,T} = \frac{\sum_{k=1}^{N_\omega} \sum_{l=1,\ l\neq k}^{N_\omega} \phi^{T}_{k,l}}{N_\omega * (N_\omega - 1)} \tag{2.24}$$

For instance, Fig. 2.4 shows the average similarity score of $\phi^{class}_{avg,T}$ among all time slots under the different number of selected classes (K). By increasing the number of classes, the similarity score exhibits a logarithmic growth, achieving its highest value of '1', when there are as many classes as patterns. As the number of classes directly impacts the execution time of the ML algorithm, but a high similarity is needed to achieve good accuracy, a similarity score of 0.5 is heuristically chosen, which leads to 150 classes (K).

Because managing 150 classes is still unfeasible for the ML algorithm, I analyzed the amount of traces in each class. I found that most classes contained few traces and, therefore, Algorithm 2 is proposed to reduce the number of classes by combining their centroids. Centroid combination is as follows: I find the two classes ($\omega 1$ and $\omega 2$) with minimum euclidean distance ($Dist(\omega 1, \omega 2)$). Then, these two classes are combined such that the centroid of the new class ($Cntrd_{\omega 1,\omega 2}$) is the mean of centroids of the original classes. The algorithm iterates until reaching a maximum number of iterations (N_{itr}). N_{itr} is heuristically selected based on the number of classes that are below a given distance. After combination, the number of VMs available in each class is computed. If the number of VMs in the i^{th} class per time slot ($Num^{VM}_{a,i}$) and the maximum among all time slots ($Num^{VM}_{b,i}$) are less than TH_a and TH_b, respectively, then that class is deleted. Thus, classes are only kept that have a certain amount

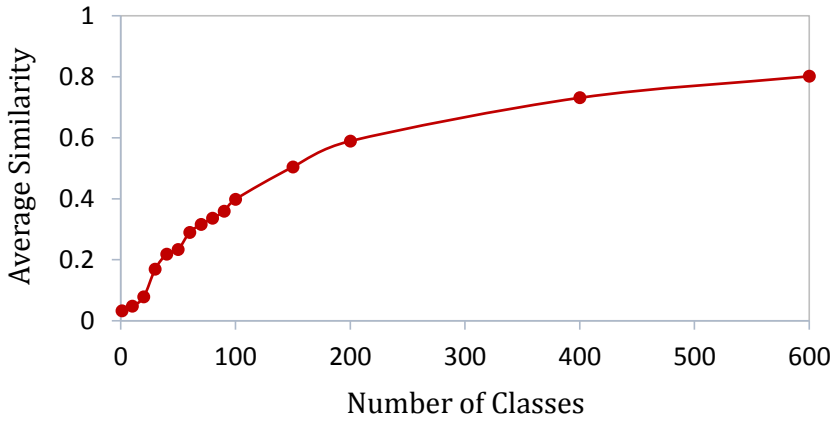

Figure 2.4 – Average similarity of classes under different number of classes among all time slots in one week.

Algorithm 2 Class Combination and Deletion

Input: Number of iterations (N_{itr}), centroids of classes ($Cntrd$) and number of classes (N_{class})
Output: Centroids of new classes after combination and deletion

$M^{Dist} \leftarrow$ Matrix of euclidean distances between centroids
for $i = 1 : N_{itr}$ **do**
  $Dist(\omega1, \omega2) \leftarrow \min(M^{Dist})$
  $Cntrd_{\omega1,\omega2} \leftarrow \text{mean}(Cntrd_{\omega1}, Cntrd_{\omega2})$
  Update $Cntrd$, N_{class} and M^{Dist}
end for
$Num_a^{VM} \leftarrow$ Number of VMs in each class per time slot
$Num_b^{VM} \leftarrow$ Max. VMs in each class among all time slots
for $i = 1 : N_{class}$ **do**
  if $(Num_{a,i}^{VM} < TH_a)$ & $(Num_{b,i}^{VM} < TH_b)$ **then**
    Delete class i
  end if
end for

of VMs and are useful to avoid degrading the quality of ML algorithm. By appropriately setting these thresholds, the proposed method allows to reduce the number of classes from 150 to 27, i.e. $K = 27$, without decreasing average similarity.

2.7.2 Run-Time Classification and Value Iteration Algorithm

At run-time, I use the second week of traces to classify VMs into the classes resulting from the previous step in each time slot. First, the last-value predictor is used to obtain the last VMs' patterns and extract their features. Then, the VM is assigned to the class which has the shortest euclidean distance to the centroid. Finally, I use a RL technique, the value-iteration algorithm, to allocate the VMs to physical servers. Typically, RL models are composed of an agent and an environment with a finite set of actions (A) and a state space (S). In the environment, the states are observed and the actions determined by agent are applied. The agent maps actions to states at any decision time. There is a reward function used for each state-action pair (R) which should be maximized by the agent. $R(s, a, s')$, thus, shows the immediate reward value obtained after performing action a in current state s, representing the next state (s') reward value [77, 107].

The proposed ML approach consists of two phases, namely exploration and exploitation, as shown in Fig. 2.5. If the current state has not been previously explored, i.e., in the exploration phase, and the agent randomly chooses new actions for the new states, and records the new states and rewards obtained from each action separately. On the contrary, if a state has already been explored, I proceed to the exploitation phase, and use the value iteration algorithm to find the best action among the pool of actions obtained during exploration to maximize the reward. The essential idea is: if the true value of each state is known, I would simply choose

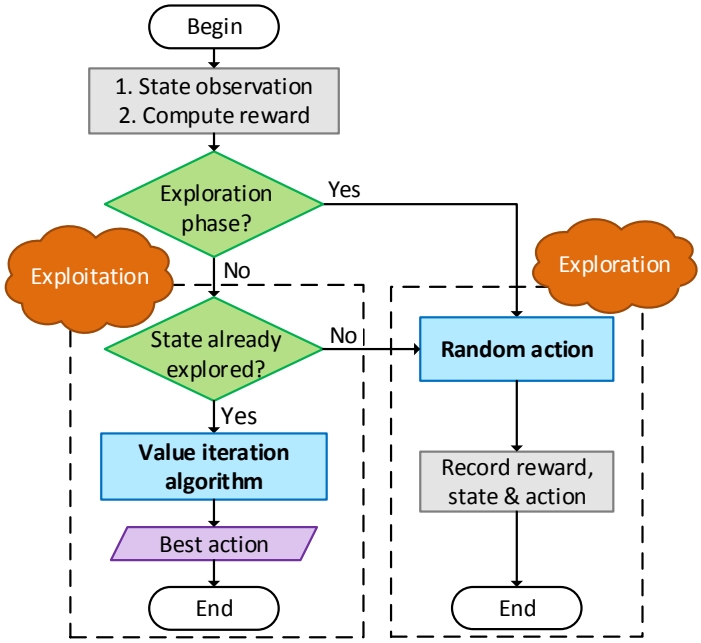

Figure 2.5 – The overall process of proposed ML approach.

the action that maximizes the expected reward. In the defined problem and case, at each communication between agent and environment, a set of actions is applied to environment (allocating only one VM to each server). Hence, the state's true value is not initially known; its immediate reward is only known. For example, a state might have low initial reward but be on the path to a high-reward state. Value-iteration progressively evaluates the states until the solution converges. Nonetheless, it assumes that the agent has a great knowledge about the reward of all states in the environment. Because in a consolidation problem the reward can be clearly stated (i.e., the higher the server utilization, the higher the reward is), the value-iteration method is suitable to solve this problem.

Because the value iteration algorithm uses a vector of actions, the amount of servers active per time slot needs to be defined before VM allocation. To this end, I first compute the maximum number of VMs than can be allocated to each server (N_{VM}^{server}) and the minimum number of needed servers, i.e., $\hat{N}_{server}$, per time slot. Then, the agent decides the VM placement by choosing which VM class should be allocated to each server. If the selected class has no VMs, it picks a VM from the class with a shortest euclidean distance to it. After applying the actions, I update the utilization of each server, set its state and reward, and send them back to the agent. This communication is iterated until all the VMs are allocated.

Finally, to minimize traffic through the different switches considering bandwidth constraints, the racks are filled up one after the other with the servers that have higher data communication (after allocation, the amount of data transferred between any pair of servers is sorted in descending order). The following subsections describe the state, actions, and the proposed reward function for the value-iteration algorithm.

Table 2.2 – State definition (s) and value per server

Parameter	Definition	Range
$P1$	Number of allocated VMs	$0 - N_{VM}^{server}$
$P2$	Utilization of server	$0 - 11$
$P3$	Active/inactive in time slot	$0/1$

2.7.2.1 State and Action Definitions

Each time slot, the agent provides one action per server to the environment. This means that $\hat{N}_{server}$ actions are applied to $\hat{N}_{server}$ servers. The number of communication steps in one time slot is the maximum number of VMs allowed to be allocated to one server, i.e. N_{VM}^{server}.

The state of each server is defined based on the number of VMs allocated to it and the server utilization. In general, three parameters are used to reflect one state value. Each parameter can get a discrete value in a finite range, as shown in Table 2.2. The first parameter, P1, indicates the number of VMs allocated to server. P2 exhibits the maximum aggregated CPU utilization of co-located VMs during the time slot. Server utilization is discretized (from 0 to 100%) in 11 uniform levels. The predicted server utilization is mapped to the nearest higher level to minimize violations due to miss-predictions. P3 is a binary parameter that shows that if the server is active.

The allocation of a VM from one class to a server is considered as an action. The first VM available from the selected class is chosen. If the class is empty, a VM is chosen from the non-empty class which has the minimum distance to that class.

2.7.2.2 Reward Function

Each pair of state-action reward (next state reward) is defined per server ($R_j(s,a,s')$ for j^{th} server; it is simply named R_j) according to two parameters: i) the gap between the current and the maximum utilization (i.e. utilization gap, R_g^j) and, ii) the amount of data transferred by the server (R_d^j) as:

$$R_j = R_g^j - \frac{R_d^j}{\lambda} \tag{2.25}$$

where λ is a weighting factor used to keep the reward factors in the same range.

Given that the consolidation technique is used as a strategy for power minimization, there are three different situations: i) fully utilized server (i.e., $= C^s$), ii) underutilized server ($< C^s$), and iii) overutilized server ($> C^s$). To minimize power consumption via consolidation, R_g^j needs to be highest when the server is fully utilized (1000 is chosen). Hence, it is enough to set lower reward values for the rest of situations. For underutilization situations, a positive proportional

range is chosen between 1 and $C^s/10$, i.e., the higher the server utilization, the higher the reward is. On the other hand, to minimize violations and QoS degradation, it is needed to avoid surpassing maximum utilization (C^s). For this purpose, due to the higher importance of server overutilization compared to underutilized server situations, it is enough to choose a lower value to decrease the server reward. Therefore, a negative value is set to utilization above C^s (-1 is chosen). In practice, any other negative value works, and gives the same results. Thus, R_g^j can be computed as follows:

$$R_g^j = \begin{cases} \frac{C^s}{C^s - \hat{U}^T_{cpu_j}} & \hat{U}^T_{cpu_j} < C^s \\ 1000 & \hat{U}^T_{cpu_j} = C^s \\ -1 & \hat{U}^T_{cpu_j} > C^s \end{cases} \tag{2.26}$$

where $\hat{U}^T_{cpu_j}$ represents the maximum utilization of j^{th} server among all the samples in the T^{th} time slot, i.e., $max_n(U^T_{cpu_j,n})$.

Similarly, R_d^j represents the total amount of data transferred by j^{th} server, as follows:

$$R_d^j = \sum_{n=1}^{N_t} D^T_{j,n} = \sum_{k=1}^{N_{VM}} \sum_{l=1}^{N_{VM}} \sum_{n=1}^{N_t} VMdata^T_{k,l,n} \quad VM_k \in server_j \;\&\; VM_l \notin server_j \tag{2.27}$$

The reward function is computed per-server, aiming to maximize server utilization while minimizing the amount of data should be exchanged between servers.

2.8 Proposed Hyper-Heuristic Method

In this section I present a hyper-heuristic algorithm to dynamically determine which method, among Heuristic and ML, should be used at each time slot T to achieve a specific trade-off across the different objectives. The proposed hyper-heuristic relies on the long-term periodicity of the workloads being executed, and learns the performance of the methods over time. The considered trade-offs (objective set $\mathbb{O}$) are power consumption (P_{DC}), worst-case server overutilization (WCV), and total network traffic of ToR (TN_{tor}), aggregation- (TN_{agr}), and core-layer (TN_{cr}) switches, as the most important metrics from the data center providers perspective. They are computed for each method i in a cost function (*CostFunction*($\mathbb{O}_i$)) as:

$$Cost_i = \alpha_1 P_{DC} + \alpha_2 WCV + \alpha_3(\beta_1 TN_{tor} + \beta_2 TN_{agr} + \beta_3 TN_{cr}) \quad \sum_{j=1}^{3} \alpha_j = 1 \;\&\; \sum_{k=1}^{3} \beta_k = 1 \tag{2.28}$$

where α_j and β_k are user-defined weighting factors that need to be set with respect to the importance that the user gives to a specific objective (each normalized to (0,1]), and whose value can be changed during run-time. For the sake of clarity, the same priority has been given to all objectives (i.e., $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$). Due to the communication distance, higher

weight is considered for upper network layers ($\beta_1 = 0.1$, $\beta_2 = 0.3$, and $\beta_3 = 0.6$). Thus, the lower the power consumption, overutilization and network traffic, the lower the cost value is. The proposed algorithm is as follows.

At the beginning of each time slot, one of the methods in the pool of candidate methods ($\mathbb{M}$) (i.e., Heuristic and ML in my case) is selected. To choose which algorithm to be executed, the proposed hyper-heuristic builds a history of the performance of the Heuristic and ML methods by using the cost function (Eq. 2.28). As described in Algorithm 3, at the beginning of time slot T, I create a hash code of the previous execution ($T-1$) that is stored in a hash table ($HashTable$). For the first time slot, $HashTable$ is empty and one of the methods is randomly selected. The hash is generated by using the function *HashGenerator* (line 1) that creates a binary string with the length of the selected objectives ($\mathbb{O}$), in which each character is '1' if the ML performed better in that objective than the Heuristic, and is '0' otherwise. For example, hash code "11011" shows that ML has the best results for four objectives with respect to Heuristic. In $HashTable$, for each observed hash, two entries per method are also stored including the cost value ($Cost_i^{Hash}$, $i \in \mathbb{M}$) and how many times that method has been selected in the past (Num_i^{Hash}, $i \in \mathbb{M}$). After generating the hash, the algorithm checks in $HashTable$ whether the hash had been already observed. If it exists, I select the method with the minimum $Cost_i^{Hash}/Num_i^{Hash}$ (line 4). Otherwise, I select the method with minimum cost value for $T-1$, and the observed hash is only recorded (lines 6 and 7). The hash entries are updated, i.e., $Cost^{Hash}$ and Num^{Hash}, at the end of T as follows.

After executing the selected method, at the end of time slot T, the results ($\mathbb{O}^T$) per method are collected (lines 9 and 10). Then, the entries of the method are just updated with the minimum

Algorithm 3 Hyper-Heuristic Algorithm

Input: $\mathbb{O}_i^{T-1} = \{P_{DC}^{T-1}, WCV^{T-1}, TN_{tor}^{T-1}, TN_{agr}^{T-1}, TN_{cr}^{T-1}\}$, $i \in \mathbb{M}$
$HashTable$
Output: Select one method from $\mathbb{M}$
1: $Hash \leftarrow$ HashGenerator($\mathbb{O}_{\mathbb{M}}^{T-1}$)
2: $HashObserved \leftarrow$ IsHashObserverd($Hash$, $HashTable$)
3: **if** $HashObserved == True$ **then**
4: $m \leftarrow$ Select method with $min(Cost_i^{Hash}/Num_i^{Hash})$, $i \in \mathbb{M}$
5: **else if** $HashObserved == False$ **then**
6: Record $Hash$ in $HashTable$
7: $m \leftarrow$ Select method with minimum CostFunction($\mathbb{O}^{T-1}$)
8: **end if**
9: Execute method m for time slot T
10: Obtain $\mathbb{O}_i^T$ at the end of time slot T, $\forall i$
11: $Cost_i \leftarrow$ CostFunction($\mathbb{O}_i^T$), $\forall i$
12: $m \leftarrow$ Find method with $min(Cost)$
13: $Cost_m^{Hash} \leftarrow Cost_m^{Hash} + Cost_m$
14: $Num_m^{Hash} \leftarrow Num_m^{Hash} + 1$

cost value for corresponding hash (lines 13 and 14).

2.9 Experimental Setup and Scenarios

In this section I present the experimental setup and introduce two scenarios to compare the proposed methods.

2.9.1 Experimental Setup

2.9.1.1 Data Center Configuration

I consider two rows of racks in the data center. Each row consists of four 42U racks, and each rack has ten servers. I target Intel S2600GZ servers consisting of 6-core CPU (Intel E5-2620), 9 frequency levels varying from 1.3 to 2.4GHz, and 32GB of memory.

The server power consumption is modeled as in Section 2.3.2; each server consumes constant $16W$ and $27.2W$ for disk (P_{disk}) and cooling fan (P_{fan}), respectively. A high fan speed ($8000rpm$) and a low inlet temperature ($22°C$) are considered to reduce the effect of temperature-dependent leakage power. Under this condition, leakage power (P_{cpu}^{leak}) is almost constant and $3.1W$ in the worst-case. Idle power for CPU (P_{cpu}^{idle}) and memory (P_{mem}^{idle}) are $50W$ and $4W$, respectively. The dynamic power of CPU (P_{cpu}^{dyn}) and memory (P_{mem}^{dyn}) are $42.5W$ and $56W$ at 100% utilization, respectively [96].

A three-layered tree network topology is considered. The types of ToR, aggregation-layer switches and the core router are HP5920 with $60GBps$ bandwidth (B_{tor}), HP6600 with $180GBps$ bandwidth (B_{agr}), and HP8800 with $430GBps$ bandwidth (B_{cr}) that dissipate $366W$, $405W$ and $3500W$, respectively [79]. For cooling power consumption, a time-varying PUE model is used ranging from 1.25 to 1.55, as presented in previous work [100].

2.9.1.2 Simulation Framework

To simulate a realistic scenario, the VMs' CPU and memory traces of Bitbrains have been utilized for a time horizon of two weeks [101]. I used the first week of traces for class generation and exploration phase, and the second week for VMs classification, exploitation phase of ML and for evaluating all the methods. The validated power model of Section 2.3.2 has also been used to compute data center power consumption by exploiting an in-house simulator tool written in C++, where all the algorithms used in this chapter have been implemented.

The VM allocation and the frequency updating (t_f) are invoked every 1 hour, whereas the cooling system update (t_c) is invoked every 10 minutes. For each experiment, different number of VMs (from 50 to 1000) has been considered in data center.

Data communication between a pair of VMs is modeled by a log-normal distribution [102].

As 80% of the VMs have 800 kB/min traffic among each other while 4% of the VMs have 10 times higher traffic [103], a log-normal distribution is tuned with a mean of 800 and uniform variance in the range of [1,4] for each time slot. The amount of data communication between VMs varies every 5 minutes during the one-hour time slot.

2.9.1.3 Simulation Environment

The proposed methods are carried out on a separate server equipped with a 24-core Intel CPU@1.60GHz and 50GB of memory. To solve the ILP, I used the CPLEX 12.3 solver available in GAMS 23.7 [108].

2.9.2 Scenarios

2.9.2.1 Scenario I - Optimality Assessment

To evaluate the efficiency of the proposed methods I compare them to the ILP-based methods (optimal solutions), for a few number of VMs (50, 100 and 150) and a time horizon of one day. I also take advantage of the fact that in multi-service cloud scenarios, data exchange only occurs between the tenants (VMs) of each service. In other words, the VMs can be grouped together that exchange data between each other, while these groups are isolated and do not share data to other groups. In this scenario I assume a number of groups equal to 20% of the available VMs in the data center. The VMs are uniformly distributed to groups, limiting the number of VMs per group (group size) to 10. Different traffic (data communication) are generated between VMs of the same group for each sample during one time slot. The VMs are also redistributed per time slot under the group size limitation. For instance, for the 50 VM scenario 10 groups of sizes between 1-10 are generated, and ensuring that each VM is assigned to one group. Moreover, to fairly compare ILP method to other approaches, the total data communication among the servers (total network traffic) is computed for all the methods regardless of network topology constraints.

2.9.2.2 Scenario II - Comparison Heuristic, ML and Hyper-Heuristic in Large-Scale Scenarios

Communication patterns contain a wide range of variations from one-to-one to all-to-all traffic between VMs [16]. As opposed to Scenario I, in application-specific private data centers, a high number of VMs communicate with each other, e.g., bank transactions between any two customers. Thus, in this scenario a more general data communication pattern between VMs is considered, assuming that half of the VMs in the data center communicate with each other. The communicating VMs are randomly selected using a uniform distribution. Then, each selected VM is set to exchange data with any other 50% of the VMs, also selected according to a uniform distribution. I analyze the network traffic through different layers as it is considered in the heuristic, ML, and hyper-heuristic methods. Moreover, the number of VMs is increased

from 200 to 1000, to compare these methods in a large-scale scenario for a time horizon of one week.

2.10 Results - Optimality Assessment (Scenario I)

In this section I first evaluate the effectiveness of consolidation strategy compared to load balancing on real hardware. Then, I compare the total energy consumption of data center, QoS (violations caused by overutilized servers), network traffic, number of migrations, and execution time of the algorithms for eight different state-of-the-art methods:

- Correlation-aware VM Allocation (CVMA) [20] that is the best in its class to optimize energy consumption.
- Network-aware VM allocation (GH) presented in [16] for network traffic minimization.
- Heuristic: the proposed heuristic (Sect. 2.6).
- Heuristic-Cap: for a realistic and fair comparison with ILP-based methods, the servers capacity is reduced to 80% during the VM allocation phase to guarantee that no violation occurs due to miss-prediction. This selected cap empirically represents a trade-off between energy efficiency and violation compared to Heuristic. It is also assumed that all the active servers use the maximum frequency level; i.e. 2.4GHz (100%), to compute violations.
- ML: the proposed ML algorithm (Sect. 2.7).
- ML-Cap: the proposed ML algorithm with 80% servers capacity cap and setting maximum frequency level.
- ILP-Power: the proposed ILP-based method for data center energy optimization (Sect. 2.5.1).
- ILP-Data: the proposed ILP-based method for data communication optimization (Sect. 2.5.2).

2.10.1 Consolidation Technique Efficiency on x86 Servers

In this section I show the benefits of consolidation strategy compared to load balancing in terms of power and energy consumption on x86 platform. For this evaluation, the Xeon E5 v4 platform [109] is used for our target workloads. This processor includes an 8-core CPU with 2 threads per core (i.e., 16 virtual CPUs - vCPUs). I consider two main contributors to the overall power consumption of the server: i) the whole CPU package, and ii) DRAM as main memory of the server system. Concerning the CPU and memory power measurements, I use the information extracted using the Running Average Power Limit (RAPL) interface.

Table 2.3 – Energy efficiency of the consolidation versus load balancing technique

Technique	Power Consumption (W)	Application Execution Time (sec)	Energy Consumption (kJ)
Consolidation	87	136	11.83
Load Balancing	161	132	21.25

To characterize the power and performance of these techniques, I make use of synthetic workloads that are representative of real banking applications according to our industrial partners in this work, namely, Credit Suisse S.A. and Eaton Corporation. For the Consolidation scenario, 16 applications with the same characteristics are executed on the cores to maximize the CPU utilization and 1GB memory stress per application. On the other hand, for the Load Balancing scenario, the applications are distributed to two servers such that each server reaches and stabilizes on a 50% CPU utilization.

Table 2.3 shows the energy consumption and application execution time for both techniques. As this table shows, Consolidation obtains 46% power improvements with only 3% performance overhead compared to Load Balancing due to the reduction of the number of active servers and subsequent static power. In the following sections, the proposed methods, which elaborate on the basis of the Consolidation technique, are assessed at data center level.

2.10.2 Energy Efficiency Analysis

Figure 2.6 shows the energy consumption breakdown of the data center including both IT and cooling components. As a result of turning off more servers, all approaches show an overall energy improvement higher than ILP-Data; on the contrary, ILP-Data further reduces (up to 66%) network energy, as it turns off switches for longer periods of time. Heuristic and ML exhibit less than 2% energy savings compared to CVMA; while providing 10% and 9% improvements compared to GH, respectively. This is because CVMA only considers CPU-load correlation between VMs; but, GH allocates the VMs with high data correlations to fewer servers. On the other hand, ILP-Power only improves up to 5% the results of ML by optimizing the number of active servers. In general, Heuristic-Cap and ML-Cap lead to higher energy consumption, due to the conservative capping approach, that increases the number of used servers.

2.10.3 QoS - Analysis of Violations

Figure 2.7 (right y-axis) shows the total number of violations, defined as the number of overutilized servers during one day. Heuristic provides a drastic reduction of the violations, from 30% to 87% in worst and best cases compared to ML, respectively. This is because, in the ML approach for a low number of VMs, most of the classes are empty after classification. Therefore, to fill up the servers, ML chooses one VM from the nearest non-empty class, decreasing

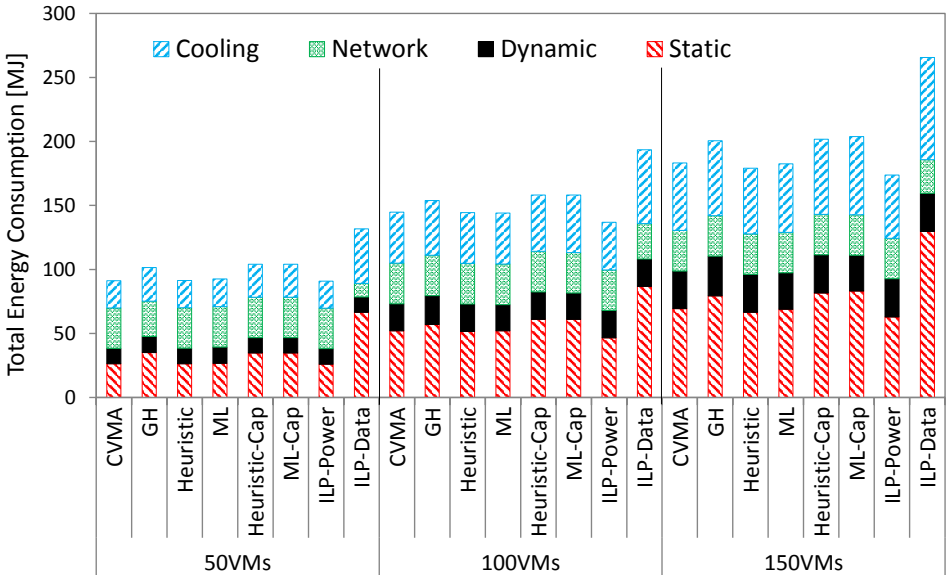

Figure 2.6 – Energy consumed by the data center for one day.

classification accuracy and leading to violation. On the other hand, Heuristic and ML achieve 94% and 65% improvements compared to CVMA, respectively. GH drastically decreases the number of violations in comparison with the other approaches due to partially filling up the servers. Due to the nature of the ILP-Power, ILP-Data, Heuristic-Cap and ML-Cap, these methods do not present any violation. Thus, they are not shown in Fig. 2.7.

Figure 2.7 (left y-axis) shows the average and worst-case amount of overutilized servers for a time horizon of one day, which determines the degree by which the negotiated QoS require-

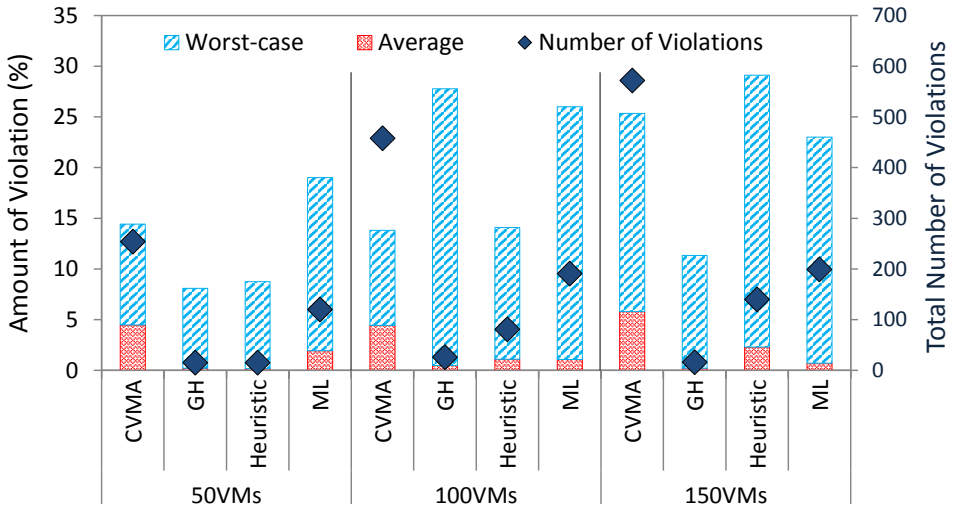

Figure 2.7 – Average, worst-case percentage amount and total number of violations for one day.

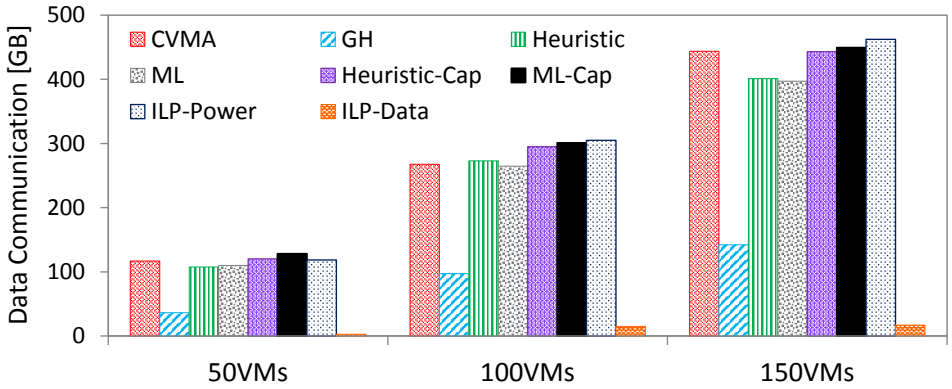

Figure 2.8 – Total amount of data exchanged among the servers for one day.

ments can be violated. Basically, quality degradation is observed due to the miss-predictions, especially during abrupt workload changes. The results show, for lower number of VMs, Heuristic and CVMA provide better worst-case violation reduction compared to ML. But, for higher number of VMs, the violation of ML remains below the violation of Heuristic and CVMA because classification accuracy is improved. As a result, 70% and 19% less violations are obtained on average and in the worst case for ML compared to Heuristic, and also 88% and 10% improvements compared to CVMA, respectively, for 150 VMs. This figure also shows that GH obtains better results compared to the other approaches due to the servers' underutilizations.

2.10.4 Network Traffic Analysis - Data Communication

Figure 2.8 shows total amount of data exchanged among the servers. Results demonstrate that ILP-Data (optimal solution) reduces the traffic ≈41, 42 and 46x in best case and ≈19, 17 and 21x in the worst case compared to Heuristic, ML and ILP-Power, respectively. This is due to the fact that ILP-Data distributes the non-communicating groups of VMs to different servers to minimize traffic. However, the number of turned-on servers is increased, leading to higher energy consumption. GH achieves up to 3x less network traffic compared to Heuristic and ML since its goal is to minimize the network traffic. On the contrary, the other approaches first try to use the minimum number of servers and then minimize the data communication.

In this sense, heuristic and ML increase the capability of absorbing time-varying data communication between the servers compared to ILP-Power and CVMA. These results show up to 14%, 11% and 3% improvements for ML compared to ILP-Power, CVMA and Heuristic, respectively.

2.10.5 Evaluating The Number of Migrations

Table 2.4 shows the total number of migrations for one day. CVMA reduces the number of migrations compared to other methods since this considers only CPU-load correlation. On the

Table 2.4 – Total number of migrations for one day

Method	50VMs	100VMs	150VMs
CVMA	429	1271	2180
GH	868	2038	3192
Heuristic	715	1862	3003
ML	619	1410	2251
Heuristic-Cap	791	1914	3180
ML-Cap	659	1561	2464
ILP-Power	888	2160	3120
ILP-Data	936	2352	3528

contrary, ILP-Power and ILP-Data have the highest number of migrations in order to find the optimal allocation solutions. In the best case, ML obtains up to 31% and 25% improvements in the number of migrations compared to GH and Heuristic, respectively, due to the trace classification strategy. Moreover, ML-Cap outperforms Heuristic-Cap by up to 23%.

2.10.6 Execution Time of Proposed Algorithms

The proposed methods trade-off solution optimality by execution time. To obtain the results shown in Table 2.5, I run the VM allocation methods for all time slots in 1 day, and its average is computed. The execution time of ILP-based methods is the highest (> 2 hours in some cases), making run-time allocation unfeasible. On the other hand, ML is the fastest algorithm (<10 ms in the worst case) making it particularly suitable to solve large-scale problems.

2.11 Results - Large-Scale Scenario (Scenario II)

In this section, I first assess the optimality of the proposed hyper-heuristic method. Then, I show, for the same metrics than in the previous case, a comparison between the heuristic, ML, hyper-heuristic (Hyper) methods, and the state-of-the-art methods. In addition, the proposed methods are compared to the Load Balancing (LB) strategy that aims to spread VMs across

Table 2.5 – Execution time (sec.) of the algorithms

Method	50VMs	100VMs	150VMs
CVMA	0.19	0.93	2.49
GH	0.005	0.026	0.073
Heuristics	0.969	3.907	12.897
MLs	0.003	0.005	0.009
ILPs	10.087	287.694	8619.48

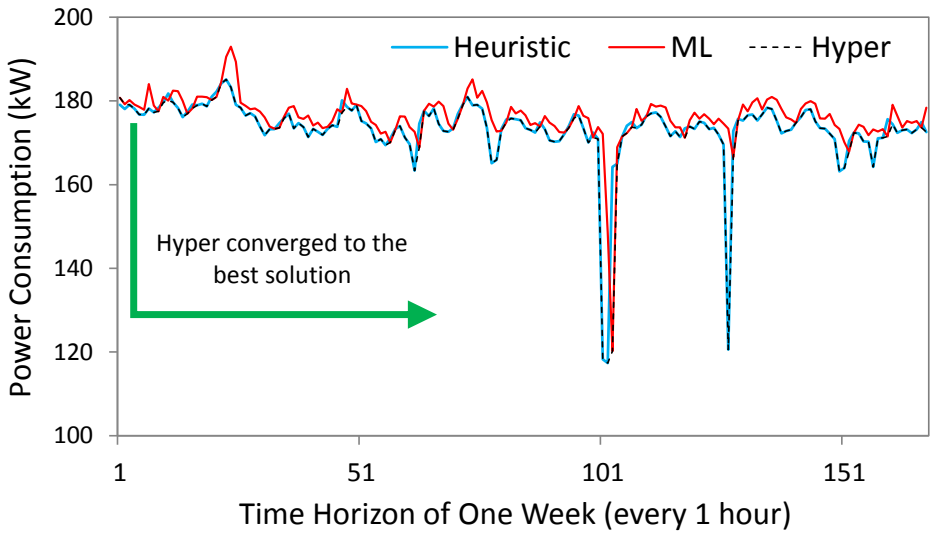

Figure 2.9 – Proposed hyper-heuristic method performance evaluation in terms of power consumption with 1000 VMs for a time horizon of one week.

servers, reaching an average server utilization close to 50%, which is a typical scenario in today's data centers when the servers are busy for half of the processing time [69].

2.11.1 Hyper-Heuristic Performance Evaluation

In order to evaluate the performance of hyper-heuristic method, I consider one objective, i.e., power consumption, to show that Hyper can select the best method among Heuristic and ML with the minimum total power consumption per time slot using an oracle predictor for VMs' loads. In particular, Fig. 2.9 shows that when there is no information about the history of the performance of the Heuristic and ML methods (i.e., hash table is empty), Hyper may not select the best method. However, it converges to the best method after few time slots when a history of power consumption of the both methods is stored in the hash table.

In real scenario, the data center providers deal with a multi-objective optimization problem. In addition, the VMs characterizations are not known at the beginning of the time slot. Therefore, I assess the proposed hyper-heuristic method in real and dynamic environment with the same metrics used in the previous case in the following sections.

2.11.2 Energy Efficiency Analysis

Figure 2.10 shows that heuristic, ML and Hyper reach almost similar results for energy consumption (< 2%). Hyper provides better energy savings compared to ML by selecting the Heuristic method in some time slots where Heuristic dramatically outperforms ML in terms of energy consumption. We observe up to 35%, 34%, and 51% energy improvements for the

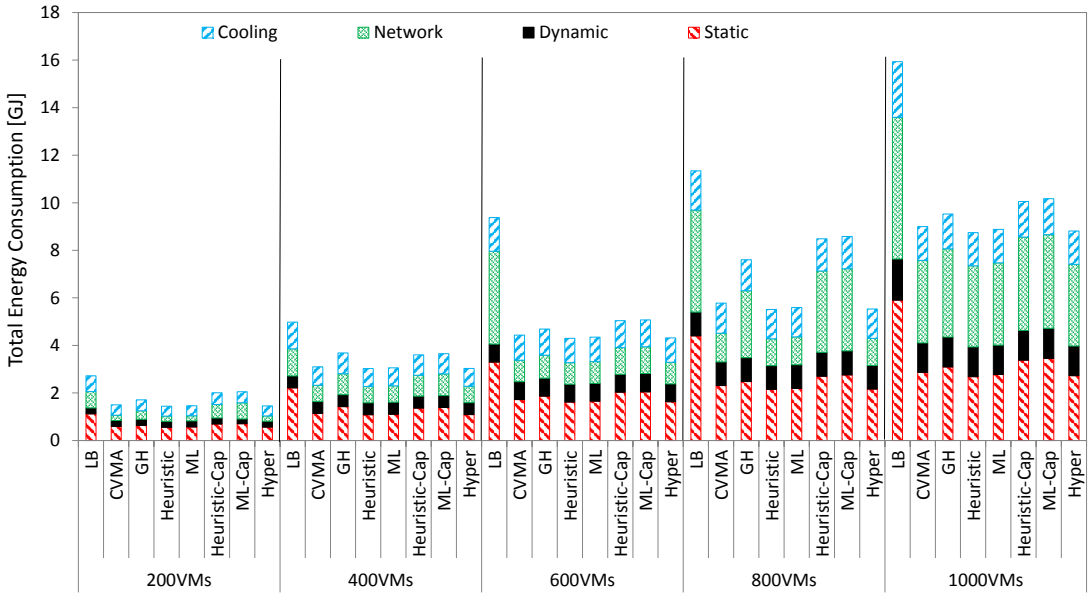

Figure 2.10 – Energy consumed by data center for one week.

proposed Heuristic and ML algorithms in comparison to Heuristic-Cap, ML-Cap, and LB, respectively, for 800 VMs. For larger scenarios, above 800 VMs, it is needed to turn-on a new rack and, thus, the second aggregation switch and the core router. Thereby, energy consumption increases due to the higher network energy consumption. For LB, the core router is turned on from 600 VMs due to spreading the VMs across higher number of racks and servers. Also, Heuristic and ML result in high energy savings compared to GH and CVMA, reducing the number of active servers when the total demand of co-located VMs nearly reaches their server capacity during period.

2.11.3 QoS - Analysis of Violations

Figure 2.11 (right y-axis) shows that ML provides a violation reduction, up to 15% and 63% compared to Heuristic and CVMA, starting from 400 VMs. On the other hand, for the lower number of VMs, Heuristic performs better than ML. In order to provide a better trade-off, Hyper reduces the number of violations by 11% compared to Heuristic while decreasing the energy consumption compared to ML. Moreover, GH performs better since it is not fully utilizing CPU resources which is not a case for energy efficiency. Also, Heuristic-Cap and ML-Cap reduce the number of violations dramatically because of the cap set on server load. Due to the nature of load balancing, LB does not present any violation, thus, this is not shown in Fig. 2.11.

Figure 2.11 (left y-axis) shows that Heuristic outperforms ML in terms of the amount of violations for 200 VMs, reaching up to 18% improvement in the worst case. As the number of VMs increases, e.g. 800 and 1000 VMs, the violations of ML get closer to Heuristic and

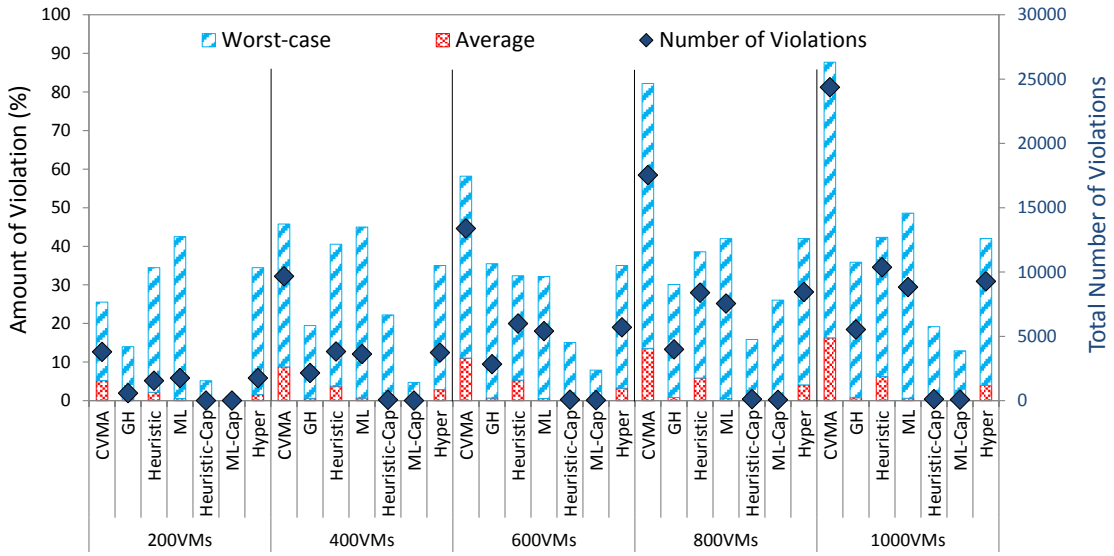

Figure 2.11 – Average, worst-case percentage amount and total number of violations for one week.

GH, inverting the trend for 600 VMs. This is because ML has less control on the worst-case violation during peak loads. Hyper outperforms ML and Heuristic by up to 23% and 14% due to selecting the best method per time slot with lower violation, while leading to only up to 8% overhead compared to both approaches over all cases.

In average, ML provides better results than the other approaches by managing the off-peak VMs load. Finally, for all the cases, Hyper is able to exploit the strengths of Heuristic and ML for providing intermediate solutions.

2.11.4 Multi-Layer Network Traffic Analysis

Figure 2.12 shows the total traffic through the ToR, aggregation-layer switches, and core router. From 200 to 800 VMs, when the core router is turned off, the results demonstrate that ML reduces the ToRs traffic up to %9 compared to Heuristic; while, Heuristic improves the aggregation-layer up to 4%. Note that for 200 and 800 VMs, traffic in the aggregation and core layers is very low for ML, while for Heuristic they are zero. This increase is due to turning on a server in a new rack. For 1000 VMs, ML provides less ToR and aggregation-layer traffic, but higher core traffic compared to Heuristic. Basically, Heuristic results in lower traffic in the upper layers of the network, due to its fine-tuning capabilities. Comparing the ML to CVMA and GH, significant improvements are obtained especially in upper layers for higher number of VMs, when CVMA and GH are less sensitive to dynamic environments, and their benefits become limited for large problems. Hyper achieves up to 5% and 7% improvements in ToR and aggregation compared to Heuristic; while 6% and 9% overheads compared to ML, respectively. For the core layer, Hyper improves 53% compared to ML, but presents an overhead of 16% compared to Heuristic.

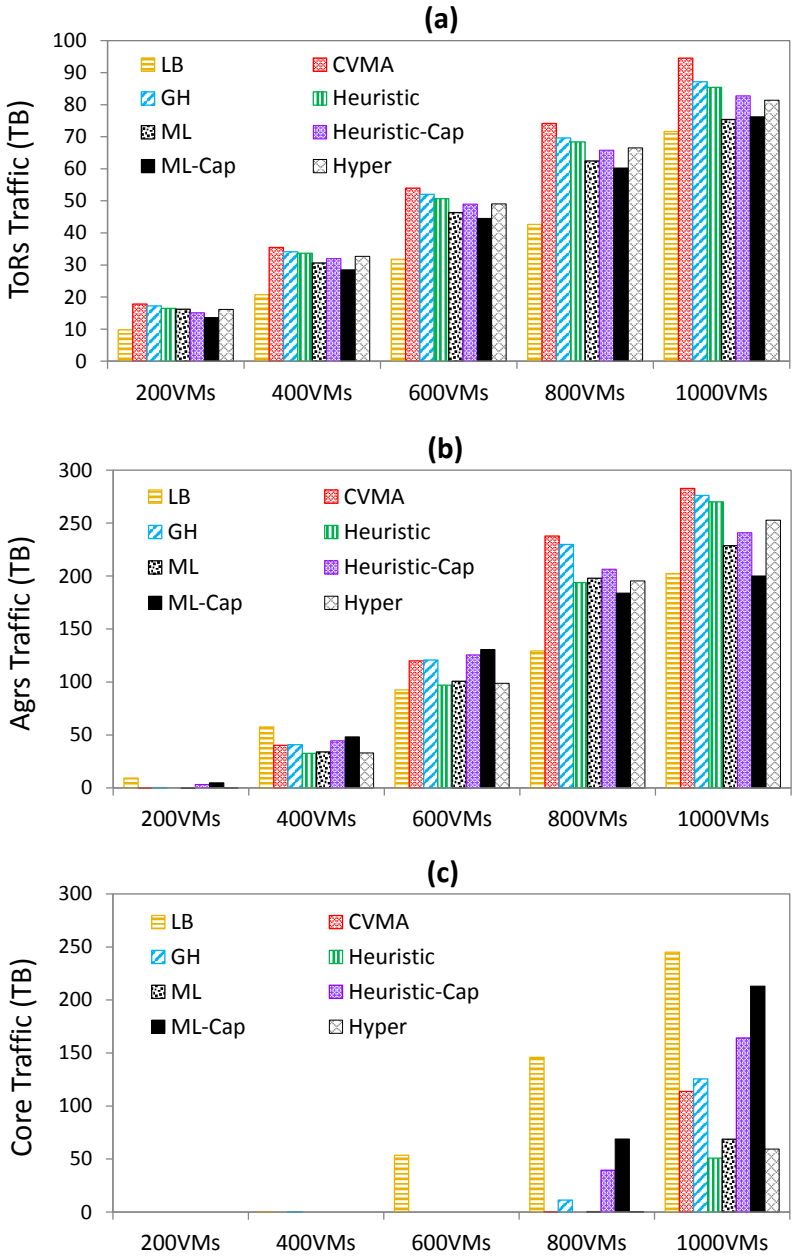

Figure 2.12 – Network traffic of (a) ToR, (b) Aggregation-layer switches and (c) Core router for one week.

Following the same trend, Heuristic-Cap outperforms ML-Cap in upper network layers. Differently, it is needed to turn on the core router for both methods, starting from 800 VMs, due to setting a conservative cap and consequently turning on a server from a new group of racks connected to the aggregation switch. For the same reason, for LB, the core router is turned on starting from 600 VMs, due to balancing the loads among higher number of racks. In this case, the traffic is increased more in upper layers than in lower layers of the network.

Table 2.6 – Total number of migrations for one week

Method	200	400	600	800	1000
LB	30707	64331	98027	131317	165786
CVMA	22904	55102	87642	121332	153891
GH	30072	63578	96836	130260	163643
Heuristic	28608	61724	95014	128540	158635
ML	22539	47437	65690	78726	98407
Heuristic-Cap	29406	62607	96061	129465	163007
ML-Cap	23929	49380	69299	86689	105262
Hyper	25609	57471	83143	111935	132822

2.11.5 Evaluating The Number of Migrations

Table 2.6 represents the total number of migrations for one week. ML reduces the number of migrations by up to 36%, 40%, and 39% compared to CVMA, GH, and Heuristic, respectively. This happens because in ML, the classification accuracy increases for higher number of VMs, thus allowing to place together better candidate VMs. In addition, ML-Cap improves this metric by up to 35% and 37% compared to Heuristic-Cap and LB, respectively. On the other hand, Hyper achieves 13% improvement and 20% overhead on average over different number of VMs, trading-off the benefits of Heuristic and ML. In the worst case, i.e., 1000 VMs, 4.8% and 6.5% of total number of VMs are only migrated on average every sample for ML and Hyper, respectively, that does not lead to high live migration overhead.

2.11.6 Computational Overhead (Execution Time) and Discussion

Table 2.7 shows the average execution time of the proposed VM allocation methods for one week of traces. The results follow the same trend than for the small-scale scenario, with Hyper exhibiting a trade-off between the execution time of Heuristic and ML. In summary, the ML method provides almost the same energy efficiency, but dramatically lowers computational

Table 2.7 – Execution time (sec.) of the proposed algorithms for different number of VMs

Methods	200	400	600	800	1000
LB	3.04	12.38	30.47	49.56	64.44
CVMA	4.45	28.5	88.3	200.6	471.6
GH	1.01	4.12	9.65	17.49	28.6
Heuristics	20.3	88.8	188.7	341.4	519.9
MLs	0.047	0.19	0.421	0.711	1.107
Hyper	10.4	60.86	102.1	199.72	282.35

overhead when compared to Heuristic, while the heuristic method obtains better network traffic and QoS. This situation is because heuristic methods (as well as LB) allow more fine-tuning on the allocation, but present a larger computational overhead, which makes them unsuitable for large-scale scenarios. The results show that Hyper ensures a good trade-off between solution quality (energy, QoS, and network traffic) using the benefits of both Heuristic and ML approaches.

2.12 Summary

Modern cloud data centers need to tackle efficiently the increasing demand for computing resources and address the energy efficiency challenge. An approximate power breakdown shows that both IT (i.e., server, storage and network) and cooling systems encompass over 88% of the total power of a modern data center. Moreover, the explosion of data-intensive applications in current data centers, has led to uneven traffic, bandwidth and communication needs across applications. Hence, the dynamic nature of cloud applications impacts the efficient VM allocation techniques, i.e., consolidation, in two aspects: i) the CPU-load correlation across VMs (i.e., the similarity of CPU utilization traces and the coincidence of their peaks), and ii) the data exchange across VMs (i.e., data correlation). Therefore, jointly incorporating both metrics in a multi-objective optimization is an important aspect to develop resource provisioning policies that are applicable in dynamic scenarios.

As complexity raises, ILP-based methods become unfeasible at run-time to provide an optimal solution. Similarly, heuristics are problem-specific and less sensitive to dynamic environments, and their benefits become limited for large problems. Thus, when tackling dynamic problems with large state and/or action spaces, ML methods, and in particular RL, are suitable techniques. However, in real data center scenarios, VM allocation faces the need to incorporate and assess a wide range of metrics (energy, QoS, network, etc.). This challenges the deployment of ML methods due to their limited configurability. The proposal of methods that balance the trade-offs across these metrics, or dynamically change the optimization goals during run-time to meet data center constraints, remains an open challenge, as it requires a deep assessment on the previous techniques, together with the integration of their strengths. As a result, hyper-heuristics are a promising solution to leverage the benefits of VM allocation approaches determining which method should be executed at each time. This leads to providing better trade-offs than when using the methods separately.

In this chapter I have first proposed a two-phase greedy heuristic and a ML method to tackle the challenge of energy- and network-aware VM allocation, evaluating them in terms of energy, network traffic, QoS, migrations and scalability. Second, I have compared them to the optimal solutions (implemented using ILP), and to two algorithms in the state-of-the-art that are the best in their areas. Third, I have presented a novel multi-objective hyper-heuristic method for the VM allocation problem able to find better solutions by leveraging the strengths of heuristic and ML methods, while allowing users to decide on the importance of each metric.

To conclude this chapter, the experimental results have shown that heuristic and ML methods reach similar results in energy consumption (< 2% difference), consuming only up to 6% more energy than the optimal solution. The ML approach obtains up to 24% server-to-server network traffic improvements when compared to all other methods, and achieving execution time speed-up up to 480x for large-scale problems. On the other hand, the heuristic algorithm results in better QoS and lower traffic in the upper layers of the network structure, due to its fine-tuning capabilities. Finally, the hyper-heuristic algorithm integrates the benefits of heuristic and ML to ensure a good trade-off between solution quality and computational overhead.

3 Multi-Objective Optimization in Green Data Centers

3.1 Introduction

As the electricity cost of data centers doubles every five years, the latest generation of data centers tend to be equipped with on-site Electrical Energy Storage (EES) systems and renewable energy sources (e.g., solar and wind) to reduce costs, carbon emissions, and their dependency on energy from the power grid [4, 8], becoming what it is herein named *green data centers*. However, as renewable energy sources are not constant over the time, it is challenging for data center providers to adjust the power consumption to intermittent renewable energy sources.

Current cloud providers tend to use geo-distributed data centers (i.e., multiple data centers built in different geographical locations, connected through the network, and coupled with renewable energy sources) to reduce costs. They are also used to provide better Quality-of-Service (QoS) for users (placing user data onto closest data centers), and for redundancy purposes (natural events such as fires) [50, 110]. As a result, the complexity of the problem increases dramatically in geo-distributed data centers, where we need to consider inter-data center Virtual Machines (VMs) migration and price diversities while maximizing the renewable and battery energy utilizations. Therefore, these emerging modern data centers require innovative approaches to optimize operational cost (the cost of the energy from the grid) and balance between energy and performance.

Furthermore, instead of renewables being located only at the data center side, they are being integrated also on the supply-side. That is, the power markets operators integrate the share of renewables in gross energy production for carbon emission rate reduction. However, the volatility and intermittency of renewable generation pose significant challenges to Independent System Operators (ISOs), who need to match supply and demand in the power grid in real-time. One potential solution to tackle the challenge of intermittency of renewables is to use large scale EESs [111, 112, 113]. However, they are costly and sometimes unreliable. To overcome this problem, in emerging power markets, the operators provide competitive prices and services for the consumers to better match the demand for power with the supply [114]. Among the services, the most suitable solution is to use demand-side capacity reserves

[64, 65]. That is, the ISO requests consumers to adapt their power consumption depending on its requirements (supply-demand matching).

As data centers are among the fastest growing electricity consumers, they are highly promising candidates to provide demand-side capacity reserves and reduce their electricity costs. A major challenge in this context is that the participation of data centers are largely impacted by the availability of demand-side renewable and EES, incoming workload and efficient server selection, and VM allocation policies. Therefore, to achieve the highest savings, a low-overhead method that incorporates all these aspects is required.

3.1.1 Contributions

In this chapter, I first introduce and propose a multi-level and multi-objective framework for the optimization of green virtualized data centers, to jointly minimize the energy consumption and the carbon footprint, exploiting renewable energy sources, state-of-the-art VMs allocation schemes and Hybrid Electric Systems (HES) (HES are heterogeneous EES systems with different battery technologies). Then, I propose a multi-objective VM placement for green geo-distributed data centers exploiting CPU-load and bidirectional data correlations. During the VM placement process, VMs need to be migrated among data centers to avoid QoS degradation, while defining a hard time constraint to limit the number of migrations. Finally, I jointly optimize the participation of data centers in emerging power markets by adequately selecting bidding parameters, together with the optimal number of active servers and the allocation of VMs to servers, considering the demand-side renewable and EES power.

3.2 State-of-the-art on Energy Optimization in Green Data Centers

3.2.1 Green Energy Sources Optimization

Renewable energy sources integration in the electricity grid, and in particular in green data centers, are currently a hot-topic. Different research ideas have been presented in the last few years that address the problem of exploiting local energy generation to mitigate grid energy demand of data centers [38, 40] and in general of any human activity, such as smart buildings [115]. To address this challenge, Goiri *et al.* [36] proposed a parallel batch job scheduler to adjust the available solar energy to computational workload in a data center regardless of battery management. Ghamkhari *et al.* [37] demonstrated how a convex-mathematical model can be used to maximize the total profit in data centers with respect to stochastic nature of workload and Service-Level Agreements (SLAs) requirements. However, the authors did not utilize a dynamic energy- and cost-efficient workload management strategy based on workload characteristics like CPU-load correlation.

In multiple data centers, Zhang *et al.* [4] formulated the energy and carbon footprint cost minimization problem as non-linear programming, which is then transformed into a linear-

fractional programming. However, they do not take into account techniques such as VM allocation and migration to utilize green energy sources efficiently. Abbasi *et al.* [55] introduced OnlineCC to minimize the operational expenditure while satisfying the carbon footprint reduction. In OnlineCC, the Lyapunov optimization technique and a heuristic algorithm are presented to achieve a near-optimal electricity cost imposing carbon footprint limits to encourage brown energy conservation. Nevertheless, EES (battery bank - cells connected to gain higher capacity) can be used to efficiently tackle the demand peak during the high-price periods to maximize renewables usage and reduce carbon footprint.

EES have been addressed in several works available in the literature [57, 58, 59]. The fundamental idea behind EES management is to use batteries as energy buffers to store the amount of green energy that cannot be used directly by the connected loads. Different management approaches have been proposed to automatically control the energy flows from renewables to loads and storage units [116] and also hybrid solutions (e.g., HES with different types of EES system) for battery banks have been demonstrated [117]. This is particularly interesting nowadays because of the large availability of second-life batteries from electric vehicles that can have up to 75% remaining capacity available for storage applications [118, 119]. Despite HES being still far from market availability (i.e., commercialization), literature demonstrates that this technology is worth the implementation effort. In this work, I followed the approach proposed in previous work [120] to shape the active-Uninterruptible Power Supply (UPS) (or HES) system presented in the following. Rossi *et al.* [120] proposed a two-phase control scheme that exploits intrinsic advantages of different battery technologies mitigating, at the same time, their drawbacks.

A number of research works present allocation methods for energy efficiency in data centers based on workload characteristics. However, there is no evidence in the literature of the joint application of HES optimization and CPU-load correlation-aware allocation techniques to the optimization of data center energy consumption, and the potential savings (both from environmental and money perspectives) are clearly worth the effort for further investigation.

3.2.2 Energy-Aware VM Allocation

At the server level, Dynamic Voltage and Frequency Scaling (DVFS) manages power by leveraging different CPU frequency levels. CPU resource limits are exploited by VMs to control power, providing finer granularity than DVFS [69]. However, at a larger scale, VMs allocation and migration offer additional degrees of freedom in energy savings.

Regarding energy-aware allocation methods, server consolidation solutions have been proposed based on per-VM workload characteristics, i.e., the peak, off-peak, and average workload utilization [12, 13]. These techniques aim to reduce the heat dissipation of hot-spot zones and improve overall power utilization in data centers [121, 122]. Tang *et al.* [123] proposed abstract models to balance computing power in a data center by minimizing peak inlet temperatures. A holistic approach that manages Information Technology (IT), power and cooling equipment

by dynamically migrating servers' workloads and adjusting cooling has been presented in previous study [124]. Experimental results for a virtualized data center demonstrated a reduction of 35% in IT power consumption and 15% in cooling power. Parolini *et al.* [125] presented a control-oriented model that considers cyber and physical dynamics in data centers to study the potential impact of coordinating the IT and cooling management. To achieve further power savings while maintaining the QoS level, joint relationships among VMs, like CPU-load correlations (i.e, the similarity of CPU utilization traces and the coincidence of their peaks), have been exploited in recent works [20, 22, 28, 29].

Verma *et al.* [22] presented a static clustering-based VM placement method by defining VMs' CPU utilization in a time series as a binary sequence where the value becomes '1' when CPU utilization is higher than a threshold value, and is '0' otherwise. However, the envelopes of VMs have a single value to represent all the CPU utilizations ignoring the original time-series of each application. Thus, this static consolidation is applicable only when the envelopes of the VMs are stationary. Meng *et al.* [28] proposed a VM sizing technique that pairs two uncorrelated VMs into a super-VM by predicting their loads. However, once the super-VMs are formed, this solution does not consider dynamic changes of the VMs' load, which limits further energy savings. Therefore, these approaches do not work well with non-stationary and fast-changing VM behaviors in particular for scale-out applications. In recent research [20], a power-efficient solution has been proposed based on the First-Fit-Decreasing heuristic to separate CPU-load correlated VMs especially targeting the characteristics of the scale-out applications. They also exploit server's DVFS techniques to achieve further energy savings. Lin *et al.* [85] used the peak workload characteristics to measure the similarity of VMs' workload.

Dynamic allocation via migration is also used for minimizing data centers cost and energy consumption. Ruan *et al.* [87] proposed a dynamic migration-based VM allocation method to achieve the optimal balance between server utilization and energy consumption such that all servers operate at the highest performance-to-power levels. Wang *et al.* [88] also addressed a matching-based VM consolidation mechanism using migration such that active servers can operate close to a desirable utilization threshold. The main drawback of those approaches is their high VM migration overhead. Thus, as opposed to short-term decision, Chen *et al.* [93] proposed a long-term VM consolidation mechanism such that the total demand of co-located VMs nearly reaches their host capacity during their lifetime period. This algorithm detects the utilization pattern of each VM based on the four types of simple pulse functions. Nevertheless, these schemes do not take into account the renewable energy sources and data center system model in modern green data centers.

3.2.3 Network-Aware VM Allocation

To provide better network resource usage and, thereby, improve the performance of applications (i.e., response time), certain algorithms, [21, 23], take into account the communication among VMs. Agarwal *et al.* [50] defined a system called Volley to automatically migrate data

across data centers. This solution uses an iterative optimization algorithm based on weighted spherical means considering the data locality, bandwidth costs, and storage capacity. The goal of this placement is to minimize user-perceived latency. Xin *et al.* [51] introduced an algorithm to split a request into partitions and then distribute them amog the data centers using a workload balancing method. Cordeschi *et al.* [52] developed an optimal minimum-energy scheduler for the adaptive joint allocation of the task sizes, computing rates, communication rates and communication powers that operate under hard delay constraints. The goal is to minimize the overall communication and computing energy consumption by dividing the problem into two simpler sub-problems.

However, previous works assume that data dependencies are given in the form of a Directed Acyclic Graph (DAG). Differently, in practice, there are often cyclic communication scenarios, where two VMs regularly exchange information in both directions. As a result, Biran *et al.* [16] proposed two heuristic algorithms to address bidirectional data communication under time-varying traffic demands. The first one, 2PCCRS, can be applied only if the network topology is a tree. The second one, GH, has more freedom during VM allocation and is applicable for different types of network topology. Nonetheless, both approaches neglect the main providers' objectives, including operational costs and energy consumption.

3.2.4 Operational Costs

The use of geo-distributed data centers allows designers to minimize the overall electricity cost by exploiting dynamic workload allocation across data centers based on the renewable sources, and temporal and regional diversities of electricity price [42, 43, 60]. However, data transfer among VMs is an important aspect missing from these problem formulations which directly affects the response time and user experience. In addition, an energy-efficient management based on existing CPU-load correlation to achieve more energy and cost savings is missing from these works. Le *et al.* [44] presented a workload assignment and migration technique to minimize the costs of energy consumed by IT and cooling equipment considering the fluctuations of electricity price and the variability of the data centers' Power Usage Effectiveness (PUE). Zhao *et al.* [47] addressed the problem of dynamic pricing by designing an efficient online job scheduling and server provisioning in each data center to maximize the time-average overall profit of the cloud provider with respect to delay constraints. Work by Gao *et al.* [3] addressed the same problem targeting energy costs and the delay based on the data center distance. Gu *et al.* [41] presented an optimization problem, which is formulated as a Mixed Integer Linear Programming (MILP) problem and then solved by a computation-efficient heuristic algorithm to minimize electricity cost via data center resizing. However, without the consideration of the characteristics of the workload, these research works are sub-optimal to minimize operational cost, energy consumption, and performance (i.e., response time).

All in all, the effects of joint CPU-load and data correlation on VM allocation have not been

previously considered for green geo-distributed data centers to optimize operational costs, energy consumption and response time. Furthermore, the research on power and cost management in green data centers should also be considered, when the data centers participate in emerging power markets.

3.2.5 Data Center Cost Optimization On Emerging Power Markets

Power market operators have recently introduced smart grid demand-response programs, in which electricity consumers regulate their power usage following provider requirements in real-time [68]. There are several power markets and demand-response programs with different timescales and the frequency of request for power regulation. Short-term power markets contain different pricing types: i) day-ahead markets, allowing participants to determine their power and reserve bids for the next day, ii) hour-ahead markets, and iii) 5-minute (close to real-time) markets [126, 127, 128]. Also, power markets provide a request command for consumers to regulate their power every millisecond (known as frequency control), few seconds (known as Regulation Service (RS)), or few minutes [129, 130].

A recent systematic comparison of multiple types of service markets [66] demonstrates that RS reserve provision is the most suitable and profitable program for data centers. A few offline and online control policies for data centers RS reserves provision are proposed in the literature [69, 70, 131, 132, 133]. Most of these studies [131, 132, 133] use highly simplified data center models for RS reserves provision. Chen *et al.* [69] propose an online policy that simply regulates the server power to track the instant value of the RS signal as accurately as possible. Chen *et al.* [70] also present a dynamic power control policy that modulates data center power consumption using server power capping techniques and different server power states.

As described in this section, none of the previous works tackles this problem on a green data center equipped with on-site renewables and EES (a popular research direction [113, 134]). Moreover, none of them has proposed a low-overhead joint strategy that computes the market power and reserve bidding problem in a fast analytical way, along with determining the number of active servers needed for the allocation phase, while minimizing at the same time the electricity cost of the green data center.

3.3 A Novel Electric System Model for Green Data Centers

In this section, I introduce the proposed model where data center equipment, Photovoltaic (PV) modules, power grid, and UPS are connected as shown in Fig. 3.1. The IT equipment and cooling system inside the data center are the major contributors to power consumption in comparison with the other data center components. These components are combined using a Power Distribution Unit (PDU) that eventually connects to the Charge Transfer Interconnect (CTI) bus that serves the whole data center equipment [135]. In this framework the UPS is

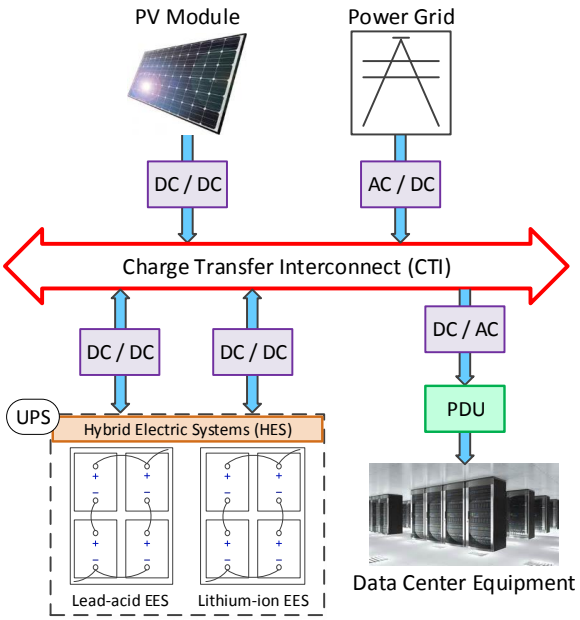

Figure 3.1 – The complete electric system modeling framework for green data center.

designed as a HES to provide both supply in case of power grid outages and a buffer for green energy.

The system is equipped with different EES systems (e.g., two battery banks), a PV module and the bidirectional CTI bus, managed by a dedicated controller (not shown) as presented in prior work [136]. Hence, the system comprises both Alternating Current (AC) and Direct Current (DC) sources/loads, while the CTI is a DC path. Therefore, each unit is connected to the CTI through a DC-DC, AC-DC, or DC-AC converter interface for level shifting and charge routing.

One constraint is defined on the system: if the exceeding renewable energy cannot be stored, it cannot be injected back into the main grid. Thus, renewable energy and batteries should completely sustain the load of the data center or, at least, provide supply during outages and periods with the highest price. These choices are justified by the fact that selling energy back to the grid, namely providing net-metering ancillary service to the Distribution System Operator (DSO), follows rules that are country-specific and strongly dependent on the interface between data center and energy network. Moreover, data centers are usually big energy consumers and it is unlikely to have enough excess green energy to justify the effort (economically and technologically) of improving the electric system to handle this task. The peak/off-peak price scenario in a regulated energy market instead, can be easily implemented also in a free energy market scenario where the energy price is continuously evolving. In this case, the proposed assumption can be seen as a threshold on the time-varying price values: when the free market price is below the threshold it is more convenient to buy from the grid, and the opposite when the price rises. In the following sections, I describe the models used for PV, EES, and power

management at the CTI.

3.3.1 Hybrid Electric Systems (HES)

The HES comprises two heterogeneous EES systems (battery banks) managed in a hierarchical fashion: i) a lead-acid array and ii) a lithium-ion array. The battery model is based on Peukert's law [137], and has been conceived as a plug-and-play component that can be easily replaced and adapted. The goal is to model HES that combine the advantages of the different battery technologies (lead-acid and lithium-ion). To correctly account for the benefits of the HES, the charging sequences of different battery banks need to be controlled, and batteries aging (i.e., power loss in each EES or battery bank) should be modeled.

First, Eq. 3.1 defines the State-of-Health (SoH) of the battery as the ratio of currently available charge capacity (C_{ref}) to the nominal charge declared by the manufacturer (C_{nom}). Then, Eq. 3.2 defines the charge capacity as a linear combination of the previous charge and the charge that is drained. Z_b denotes the linear aging coefficient, which is dependant on the battery technology [138]. Finally, Eq. 3.3 and 3.4 determine the State-of-Charge (SoC) and the equivalent battery current (I_{eq}), respectively. These are a function of the current flowing from batteries (I_b), and the nominal battery parameters: i) the reference discharge current (I_{ref}) provided by the manufacturer and used to compute the reference charge, ii) the Peukert's coefficient (k_b), and iii) the charge actually used by the system and computed as current I_{eq} times time (t) in seconds. The SoH of the battery only decreases during discharge (therefore it is only calculated during discharge), whereas the SoC is updated during both charge and discharge cycles. Further details about the model and its usage can be found in literature [137, 138].

$$SoH(t+1) = \frac{C_{ref}(t+1)}{C_{nom}} \tag{3.1}$$

$$C_{ref}(t+1) = C_{ref}(t) - C_{nom} \cdot Z_b \cdot (SoC(t) - SoC(t+1)) \tag{3.2}$$

$$SoC(t+1) = \frac{C_{ref}(t) \cdot SoC(t) - I_{eq}(t) \cdot t}{C_{ref}(t)} \tag{3.3}$$

$$I_{eq}(t) = \left(\frac{|I_b(t)|}{I_{ref}}\right)^{(k_b-1)} \cdot I_b(t) \tag{3.4}$$

While lead-acid technology is cheaper, easier to recycle and has a wider working temperature range, it suffers from a limited number of sustainable cycles (i.e., lifetime). On the contrary, the lithium-ion technology instead offers at least one order of magnitude higher number of cycles, but at the expense of a higher cost. To maximize the lifetime of the storage (in particular the lead-acid bank), some constraints should be put on the allowed Depth-of-Discharge (DoD) of the battery banks. Hence, to force battery bank to work in the optimal range of SoC, DoD is set to 65% for the lead-acid battery bank and 70% for the lithium-ion bank [139]. The remaining capacity is still available in the event of an outage, thus providing standard UPS support.

The parameters of the general purpose model (maximum and reference charge/discharge currents) are tuned according to commercial devices, a VARTA Professional Dual Power (230 Ah @ 12 V) [140] as lead-acid battery, and a StarkPower 'UltraEnergy' (100 Ah @ 12 V) [141] as the lithium-ion battery.

3.3.2 Photovoltaic (PV) Module

The PV module provides energy proportional to the intensity of the solar irradiance impinging on it, which in turn depends mostly on the weather. In this framework, it is implemented as a linearly varying voltage source, with an integrated Maximum Power Point Tracking (MPPT) controller [120] and tuned accordingly to real device characteristics [142]. Real sun irradiance [143] and temperature profiles [144] are used for the experiments. Equation 3.5 presents the linear model of the PV array as follows:

$$P_{PV} = \left[P_{PV,STC} \cdot \left(\frac{G_T}{1000} \right) \cdot \left(1 - \gamma \cdot (T_j - 25) \right) \right] \cdot N_{PV,S} \cdot N_{PV,P} \tag{3.5}$$

$$T_j = T_{amb} + \left(\frac{G_T}{800} \right) \cdot NOCT - 20 \tag{3.6}$$

The parameters are evaluated in Nominal Operating Cell Temperature (NOCT) and Standard Test Conditions (STC), which yield the nominal output power ($P_{PV,STC}$) of $2.65W$, irradiance level (G_T) of $1000W/m^2$ @ $25°C$, and a temperature coefficient (γ) of $0.0043\%/°C$. $N_{PV,S}$ and $N_{PV,P}$ are the number of series and parallel cells in the module. The cell temperature (T_j) is obtained using Eq. 3.6, where T_{amb} is the environmental temperature, $G_T = 800W/m^2$ @ $20°C$ and $NOCT = 45.5°C$. The PV module size (the number of cells and panels) is also tuned based on the peak power of the green data center with respect to its load, which is the most common approach to PV sizing [145].

3.3.3 Power Management on The Charge Transfer Interconnect (CTI) Bus

To correctly connect the DC and AC power sources, control the charging/discharging current sequences of the battery bank, and to model EES aging (i.e., two power losses in EES, charge capacity rate and SoH degradation, due to the charge rate), a fine-grained system model is used to manage the energy sources in a realistic scenario.

In this system, the IT equipment is connected via a PDU to the CTI bus that serves the whole facility [135]. The battery banks are attached to the CTI by means of a bidirectional DC-DC converter, whereas the PV one is unidirectional. Power grid and data center are modeled as power source and load (i.e., P_{Grid} and P_{DCT}), connected to the CTI by means of AC-DC and DC-AC converters, respectively. All converters have an efficiency (η_X), defined as the ratio of power requested by the system with respect to the nominal power delivered by the converter. Equations 3.7 to 3.10 describe the AC-to-DC and DC-to-DC conversion functions used for

each system component, as follows:

$$P^{CTI}_{Grid}(t) = P_{Grid}(t) \cdot \eta_{ACDC} \tag{3.7}$$
$$P^{CTI}_{PV}(t) = P_{PV}(t) \cdot \eta_{DCDC} \tag{3.8}$$
$$P^{CTI}_{EES,n}(t) = P_{EES,n}(t) \cdot \eta_{DCDC} \tag{3.9}$$
$$P^{CTI}_{DCT}(t) \cdot \eta_{DCAC} = P_{DCT}(t) \tag{3.10}$$

To control and optimize the amount of power usage of each source, the power management problem is solved at the CTI bus level [136, 146]. Equation 3.11 represents the power balance of the system and states that the sum of the input from the power grid, PV and HES (N_{EES} shows the number of battery arrays that compose the HES and $N_{EES} = 1$ indicates a homogeneous system, or EES) must be equal to the data center requirements. α is a directional parameter that can be -/+1 depending on the charging/discharging status.

$$P^{CTI}_{DCT}(t) = P^{CTI}_{Grid}(t) + P^{CTI}_{PV}(t) + \sum_{n=1}^{N_{EES}} \alpha \cdot P^{CTI}_{EES,n}(t) \tag{3.11}$$

In order to reduce the computational complexity and generalize the system models, a fixed CTI voltage level (V_{CTI}) and converters with $\eta_X = 90\%$ efficiency are considered. Detailed efficiency curves for high-power equipment are not publicly provided by manufacturers [147] but still efficiency is claimed to stay within the 80-95% range for loads down to 20%.

3.4 Joint Computing and Electric Systems Optimization Framework for Green Data Centers

In this section I present the design of a dedicated control loop which connects the VMs allocation scheme to the HES manager and optimizes the resources in real-time. At the same time, this modular structure allows to use the general purpose models in each module for performance evaluation, model verification and feasibility analysis. The framework consists of two modules running concurrently, the Datacenter Energy Controller which minimizes the energy consumption of data center without any significant QoS degradation and shares the real energy consumption data with the Green Energy Controller; and the Green Energy Controller that manages renewable sources and HES, providing feedback to the Datacenter Energy Controller.

The Datacenter Energy Controller is based on a state-of-the-art CPU-load correlation-aware VM allocation scheme [20] due to the existence of high CPU variability in applications' patterns. The Green Energy Controller, based on previous study [120], is a two-phase controller that takes into account the cost policies of the power grid energy and exploits forecasts of both the data center's load and of the incoming energy from renewables. The framework uses PV modules as green energy source and two battery technologies (lead-acid and lithium-ion) as

the HES. The battery banks are managed with different priorities and roles.

3.4.1 Simulation Framework Description

I developed a discrete-time framework that simulates the target green data center, with hourly time-steps. The Green Energy Controller manages the PV modules, the heterogeneous batteries (HES), and the CTI (presented in the previous section) considered in this framework, and has been implemented using Matlab. The Datacenter Energy Controller, implemented in C++, manages the data center and VMs allocation scheme. Both components communicate using sockets for interprocess communication.

The overall diagram of the simulation framework, that jointly manages the Green Energy and Datacenter Energy Controllers, is shown in Fig. 3.2. At the beginning of the simulation time horizon (offline phase), the Green Energy Controller computes the expected energy budget for the data center, processing historical data center power profiles as well as the sun irradiance forecasts. This task is executed only once and provides a preliminary energy budget for the whole simulation horizon.

The online phase starts when the offline phase of the Green Energy Controller sends the available energy budget to the Datacenter Energy Controller for the first time slot. Next, it waits until the VMs allocation to be completed according to the prediction of upcoming loads of VMs, then, receives back the real energy demand of the data center computed based on the

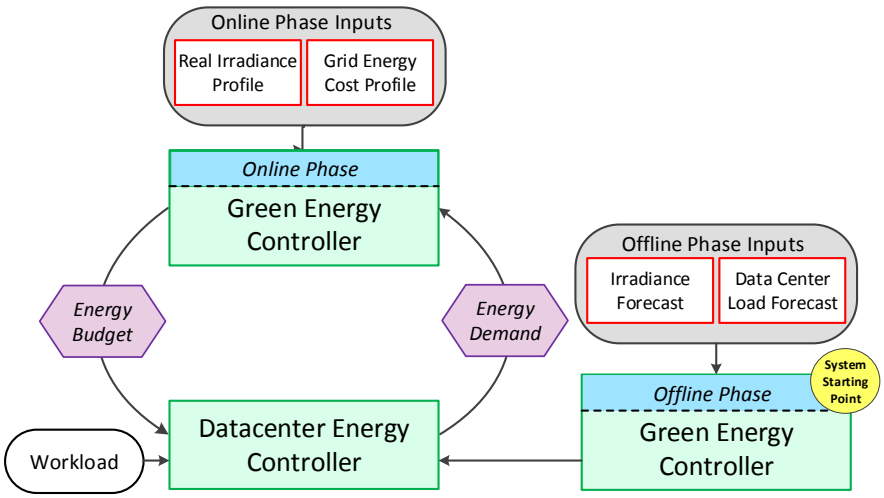

Figure 3.2 – The simulation framework that jointly manages the Green Energy and Datacenter Energy Controllers. The offline phase constitutes the starting point of simulation, and is executed once at the beginning of the simulation time to compute the expected energy budget for the data center. In the online phase, at each time slot, the Datacenter Energy Controller first receives forecasted workload and energy budget from the Green Energy Controller to allocate VMs to servers, then, sends back the real energy demand to the Green Energy Controller.

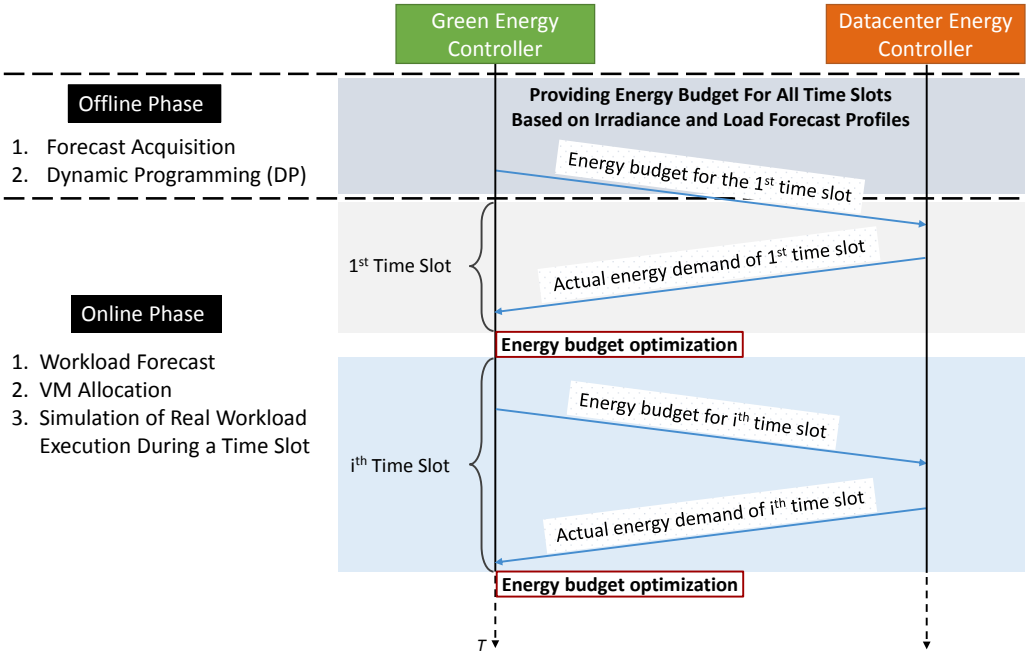

Figure 3.3 – Overall process of the proposed framework - joint Datacenter and Green Energy Controllers.

real workload. Therefore, the Green Energy Controller compensates the differences between: i) expected and available green energy, and ii) real energy consumption and energy budget for the data center, using the lithium-ion battery as additional energy reserve or the power grid if both banks in the HES have been drained. To this end, if the actual energy consumed by data center is higher than the expected, the Green Energy Controller compensates the data center energy requirements. At the end of each time slot the Green Controller provides an updated budget to the Datacenter Energy Controller for the VMs allocation of the next time slot.

On the other side, the Datacenter Energy Controller tries to find the best allocation for VMs on the servers at each time slot using the VMs characteristics from the previous time slot (i.e., using a last-value predictor) and the energy budget provided by the Green Energy Controller. The goal is to allocate VMs to the minimum number of servers, optimizing total data center energy consumption, as it will be explained in the following section. After the allocation is completed, the Datacenter Energy Controller communicates the actual energy demand for the current time slot to the Green Energy Controller. Both the controllers are invoked periodically, at every time slot, T. The overall communication process between the controllers has been shown in Fig. 3.3. In the following sections, I describe these two controllers in detail.

3.4.1.1 Datacenter Energy Controller

In order to evaluate the simulation framework, an energy-efficient workload allocation method needs to be adopted to use the servers resources efficiently. This favors consolidation and

leads to power savings by lowering the number of active servers. Due to the existence of high variability in CPU usage of the VMs, the CPU-load correlation metric should be considered for efficiently consolidating VMs into minimum number of servers. As defined in Section 1.1.2, CPU-load correlation is the VMs' utilizations coincidence during a certain time interval, in particular when the peaks of two VMs occur at the same time. Therefore, based on the VMs utilization patterns, highly CPU-load correlated VMs should be placed apart, in different servers such that the aggregated utilization of co-located VMs nearly reaches the servers maximum capacity during a time slot. In this context, a correlation-aware VM allocation method has been proposed in previous work [20] based on a First-Fit-Decreasing heuristic.

The algorithm defines a cost function to efficiently quantify the CPU-load correlation between the VMs across a certain time horizon. At the beginning of each time slot, the correlation between any two VMs is updated based on the history. For allocation phase, first, a server with largest remaining capacity is selected. Then, the VMs are allocated such that the correlation among the co-located VMs on the server is minimized, while the server does not exceed its total CPU capacity. Differently from this algorithm, I select a VM for the server minimizing the ratio of correlation to the server CPU capacity. Once all the VMs are allocated into servers, an optimal Voltage/frequency (V/f) level for each server is determined, while satisfying QoS requirements. This correlation-aware VM allocation algorithm is periodically invoked at every T.

3.4.1.2 Two-Phase Green Energy Controller

The Green Energy Controller is a two-phase controller - offline and online phases - that manages the CTI bus and provides guidelines to the Datacenter Energy Controller, by recursively solving the set of equations presented in Section 3.3.

The offline phase's goal is: i) to find the best resource allocation strategy to minimize the energy intake from the power grid, and ii) to maximize the lifetime of the lead-acid battery bank by minimizing the number of charge-discharge cycles and using as many as uninterrupted cycles possible. This phase is based on Dynamic Programming (DP). DP solves complex problems by splitting them into lower complexity ones, solving and storing each solution. Thus, when a previously solved problem is found the system looks up the previous solution, saving computational time. The controller takes as inputs the expected workload of the data center, the price profile of the energy from the grid, and the irradiance forecasts for the whole time horizon [120, 148]; in this phase (i.e., offline phase), the controller manages the lead-acid battery bank only. The algorithm ranks all the possible system states (charge to discharge, charge to charge, discharge to charge and discharge to discharge) for each time slot in the simulation horizon, that fulfills the constraints mentioned in Section 3.3. For each state transition it assigns a weight based on the battery usage. The higher the weight, the lower the ranking. Finally, it provides an optimal energy budget for each time slot and the best utilization strategy for the lead-acid bank for the whole time horizon. Only the budget for the first time slot is then sent to the Datacenter Energy Controller and this message triggers the

online phase. All the other energy budgets computed are kept in memory for the online phase to use them when the offline concludes.

The online phase, for each time slot, optimizes the initial energy budget, computed by the offline phase, trying to compensate the difference between expected workload and irradiance forecast with respect to the real data measured by the system. In this phase, the controller also manages the lithium-ion battery bank mainly to compensate error in the forecasts and to maximize the lifetime of the lead-acid bank. For each time slot, the Green Energy Controller finds the currents balance in the CTI based on the Kirchhoff currents law to minimize the energy taken from the power grid (optimization goal), to fulfill the offline lead-acid battery scheduling and to supply the load. For each component of the system (grid, PV, batteries and load), constraints are set for the currents (e.g., maximum charge and discharge current) and the input power from the grid. Problem constraints (current flow direction for batteries and use of the grid) change in accordance with the system state. In this way, on CTI bus, it is possible to force the lithium-ion battery to be discharged when the lead-acid battery is recharged, in particular when the green energy is unavailable or lower than the load, and the grid price is high. In this case, the lithium-ion battery is used to cover the error of offline phase considering the grid price, available renewable energy and the current lead-acid state. At the end of time slot, the actual energy balance is updated to the data center and this triggers a new simulator cycle for the following time slot (i.e., $T+1$).

3.4.2 Framework Evaluation

I assessed the effectiveness and applicability of the proposed framework to larger scale problems by simulating a time horizon of two weeks, using workload traces obtained from a real data center setup, and real irradiance and temperature profiles.

3.4.2.1 Experimental Setup

A green urban data center is modeled consisting of two components: i) IT equipment including servers, and ii) Computer Room Air Conditioning Computer Room Air Conditioning (CRAC). The effectiveness of the proposed framework is evaluated with a virtual testbed consisting of 250 homogeneous servers. An Intel Xeon E5410 server configuration has been targeted which consists of 8 Intel cores and two frequency levels (2.0GHz and 2.3GHz), and used the power model proposed in [149].

To simulate the data center workload and energy demand, I used the VMs' CPU utilization of a real data center setup sampled every 5 minutes for one day. To extend the trace for up to 14 days, the samples have been repeated. In addition, to generate different samples every 5 seconds for each day, a lognormal random number generator [150] is used, whose mean is the same as the collected value for the corresponding 5-minute sample rate. Such assumption has been proved by real-trace studies, since the real data center's workload shows significant

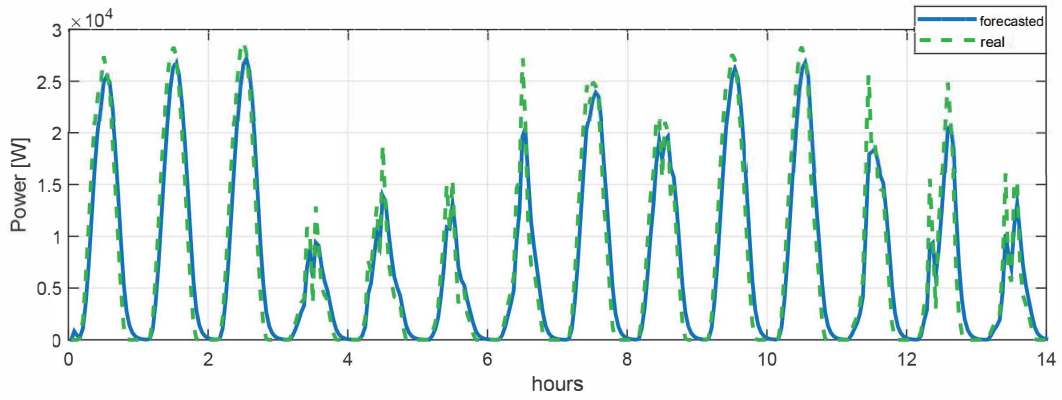

Figure 3.4 – Solar power profile, forecasted versus real.

variability and a daily pattern during one week [25].

To model the HES, the size of the lead-acid battery is usually considered larger than the lithium-ion one due to its cheaper technology. In the simulations, two configurations have been considered: i) the HES-1 with 48 kWh as lead-acid capacity (16.8 kWh available) and 24 kWh as lithium-ion capacity (7.2 kWh available); and ii) the HES-2 with 96 kWh (33.6 kWh) and 48 KWh capacity (14.4 kWh) respectively.

PV module size has been tuned considering two different cases of peak power production (hence the number of cells and panels) that are 10 kWp for the HES-1 simulation scenario and 30 kWp for the HES-2. The irradiance forecasts have been computed according to the algorithm presented in [9]. An example of the two resulting sequences is depicted in Fig. 3.4. At the same time, I used a hourly averaged energy consumption profile from the real data center as forecast, which results in a smoothed profile compared to the original one. Finally, a peak/off-peak price scenario has been considered from a regulated electricity market for the energy taken from the power grid (the Zurich's tariff 7.5/14.9 cent/kWh [151]).

3.4.2.2 Experimental Results

In this section, I first compare the following algorithms in terms of energy and QoS to show the efficiency of the CPU-load correlation-aware method. Then, I evaluate the effectiveness of the joint CPU-load correlation-aware algorithm as the Datacenter Energy Controller (Section 3.4.1.1) and two-phase Green Energy Controller (Section 3.4.1.1) in two separate sets of experiments: i) winter scenario with low renewable energy and impact of HES on cost saving, and ii) summer scenario.

- Best-Fit-Decreasing (BFD): the problem of VM allocation is a well-known bin-packing problem [100]. Among the different methods, I considered a conventional BFD heuristic approach for solving this problem. This algorithm sorts VMs in decreasing order

according to their utilization, and allocates each VM to the server that provides the closest resource requirements with respect to this VM's utilization (i.e., the server with the smallest remaining capacity that is sufficient to host the VM).

- Peak Clustering-based Placement (PCP) [22]: to consider also other attributes of the VMs, like the CPU-load correlation, I chose this method to solve the VM allocation problem for further power savings. The authors presented a static clustering-based VM allocation method by defining VMs' utilization in a time series as a binary sequence where the value becomes '1' when utilization is higher than a threshold value, and is '0' otherwise. This algorithm first clusters VMs such that the utilization envelopes of the VMs classified in different clusters do not overlap. Then, it allocates VMs to servers in order to co-locate VMs in different clusters.

- CVMP: the Correlation-aware VM allocation aPproach (explained in Section 3.4.1.1) based on the state-of-the-art research [20].

Figure 3.5 compares the total energy consumption of the three aforementioned approaches under different number of VMs in the system for a horizon of two weeks. As different VMs running the same job tend to have similar utilization patterns [93], the trace of 250 VMs has been repeated to produce the higher number of VMs.

The CVMP algorithm provides up to 11.6% and 7.3% energy savings compared to BFD and PCP, respectively, due to using lower number of servers as well as lower frequency levels more frequently. Even if high and fast-changing correlations are observed among the VMs, PCP provides similar results than BFD. This is due to classifying the VMs into only 1 cluster during most of the time periods. As a result, when the number of clusters is 1, PCP has the same

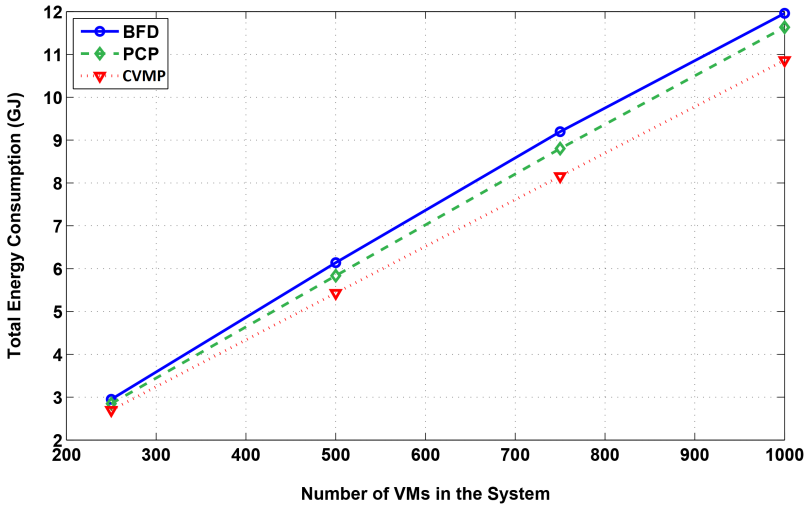

Figure 3.5 – Total energy consumption of data center under different number of VMs for a horizon of 14 days.

behaviour than BFD. Note that the semi-linear trend of the energy consumption depends on the analogous behavior of the workload among different days, in a typical data center.

Table 3.1 shows the worst-case violation defined as the maximum percentage of the number of time samples (i.e., every 5 seconds per time slot) in which servers' overutilization occurs (i.e., when the aggregated utilization among co-located VMs is beyond the CPU capacity of a corresponding server), to the total number of time samples of a time slot. A graphical representation of these data is provided in Fig. 3.6 for a time horizon of two weeks for the different number of VMs in the system. The CVMP scheme provides a drastic reduction of the violations, up to 10.4% and 9.6% compared to BFD and PCP, respectively. In CVMP, VMs are allocated based on their peak utilizations, which were predicted from their history. Despite the provision based on the peak utilization, a quality degradation is observed over the three approaches due to the miss-predictions of the peak utilization, especially during abrupt workload changes under increasing the number of VMs in the system. However, the CVMP method can statistically reduce the probability of the violation by co-locating uncorrelated VMs. Thus, the probability of joint under-predictions among the co-located VMs is drastically decreased.

Using the CVMP algorithm, I performed the complete framework simulation (VM allocation,

Table 3.1 – Worst-case violation (%) as the maximum percentage of the number of time samples (i.e., one sample per 5 seconds) per time slot in which servers' overutilization occurs, to the total number of time samples of a time slot (i.e., 720 time samples per time slot), for different number of VMs scenario.

Approach	Number of VMs			
	250	500	750	1000
BFD	2.1	4.9	9.6	18.4
PCP	1.1	2.8	3.4	17.6
CVMP	0.85	2	3.1	8

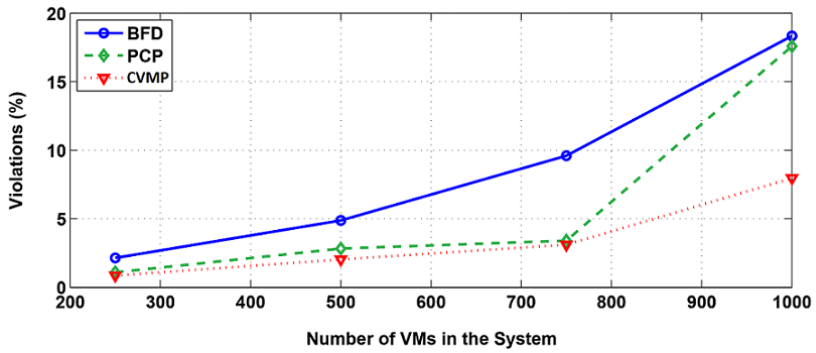

Figure 3.6 – Trend of maximum violations (%) under different number of VMs for a time horizon of 14 days.

green energy controller and communication between the two controllers) with $T = 1$ hour and with predictions of upcoming workloads of data center using a last-value predictor.

In a joint optimization framework, Table 3.2 summarizes the results in terms of cost savings according to the number of VMs, the HES-size and the season. The cost savings are computed as the difference between electricity cost to sustain the data center workload with or without the renewable energy sources. As expected with larger battery capacities (HES-2 configuration), higher savings are obtained. I also compared with the cost saving of using the PV panels without any storage (between brackets) to demonstrate the advantage of the proposed approach.

In the winter scenario, the low irradiance and the cold weather strongly impact the renewable energy generation, causing the batteries to rarely reach the full charge. However, the storage system usage still provide advantages for cost savings. On the contrary, during summer the batteries are fully exploited in the presence of higher renewable energy generation, resulting in higher savings with respect to the previous scenario. According to the model, during summer, when the HES system's usage is more intensive, a maximum SoH decreasing of 0.07% (ratio between nominal and remaining capacity) is experienced, which means a lifetime longer than 15 years to reach the 70% of nominal capacity (lead-acid battery near the end of life).

Table 3.2 – Overall framework results in terms of economic benefit of renewable-enabled data center with respect to a grid connected one. Two HES configurations are evaluated, HES-1 with 48 kWh as lead-acid and 24 kWh as lithium-ion capacity; HES-2 with 96 kWh and 48 kWh capacity respectively.

Configuration	Winter Savings (PV only)	Summer Savings (PV only)
	250 VMs	
HES-1	29.30% (25.54%)	76.46% (57.86%)
HES-2	62.22% (38.72%)	96.13% (66.45%)
	500 VMs	
HES-1	14.30% (13.16%)	55.92% (48.00%)
HES-2	38.43% (31.30%)	85.28% (61.59%)
	750 VMs	
HES-1	9.53% (8.76%)	43.49% (40.16%)
HES-2	27.69% (24.86%)	73.39% (57.35%)
	1000 VMs	
HES-1	7.05% (6.57%)	33.34% (32.51%)
HES-2	20.64% (19.16%)	65.28% (53.96%)

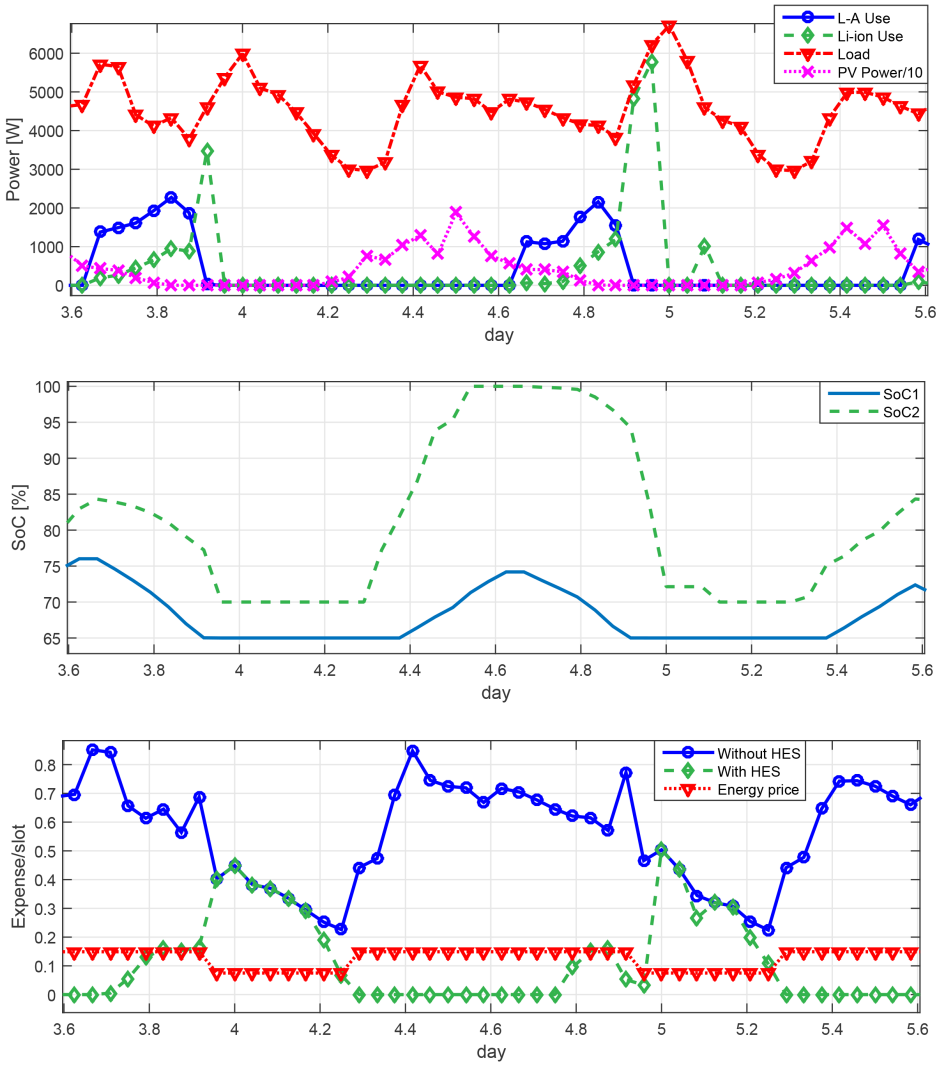

Figure 3.7 – Two days framework evolution with 500 VMs, HES-2 (96 kWh lead-acid and 48 kWh lithium-ion capacity) configuration and summer irradiance (48 time slots). Power profile of the data center components (top); percentage SoC of the lead-acid (SoC1) and lithium-ion (SoC2) battery bank (middle); cost per time slot (bottom).

Finally, Fig. 3.7 shows a two-days view (48 time slots) of the framework evolution with 500 VMs, summer irradiance and HES-2 configuration. The role of the energy buffer can be observed that allows to use green energy when there is no input from the PV panels (Fig. 3.7-top) and the resulting money saving (Fig. 3.7-bottom). In the specific time horizon depicted (Fig. 3.7-middle), a low level of irradiance is experienced compared to other days in the overall horizon (as referred to Fig. 3.4). It results in a lower amount of energy available to recharge the batteries, in particular the lead-acid battery bank which has a bigger capacity and a smaller recharge current with respect to the lithium-ion one. Similar considerations can be made for the other 3 cases.

3.5 Multi-Objective VM Allocation Method for Green Geo-Distributed Data Centers

In this section I propose a multi-objective VM placement method (i.e., clustering and allocation) for the green data center system presented in the previous section, which considers network topology, exploiting CPU-load and bidirectional data correlations in one problem. I present a two-phase controller along with a migration technique that splits the complex VM placement problem into clustering and allocation phases. In the first phase, the global controller exploits the CPU-load and data correlations to cluster the VMs for the data centers. In the second phase, the local controllers of each data center allocate the VMs cluster of each data center to servers exploiting the CPU-load correlation.

Since the proposed algorithm optimizes the VM placement problem at each time slot T ($T = 1\ hour$) to achieve the best solution according to renewable energy forecast and VMs utilization prediction, it is only needed to use a low-complexity green controller for each data center to manage its energy sources according to the real renewable energy and energy consumed by data center during the time interval of $[T, T+1)$. Differently from the previous green energy controller presented in Section 3.4.1.2, a rule-based greedy heuristic is used regardless of offline phase and high-complexity DP algorithm.

3.5.1 Network and Latency Model

In order to accurately model and compute the network latency, the proposed algorithm considers intra-data center local links with bandwidth (B_L) (to access the network-attached storage for migration), and inter-data center connections, as shown in Fig. 3.8. In this figure, the network topology is modeled with three endpoints. Then, the data centers reside with a mesh backbone network topology with a full-duplex peer-to-peer global optical fiber link between each two switches, as well as in between each data center and switch [152, 153, 154]. Regarding data transfer, the network identifies a path with a bandwidth $B_{bb}(k)$ (for k^{th} path) to transmit the data between two data centers. For this purpose, I use the shortest path algorithm to determine the path between two endpoints, which yields faster data transfers [155]. In addition, the global links are modeled in the presence of Bit Error Rates (BERs), as well as their probabilities (P_{BER}) associated to the data transmission (i.e., required bandwidth), the speed of light, and distance between data centers.

Then, in order to compute the total latency for both migrating a set of VMs (according to VMs size) at time slot T and data communication during the time interval of $(T, T+1)$, from multiple data centers to a specific data center, two parts are taken into account. First, I use the local and global latency for the i^{th} source data center (i.e., L_l^i, $L_g^{i,j}$, and L_g^j, respectively) to transmit information through the local and global networks to the j^{th} destination data center. Second, I consider the local latency for the j^{th} destination data center (L_l^j) to transmit data collected from other data centers to its storage. As a result, Eq. 3.12 represents the

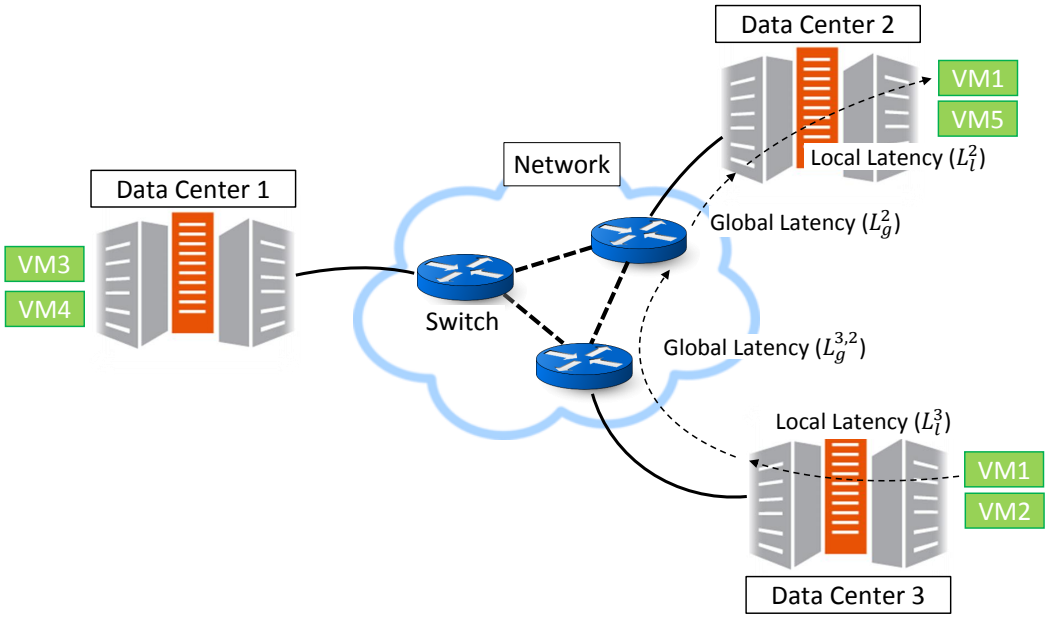

Figure 3.8 – The used geo-distributed data centers network model.

total (worst-case) latency for the j^{th} destination data center (L_t^j), which is the summation of the maximum latency between source data centers for transmitting the corresponding data through their dedicated local and global links (mesh topology) with the shortest path, and the local latency inside the destination data center. N_{DC} is the total number of data centers.

$$L_t^j = max_i(L_l^i + L_g^{i,j}) + L_g^j + L_l^j, \quad i = 1 \; to \; N_{DC} \; and \; i \neq j \tag{3.12}$$

Then, the local latency of the i^{th} source data center is dependent on the volume of data ($Vol^{i,j}$) ready to be transferred to the j^{th} destination data center and its local bandwidth (B_L^i). Moreover, for a fast data transfer, I use an all-bandwidth policy, which allocates all the available bandwidth of the link to carry the data [156]. Therefore, each source data center local latency is calculated as follows:

$$L_l^i = (Vol^{i,j})/B_L^i \tag{3.13}$$

Similarly, using the all-bandwidth policy, the local latency of the j^{th} destination data center is a function of the total volume of data received from the multiple source data centers and its local bandwidth (B_L^j), which is computed as follows:

$$L_l^j = \sum_{i=1, i\neq j}^{N_{DC}} Vol^{i,j}/B_L^j \tag{3.14}$$

The global latency includes the propagation latency as a primary source and data latency with respect to the required bandwidth (RBW) in terms of volume of data being transmitted. Then, the propagation latency is a function of how long it takes for the data to travel at the speed

Algorithm 4 Global Data Latency (L_e) with respect to BER

while true **do**
  $B_e(t) = (1 - BER(t)) \cdot B_{bb}$, $BER(t) \propto P_{BER(t)}$ and B_{bb} is the bandwidth of the shortest path
  if $RBW(Vol^{i,j}) \leq B_e(t)$ **then**
    $L_e = L_e + 1$
    $Break$
  else
    $RBW(Vol^{i,j}) = RBW(Vol^{i,j}) - B_e(t)$
    $L_e = L_e + 1$
  end if
end while

of light (S_l) from source to destination. Subsequently, the data latency (L_e) is a function of the effective bandwidth ($B_e(t)$) and the ($BER(t)$) (i.e., corrupted data must be resent). Hence, the global latency from i^{th} data center to closest switch to j^{th} data center (distance: $Dist_{i,j}$ based on the shortest distance) is calculated as follows:

$$L_g^{i,j} = Dist_{i,j}/S_l + L_e^{i,j} \tag{3.15}$$

In order to calculate the data latency ($L_e^{i,j}$) in the presence of transmission errors, I first calculate the effective bandwidth of the shortest path. Then, I use the necessary-bandwidth policy that allocates just enough bandwidth to the path with respect to the required bandwidth (RBW), resulting in more network availability and reducing power consumption of the routers and switches, as the switch power consumption has a linear model with respect to the bandwidth utilization ratio [156, 157]. Finally, I fragment the transmission into the necessary number of time steps. Algorithm 4 describes this process analytically. Similarly, from the last switch to j^{th} data center ($Dist_j$), the global latency (L_g^j) is computed with respect to the total bandwidth required from the multiple source data centers using the necessary-bandwidth policy [156].

3.5.2 Proposed Optimization Method

In this section, the problem of two-phase VM placement is first defined. Then, the proposed algorithm is presented.

3.5.2.1 Problem Definition

As shown in Fig. 3.9, the problem can be divided into two steps: i) clustering the VMs to dispatch them to the data centers (global controller), and ii) allocating clusters of VMs to servers withing a data center (local controller). At each time slot T, the global controller first receives the VMs' loads from the previous time interval $[T-1, T)$, data communication

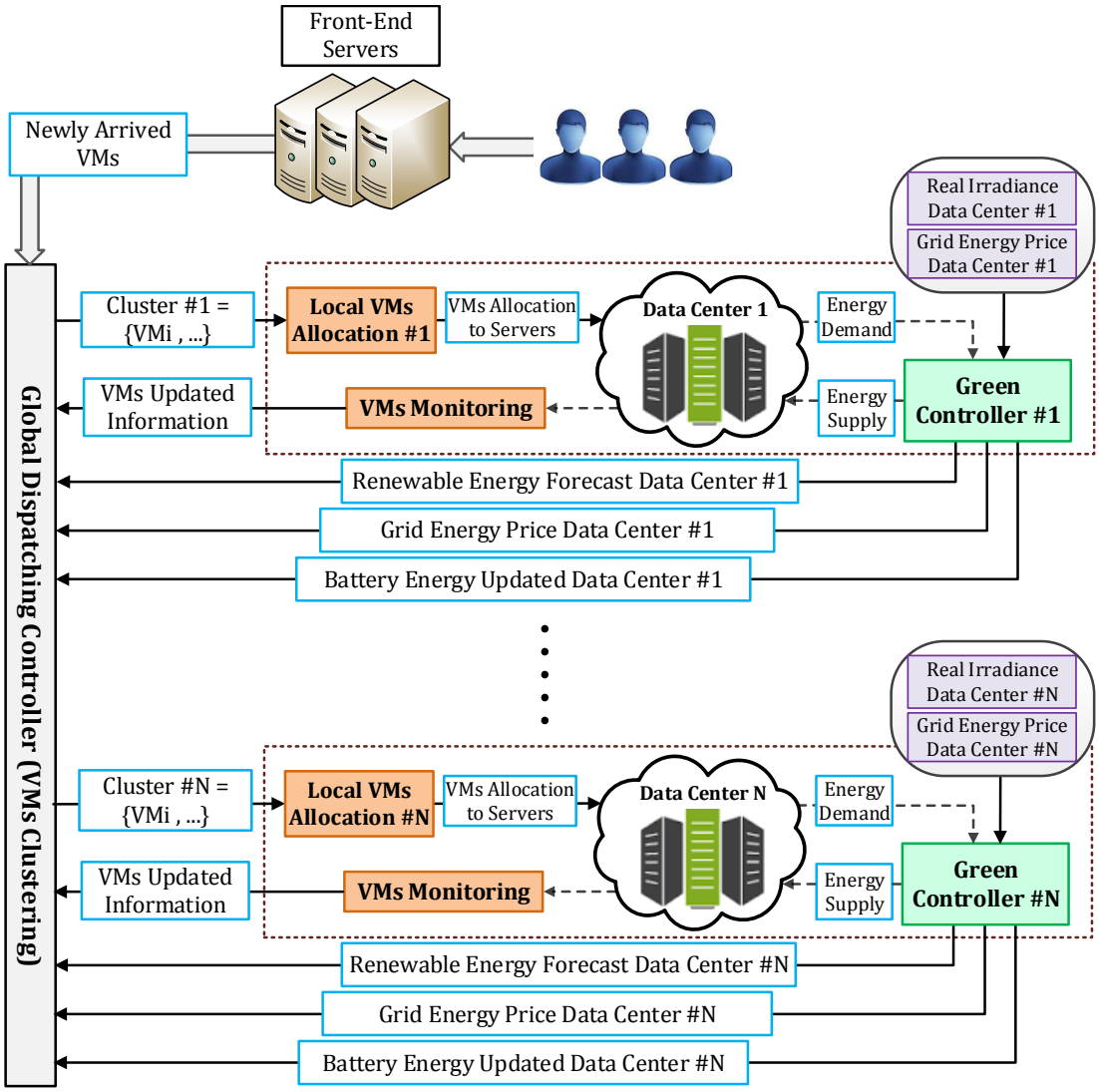

Figure 3.9 – Overview of proposed VM placement problem for green geo-distributed data centers.

patterns, renewable forecast, available battery energy and grid price from each data center; all of them are non-stationary parameters that change dynamically. Then, the VMs (available VMs in the system and newly arrived VMs) are clustered, for each data center. After clustering, at the local level, the VMs are allocated to the minimum number of servers possible. During the time interval of $[T, T+1)$, the local green controllers in each data center compensate the difference between real and forecasted load and renewable information.

3.5.2.2 Proposed VM Placement Algorithm

As optimal VM placement is an NP-complete problem, I propose a two-phase algorithm with low computational overhead that can be applied in real-time.

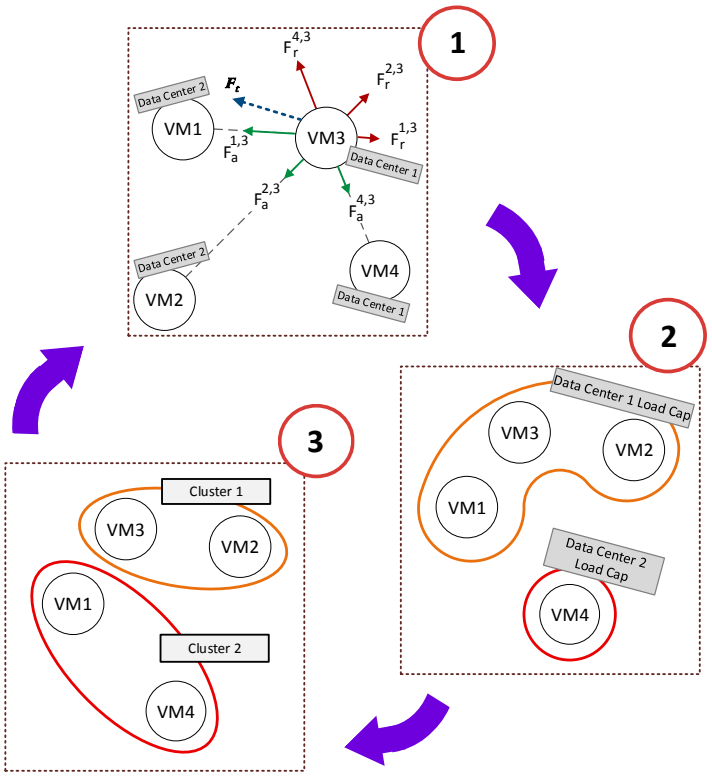

Figure 3.10 – Global phase - different steps for VMs clustering.

1) Global Phase - VMs Clustering

This phase is split into three different steps (Fig. 3.10 shows one example of different steps). First, at time slot T, all the VMs available in the system are represented as points in a two dimensional plane (2D plane). Based on the data and CPU-load correlation properties, as highly data-correlated VMs should be clustered together while highly CPU-load correlated VMs should be placed apart, a function is defined to calculate attraction and repulsion forces between each two VMs. Equation 3.16 calculates the force from i^{th} to j^{th} VM ($F_t^{i,j}$) as a function of attraction force ($F_a^{i,j}$) based on the data correlation ($Corr_{data}^{i,j}$) normalized as $[-1,0)$, and repulsion force ($F_r^{i,j}$) based on the CPU-load correlation ($Corr_{cpu}^{i,j}$) normalized as $(0,1]$. The attraction force from i^{th} to j^{th} VM is different from j^{th} to i^{th} VM due to the consideration of bidirectional data correlation and calculated as amount of data two VMs exchange. The repulsion force is computed as a worst-case peak CPU utilization when the peaks of two VMs coincide during the last time slot. α denotes a weighting factor for energy and performance trade-off calculation.

$$\begin{cases} F_a^{i,j} = Corr_{data}^{i,j} \\ \\ F_r^{i,j} = Corr_{cpu}^{i,j} \end{cases} \Rightarrow F_t^{i,j} = \alpha \,.\, F_a^{i,j} + (1-\alpha) \,.\, F_r^{i,j} \tag{3.16}$$

Initially, at time slot $T = 0$, all the points are distributed in the 2D plane. Then, the resultant forces in the X (F_x^i), and Y (F_y^i) directions are calculated amongst points ($\theta_{j,i}$ is the angle) and, as a result, the points are remapped in the 2D plane with new coordinates ($Loc_x^i(k), Loc_y^i(k)$) at each iteration k as follows:

$$\begin{cases} F_x^i = \sum_{j=1, j\neq i}^{N_{vm}} F_t^{j,i}.\cos(\theta_{j,i}) \\ F_y^i = \sum_{j=1, j\neq i}^{N_{vm}} F_t^{j,i}.\sin(\theta_{j,i}) \\ Loc_x^i(k) = Loc_x^i(k-1) \ + \ 0.5 \ . \ F_x^i(k) \ . \ t^2 \\ Loc_y^i(k) = Loc_y^i(k-1) \ + \ 0.5 \ . \ F_y^i(k) \ . \ t^2 \end{cases} \tag{3.17}$$

where N_{vm} and t denote the number of VMs (points) available in the system and time period of displacement, respectively, since: $\Delta x = \ 1/2 \ a \ . \ t^2$, where Δx and a indicate displacement and acceleration, respectively.

The process is iterated until the cost function ($Cost^{AR}$) of the current iteration k (Eq. 3.18) yields a lower value than that calculated in the previous iteration ($k-1$). A maximum number of iterations is also fixed to avoid a convergence time overhead. In this case, the algorithm provides a feasible solution for the problem especially when the correlation between all the VMs is either very low or high.

$$Cost_k^{AR} = \sum_{i=1}^{N_{vm}} \sum_{j=1, j\neq i}^{N_{vm}} F_t^{i,j}.(d_k^{i,j} - d_{k-1}^{i,j}) \tag{3.18}$$

where $d_k^{i,j}$ depicts the distance between i^{th} and j^{th} points at iteration k. This function computes whether there is either an attraction force between each pair of points ($F_t^{i,j} < 0$), and they are attracted to each other ($d_k^{i,j} - d_{k-1}^{i,j} < 0$), or a repulsion force ($F_t^{i,j} > 0$), and they separate away. The final location of all the VMs becomes the initial position for the next time slot.

In the second step, a energy capacity cap (in Joules) is first defined per each data center (cluster) to minimize the operational cost, computed according to the available battery energy, renewable energy forecast, grid price and data centers power consumed during the last previous time slot (i.e., last-value predictor).

Then, a modified version of the k-means algorithm is utilized to cluster VMs with respect to each cluster capacity cap, VMs load, and the distance between two VMs obtained from the repulsion and attraction phase in the 2D plane. In the modified k-means, the initial centroid of each cluster is calculated based on the last position of points available in that cluster in the previous time slot. In this step, network latency is not considered.

Finally the last step is to revise the modified k-means output to meet the hard time constraint for migrating VMs across data centers based on their size as described in Algorithm 5. The output of the modified k-means creates two queues per cluster (data center): *outgoing* and

Algorithm 5 Migration Step - Modified K-means Output Revision

Input: Outgoing and incoming queues
Output: VMs migration actions

1: $Q^i_{out} \leftarrow$ Sort i^{th} data center outgoing queue based on VMs distances from its centroid (descending order)
2: $Q^i_{in} \leftarrow$ Sort i^{th} data center incoming queue based on VMs distances from its centroid (ascending order)
3: $i \leftarrow 1$ Initial data center
4: **while** (Q_{in} and Q_{out} are not $NULL$ for all data centers) & (Latency constraint is not violated for all connections) **do**
5: **if** $R_i < Cap_i$ **then**
6: VM = Head (Q^i_{in})
7: $j \leftarrow$ Current data center of VM
8: **if** $L^i_t <$ Latency constraint **then**
9: Migrate VM from j^{th} to i^{th} data center
10: Update i^{th} and j^{th} data centers' load (R_i and R_j)
11: **end if**
12: Erase VM from Q^i_{in} and Q^j_{out}
13: **else if** $R_i \geq Cap_i$ **then**
14: VM = Head (Q^i_{out})
15: $j \leftarrow$ Destination data center of VM
16: **if** $L^j_t <$ Latency constraint **then**
17: Migrate VM from i^{th} to j^{th} data center
18: Update i^{th} and j^{th} data centers' load ($R_i \ and \ R_j$)
19: Erase VM from Q^i_{out} and Q^j_{in}
20: $i \leftarrow j$ Move to destination data center
21: **else**
22: Erase VM from Q^i_{out} and Q^j_{in}
23: **end if**
24: **end if**
25: **end while**

incoming. The first one contains the candidates to be migrated outside, to another data center, sorted in descending order according to their distances from the corresponding cluster's centroid (Q_{out}). The second one contains the candidates to be migrated to this data center sorted in ascending order (Q_{in}).

The algorithm first selects one data center (i^{th} data center) and checks if its previous load (R_i) is less than its capacity cap (Cap_i). Then, it selects the first VM from the head of the incoming queue of the cluster (Head(Q^i_{in})). If this VM can be migrated in less than *latency constraint* time, the migration is executed; otherwise, it is erased from the queue and the next VM is selected. I repeat and update the data center's load until there is either no VM to accept or the load of the data center becomes more than the cap (lines 5~12). In this later case (lines 13~24), the VM is selected from the head of the outgoing queue of the current cluster

(Head(Q^i_{out})) which has the maximum distance to the centroid. If this VM can be migrated, I check the current load of the destination cluster and repeat this process there. Otherwise, the next one in the cluster is selected. This algorithm iterates until the latency constraint violated for all data centers or there is no action to do. Unallocated VMs that have been available in the system will stay in their previous data center, and unallocated new VMs are assigned to the data centers determined from the modified k-means step without the consideration of the network latency constraint. In this case, it is tried to find the best solution for migrating the appropriate VMs when the number of migrations is bounded. This method also prevents network bottlenecks made by one data center when the other data centers need to migrate their VMs to the same destination data center.

2) Local Phase - VMs Allocation

In the local phase, the VMs of each cluster are allocated to servers of their corresponding data center, and the optimal frequency for each server is computed. CPU-load correlation is only used to allocate VMs to the minimum number of servers, since data correlation (and migrations) mainly contribute to inter-data center network bottlenecks [19, 30]. Hence, I base the implementation on one of the best algorithms in the state-of-the-art [20] for VMs allocation.

3) Low-Complexity Rule-Based Green Controller

The proposed VM placement algorithm reduces the dependency on grid energy using batteries and renewable energy. Therefore, a low-complexity green controller is required to compensate the difference between real and forecasted information with respect to the current electricity price of data centers. Differently from the previous two-phase green energy controller presented in Section 3.4.1.2, a rule-based greedy heuristic is used regardless of offline phase and high-complexity DP algorithm.

After allocating all the VMs to servers at time slot T, the green controller inside each data center manages the energy sources during the time interval of $[T, T+1)$ based on the real renewable energy and data center energy consumption. When the available renewable energy is higher than the data center energy consumption, this free energy is used to power the data center and the excess energy is stored in the battery bank. On the contrary, during the high price period, the renewable energy is used to power the data center's, and if more energy is needed, the battery is discharged considering its DoD. During the low price periods, the battery is charged via the grid energy and it is not used for the data center.

3.5.3 Proposed Method Performance Evaluation

In this section I first present the simulation setup. Then, I evaluate the efficiency of the proposed algorithm by comparing it to recent data center management techniques.

Table 3.3 – Data centers' number of servers and energy sources specification.

Data Center	Number of Servers	PV Capacity (KWp)	Battery Capacity (KWh)
data center 1	1500	150	960
data center 2	1000	100	720
data center 3	500	50	480

3.5.3.1 Experimental Setup

I consider three different data centers located in Europe: Lisbon (data center 1), Zurich (data center 2) and Helsinki (data center 3), along with their distances (for the network model), time zone and two-level real electricity price scenario. Each data center contains 10 rooms and, each room, has 150, 100 and 50 servers for data center 1, data center 2 and data center 3, respectively. Table 3.3 summarizes the number of servers, PV module size and lithium-ion battery capacity (with 50% of DoD, keeping the remaining capacity in case of outage) per data center. I target an Intel Xeon E5410 server consisting of 8 cores and two frequency levels (2.0GHz and 2.3GHz), and use the power model in [149]. For cooling power consumption, a time-varying PUE model is used, as presented in [100]. The data centers are connected through a mesh topology (as shown in Fig. 3.8) with 100 Gb/s full-duplex peer-to-peer optical fiber links, and the intranet uses 10 Gb/s full-duplex links. Finally, the global links experience a BER that is chosen randomly from the following distribution: 54% probability of 10^{-6}, 20% of 10^{-5}, 15% of 10^{-4}, 10% of 10^{-3}, and 1% of 10^{-2} [158].

In order to simulate a realistic scenario, data center VMs and energy demand, the VMs' utilization of a real data center has been sampled every 5 seconds for one day, and it has been extended to 7 days by adding statistical variance with the same mean as the original traces. For renewable forecast, I implemented the algorithm in previous work [9].

Arrival and life-time of each VM, given in time slots, are generated by poisson and exponential distributions, respectively. Data correlation between each pair of VMs is generated by a lognormal distribution with the mean of 10 MB and uniform variance selection in the range of $[1,4]$ [102]. For migration, the size of the VMs are in the range of 2, 4, and 8 GB according to the distribution of 60%, 30%, and 10%.

Finally, the global and local controllers are invoked every hour, and the green controller in each data center is invoked every 5 seconds. Also, a hard time constrain (latency constraint in Algorithm 5) is taken into account for migrating the VMs across data centers through the network. A value of 98% for the QoS guarantees that the migration of VMs will take less than the 2% of the time slot.

3.5.3.2 Experimental Results

I compare the proposed algorithm against three state-of-the-art approaches that are the best in their class to optimize operational costs, energy consumption and performance, respectively:

- Cost-aware approach (*Pri-aware*) [41].
- Energy-aware VM allocation (*Ener-aware*) [20].
- Network-aware VM placement (*Net-aware*) [16].
- *Proposed:* the proposed multi-objective VM placement.

All the mentioned methods are used jointly with the same local green controller to manage battery and renewable energy.

1) Operational Cost Analysis

Figure 3.11 shows the operational cost normalized to the worst-case value among the mentioned methods for a time horizon of one week. The obtained cost savings are 54%, 23% and 34% for the proposed method compared to Ener-aware, Pri-aware and Net-aware, respectively. Proposed clusters the VMs by specifying a load cap for different data centers based on the power grid price and available renewable and battery energy. It outperforms the other algorithms when a local energy-aware VM allocation method is utilized to further reduce data centers' dependency on grid energy. Differently, Ener-aware uses CPU-load correlation to reduce energy consumption and cost in each data center locally but, globally, it cannot efficiently cluster and dispatch VMs to the right data centers based on available renewable energy, battery status and grid price. In Pri-aware, the VMs are packed and placed onto data centers and servers with the lowest current grid price, but it neglects to maximize free energies usage. Finally, the Net-aware approach provides load balancing across data centers which in turn leads to better exploiting free energies (renewable and battery) compared to Ener-aware

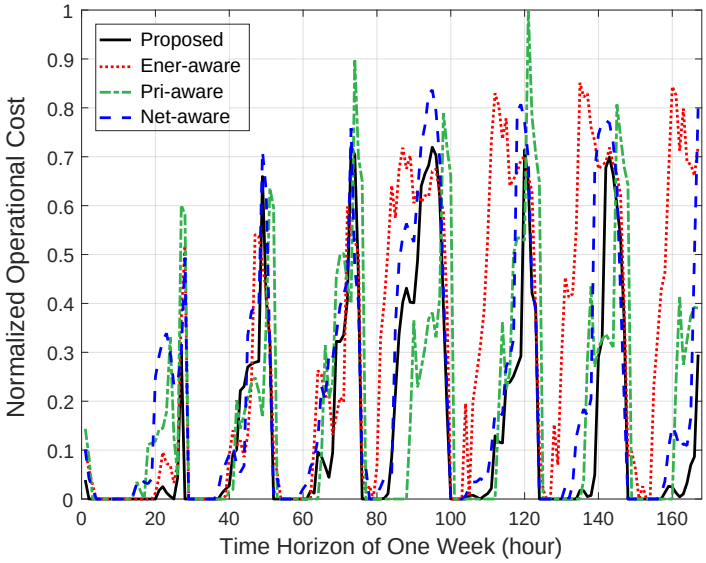

Figure 3.11 – Normalized operational cost for a time horizon of one week.

and Pri-aware. However, this algorithm does not consider the electricity price diversities and neglects to utilize an energy-efficient management to reduce its dependency on the grid.

2) Energy Consumption Analysis

Figure 3.12 shows the hourly energy consumed by the data centers for one week. The total energy consumption is 57.1 GJ, 54.3 GJ, 65.3 GJ and 66.3 GJ for the Proposed, Ener-aware, Pri-aware and Net-aware methods, respectively. The results show 13% and 14% energy improvements for the proposed algorithm with respect to Pri-aware and Net-aware, due to the consideration of the CPU-load correlation between VMs, which places highly CPU-load correlated VMs apart, i.e., in different data centers and servers. This favors consolidation and leads to power savings by lowering the number of active severs and their operating frequency. On the other hand, the Ener-aware approach first uses the First-Fit-Decreasing clustering heuristic, placing VMs into the first data center in which its load capacity fits, and then packs the VMs into the minimal number of active servers based on the CPU-load correlation. Hence, the data center local controller finds a better mapping of VMs to servers when most of the VMs are in the same data center. The presented algorithm, however, tries to find the best VMs clusters per each data center based on the CPU-load and data correlations and determined data centers' capacity cap. Although these correlations indicate opposed goals for energy and performance for the proposed algorithm, Ener-aware, which focused on energy optimization, only obtains a 4.9% energy improvement compared to the proposed multi-objective algorithm, while significantly degrading operational costs and performance (i.e., response time as shown in the next section).

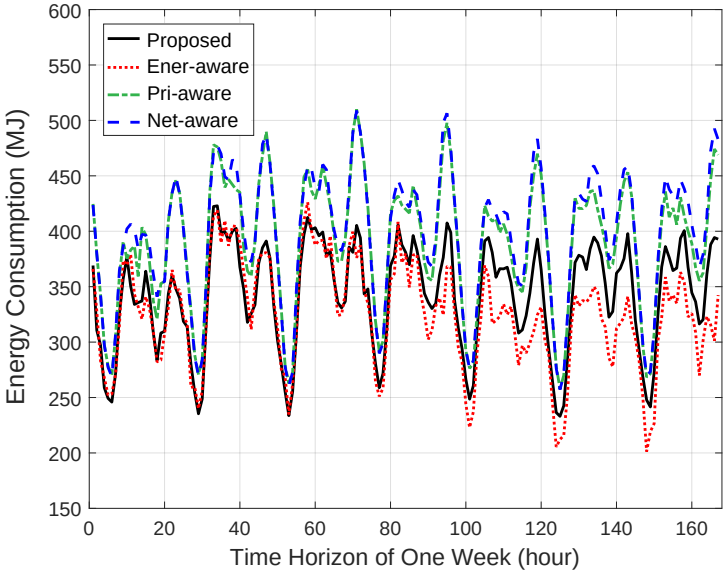

Figure 3.12 – Energy consumed by data centers for a time horizon of one week.

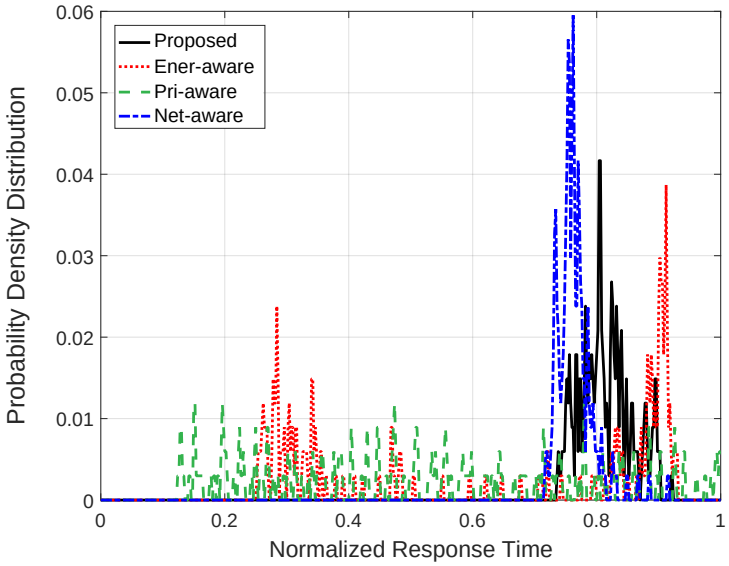

Figure 3.13 – Probability distribution of normalized response time in one week.

3) Performance Analysis

In this context, performance is defined as the response time of the VMs; i.e., the amount of time they have to wait for data from other VMs in the network. Figure 3.13 shows the probability density distribution of the response time in one week. Note that the response time results are normalized with respect to the worst-case value among the methods. As a result, Proposed and Net-aware encompass a range of response time with higher average and lower variance compared to Ener-aware and Pri-aware methods. The goal of Net-aware is to

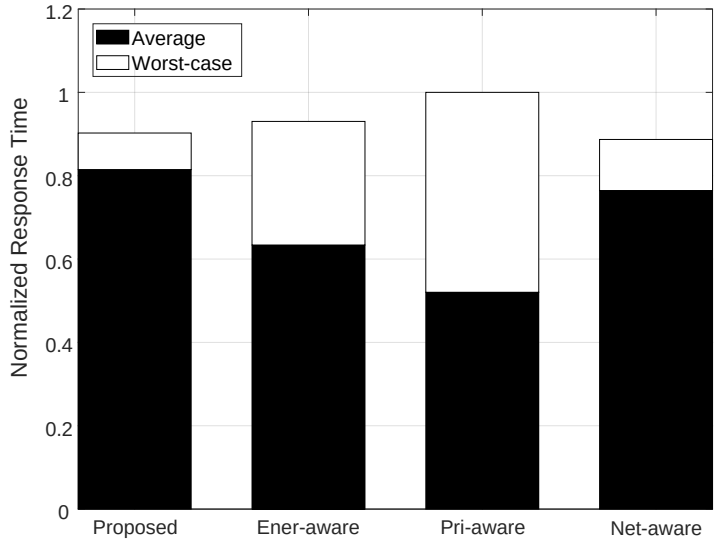

Figure 3.14 – Average and worst-case of response time in one week.

balance the network across data centers, which in turn leads to better worst-case and higher average response time (both for times of high and low data demands between VMs). However, when compared to Proposed, Net-aware only achieves 4% performance improvement on average. Ener-aware and Pri-aware tend to place the VMs on a lower number of data centers, which leads to unbalanced network traffic with bigger fluctuations and, accordingly, lower average response time. However, as shown in Fig. 3.14, since data centers providers typically consider worst-case response time in their SLA contracts, the proposed algorithm results in up to 10% performance improvement compared to state-of-the-art approaches. Also, the Net-aware method (only optimized for performance) only achieves up to 1.7% performance improvement compared to the proposed method.

4) Trade-Offs Discussion

The experimental results confirm that, by having a holistic approach, better trade-offs can be obtained in the problem of VM placement. Figures 3.15, 3.16 and 3.17 summarize the benefits of Proposed: In the first place, Fig. 3.15 depicts the totals, showing up to 54%, 14% and 10% improvements for operational cost, energy consumption and performance, respectively. Then, Fig. 3.16 shows the cost-performance trade-off, with Proposed providing 23% and 10% improvements for cost and response time, respectively, compared to Pri-aware. In comparison with Net-aware, it achieves 34% cost savings, while it leads to only 1.7% performance degradation. Finally, Fig. 3.17 presents the obtained energy-performance trade-off: the proposed algorithm results in 5% performance improvement with a 4.9% energy overhead compared to Ener-aware; and it provides 14% energy savings and 1.7% performance degradation compared to Net-aware.

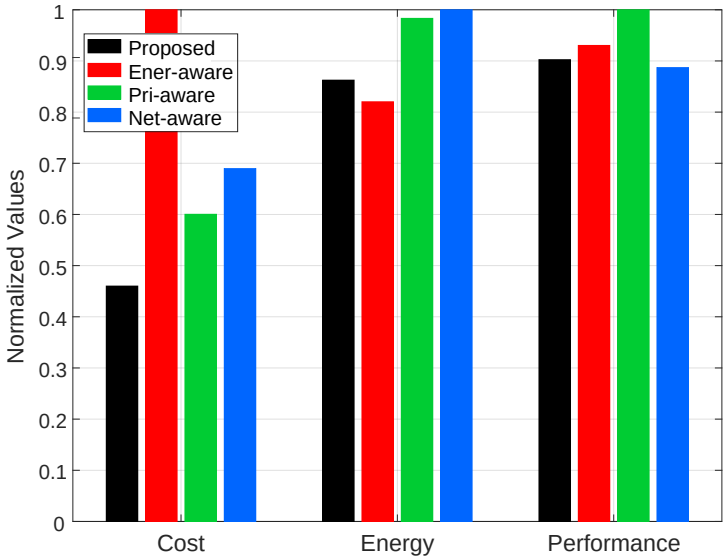

Figure 3.15 – Total cost, energy and performance for the Proposed algorithm and the other state-of-the-art approaches.

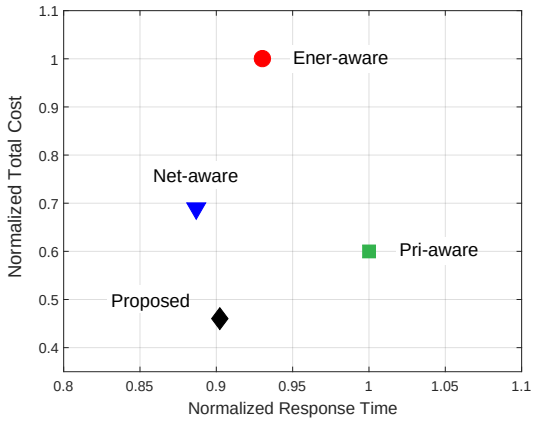

Figure 3.16 – Cost-Performance trade-off analysis for the Proposed algorithm and the other state-of-the-art approaches.

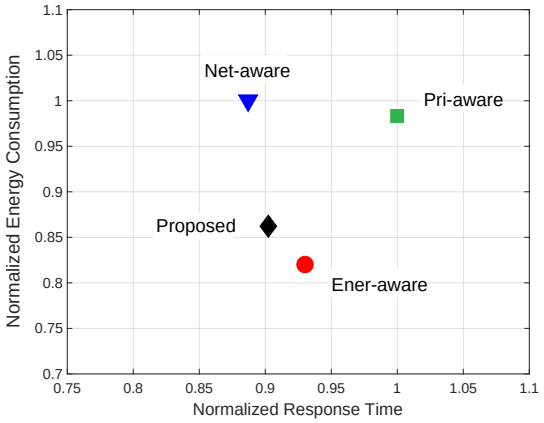

Figure 3.17 – Energy-Performance trade-off analysis for the Proposed algorithm and the other state-of-the-art approaches.

3.6 Electricity Cost Optimization for Green Data Centers in Emerging Power Markets

Power market operators have recently introduced smart grid demand-response programs, in which electricity consumers regulate their power usage following provider requirements [68]. Among the various types of capacity reserves, RS reserves [65] are particularly interesting for green data centers due to the relatively high value of such reserves and capabilities of data centers for providing high flexibility in their power consumption. In RS reserves provision, the demand-side (i.e., green data center) must dynamically modulate its power consumption to follow an RS signal broadcasted by the ISO every few seconds. In this scenario, the demand-side acts as a capacity reserve that stabilizes the ISO power from the intermittency of renewable energies, and benefits from the power market rewards. However, the demand-side itself is also affected by the instability of on-site renewables.

In this section, I introduce ECOGreen, a new Electricity Cost Optimization strategy for Green data centers that computes the best average power and reserve bidding problem considering the renewable and EES energy for RS reserves provision in emerging power markets, along with determining the number of active servers for VM allocation phase. To this end, I consider both time-changing trends of renewable energy sources and power loss in battery bank due to aging and charging sequences. Finally, an online policy is developed that enables a green data center to regulate its power and track the RS signal broadcasted every few seconds accurately, while also guaranteeing QoS constraints.

3.6.1 Problem Description

In this section I provide a description of the overall scenario, the system that is optimized, and the main assumptions taken. Figure 3.18 illustrates the proposed scenario and strategy for

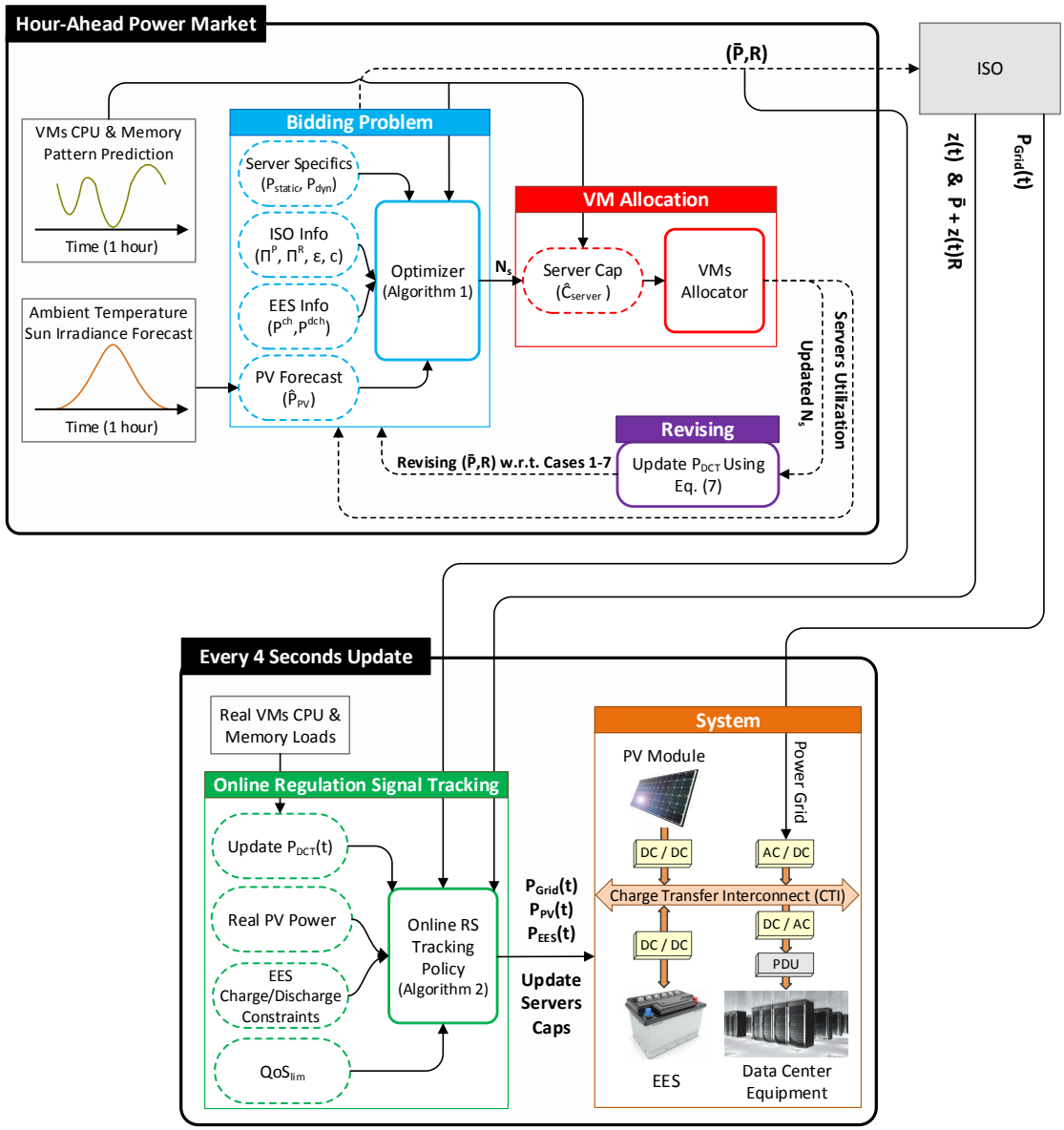

Figure 3.18 – Overall diagram of the proposed scenario and strategy, i.e., ECOGreen, including Bidding Problem, Allocation, Revising Bidding Values, and Online Policy phases.

participating in the power market. Figure 3.18-System (bottom right block - the same electric system used in Section 3.3) shows the green data center system, which comprises a green data center equipped with on-site (demand-side) renewable energy (PV modules), and an EES system, interconnected between them and to the power grid (ISO) via a CTI bus.

From the green data center perspective, in this work the EES is used to provide both supply in case of grid outages and a buffer for green and grid power provided by the ISO. In addition, it

is assumed that exceeding renewable energy cannot be injected back into the power grid (i.e., it can only be used or stored). Thus, renewable energy and EES are used to track the RS signal coming from the ISO or, at least, provide supply during outages.

From the emerging power market perspective, large ISOs (such as PJM) allow the demand-side to provide reserves. I focus on RS reserves in the hour-ahead power market as the price of reserves is high and the green data center can modulate its power at this timescale. For each participant in the RS reserves power market, an average power consumption $\bar{P}$ and reserve provision R should be declared to the ISO an hour in advance. The participant is charged for its average energy consumption and credited for the provided reserves. However, to be given a certain credit, each hour the participant is asked to modulate its power consumption dynamically to track the RS signal (z). $z(t)$ is updated every 4 seconds in increments that do not exceed $\pm R/(\tau/4)$, where τ is 150 seconds for the fast RS and 300 seconds for the slow RS [129]. The RS signal (i.e., $z(t)$) is the main factor used by ISO to balance the supply and demand in the power grid. Part of the RS credit is reduced based on the magnitude of the tracking error. Moreover, if the tracking error exceeds a statistical tolerance constraint, the participant (i.e., data center) may lose its contract [130].

In order to bid in the hour-ahead market (i.e., to find the values of $\bar{P}$ and R), the ECOGreen strategy requires predicting, at the beginning of time slot T (i.e., every hour), sun irradiance forecasts, and per-VM CPU and memory utilization patterns ($\tilde{U}_{cpu}$ and $\tilde{U}_{mem}$). Given the daily periodicity observed in the VMs of Google Cluster traces, this can be achieved by using the Autoregressive Integrated Moving Average (ARIMA) prediction model [159]. ARIMA considers the CPU and memory utilization from the previous week and forecasts the next-hour traces per VM.

Given the predicted VMs workloads, renewable energy forecasts, and battery status, I first compute the $(\bar{P}, R)$, number of active servers (N_s), and the per-server capacity cap ($\hat{C}_{server}$) provided for the VM allocation phase. Then, the VMs are mapped to the servers based on $\hat{C}_{server}$ every hour. Before sending bidding values to ISO, it is needed to recompute $(\bar{P}, R)$ with respect to the estimated power consumption of the data center ($\hat{P}_{DCT}$). This recalculation is only required once, and aims at guaranteeing that all VMs fit within the number of active servers determined during the allocation phase (N_s).

Once the ISO approves the bid, $z(t)$ is dynamically broadcasted to the green data center from the ISO every 4 seconds. Therefore, an online policy is needed for tracking the power imposed by the ISO (i.e., $P_{Grid}(t) = \bar{P} + z(t)R$) with the smallest tracking error possible, given the real data center status (i.e., real VM's CPU utilization, memory footprint, and renewable power). I also aim to optimize the usage of renewable and battery power under the above-mentioned system constraints. To propose a solution to both the bidding in the hour-ahead power market and the tracking of the RS signal using a 4-second update, an accurate system model is required.

3.6.2 Green Data Center Modeling

In the following subsections I first present the used system model, which is essential for correctly assessing the battery lifetime as well as specifying the demand-side PV power generation. Next, I detail the considered green data center power model tackled in this section.

3.6.2.1 System Modeling

I use all the models for the green data center system (as shown in Fig. 3.18-System), presented in Section 3.3. The power management is done on CTI bus, where data center equipment, PV modules, EES, and power grid are connected via a bidirectional CTI bus. The considered system comprises one homogeneous EES (battery bank) managed in a hierarchical fashion. The battery model is based on Peukert's law [137], and has been conceived as a plug-and-play component that can be easily replaced and adapted. To correctly account for the benefits of the EES system, it is needed to control the charging sequences of the battery bank, and to model battery aging (i.e., power loss in EES).

From the battery technology viewpoint, while lead-acid technology is cheaper, easier to recycle and has a wider working temperature range, it suffers from a limited number of sustainable cycles (i.e., lifetime). Therefore, the lithium-ion technology is instead chosen, which offers at least one order of magnitude higher number of cycles (useful for tracking fast time-changing RS signal), but at a higher cost. To maximize the lifetime of the storage and make the battery bank work in the optimal range of SoC, the maximum DoD is constrained to 70%. The remaining capacity is however available in the event of an outage, thus providing standard UPS support.

The PV module size (the number of cells and panels) is also tuned considering the peak power of the green data center with respect to its load, which is the most common approach to PV sizing [145].

3.6.2.2 Data Center Power Model

Total data center power consumption (P_{DCT}) is modeled as the sum of the power of servers: $P_{DCT} = \sum_{j=1}^{N_s^{DCT}} P_j$, where P_j and N_s^{DCT} specify the j^{th} server power and the total number of servers in the data center, respectively. Following the same methodology than in previous research [96, 97], the major contributors to power consumption in servers are the CPU, memory, fans and disks. Among these, CPU has the largest effect on power, and previous research shows that the power-frequency relation is linear for a given CPU-intensive workload [98].

Hence, server power can be calculated as [96]:

$$\begin{aligned} P_j &= P_{j_{static}} + P_{j_{dyn}} \\ \begin{cases} P_{j_{static}} = P_{disk} + P_{fan} + P_{cpu}^{leak} + P_{cpu}^{idle} + P_{mem}^{idle} \\ P_{j_{dyn}} = P_{cpu}^{dyn} \cdot (U_{cpu_j}/100) + P_{mem}^{dyn} \cdot (U_{mem_j}/100) \end{cases} \end{aligned} \tag{3.19}$$

where $P_{j_{static}}$ indicates all the contributions to power that are workload-independent. P_{disk} and P_{fan} are considered constants for the considered workload, and respectively account for the power consumption of disks and fans. P_{cpu}^{leak} refers to temperature-dependent leakage power. A high fan speed and a low inlet temperature are considered to reduce the effect of temperature-dependent leakage power, taking it into account as its worst-case constant. P_{cpu}^{idle} and P_{mem}^{idle} are constants that account for the idle power consumption of CPU and memory respectively. $P_{j_{dyn}}$ accounts for server dynamic power, and is proportional to the CPU and memory utilization (U_{cpu_j} and U_{mem_j}, respectively). P_{cpu}^{dyn} and P_{mem}^{dyn} are fitted constants obtained under the same experimental conditions used in previous research [96] for the same set of CPU-intensive workloads.

3.6.3 ECOGreen: Electricity Cost Optimization Strategy for Green Data Center

3.6.3.1 Bidding Solution

1) General Problem Statement

The first step in the RS hour-ahead power market is to compute the bidding for average power and reserves ($\bar{P}, R$) for every 1-hour time slot (T) (see Fig. 3.18). The best bid is the one that minimizes the monetary costs. This is achieved by trying to match $\bar{P}$ and R while reducing the tracking error on the 4-second RS signal at each time t. As the data center is equipped with on-site renewable power and EES, the power that needs to be provided by the grid at each time t (i.e., every 4 seconds), can be estimated based on the predicted data center power ($\hat{P}_{DCT}(t)$), the forecasted renewable power ($\hat{P}_{PV}$), and the current EES charge (P_{EES}), as follows:

$$\hat{P}_{Grid}(t) = \hat{P}_{DCT}(t) - \alpha \cdot P_{EES}(t) - \hat{P}_{PV}(t), \quad \alpha = \pm 1 \tag{3.20}$$

where $\hat{P}_{DCT}(t)$ is predicted based on the VMs loads (using the ARIMA model) and depends on the number of active servers (N_s). Due to the large time delay of booting a server, it is assumed that during the time slot neither the active servers are shut down nor new servers are turned on. That is, N_s is decided at the beginning of each time slot T and remains unchanged during it.

After finding the bidding values, during the time slot and as the real values for data center power and renewable energy reveal, it is needed to modulate the data center power and manage the green energy to track the RS signal (see Sec. 3.6.3.4). That is, to minimize error

(and subsequently cost), I must ensure that $\hat{P}_{Grid}(t) \approx \bar{P} + z(t)R$ for every t.

In this scenario, and without loss of generalization, the electricity cost minimization problem for every t during the time slot T can be formulated as in recent work [160], as follows:

$$\min_{\bar{P},R} \quad Cost(t) = \pi^P \bar{P} - \pi^R R + \pi^R c \frac{|\hat{P}_{Grid}(t) - (\bar{P} + z(t)R)|}{R} \tag{3.21}$$

$$Subject\ to$$

$$1.\ \bar{P} + R \leq N_s \cdot P_s^{max} + P_{EES}^{ch}(t) = a \tag{3.22}$$

$$2.\ \bar{P} - R \geq max(N_s \cdot P_{s_{static}} + \hat{P}_{DCT_{dyn}}(t) - \hat{P}_{PV}(t) - P_{EES}^{dch}(t),\ 0) = b \tag{3.23}$$

$$3.\ \frac{|\hat{P}_{Grid}(t) - (\bar{P} + z(t)R)|}{R} \leq \epsilon \tag{3.24}$$

$$4.\ \bar{P} \geq 0 \tag{3.25}$$

$$5.\ R \geq 0 \tag{3.26}$$

The minimization problem (Eq. 3.21) is subject to the constraints given in Eq. 3.22 to 3.26, where π^P is the hour-ahead price of power and π^R is the hour ahead price of reserves, both in \$/kWh, and c is the penalty coefficient on the second moment of the tracking error. As in literature, it is assumed that $\pi^P \approx \pi^R$ [69].

Constraints 1 and 2 limit the upper and lower bound of the bidding. These limits are a function of the number of active servers (Eq. 3.22 and 3.23). Upper limit (a) is computed based on the maximum data center power, given the number of turned-on servers (N_s), the per-server maximum power (P_s^{max}), and the amount of power that can be injected into the battery from the power grid. Lower bound (b) shows the excess power that cannot be provided by renewable and battery sources. The tracking error is measured during the hour by Constraint 3. Part of the credit (i.e., $\pi^R R$) is reduced proportionally to the tracking error. The reserve provider may lose its contract in further RS reserves provision if the tracking error exceeds a limit (i.e., ϵ). Finally, Constraint 4 and 5 ensure that $(\bar{P}, R)$ do not take negative values.

2) Specific Solution for ECOGreen

The previous formulation describes the general cost function and constraints for the problem. In what follows, I present a specific formulation for the worst-case scenario (i.e., the case when $z(t) = 1$ or $z(t) = -1$), to avoid losing the contract due to a too large error in following the RS signal.

Within a time slot, the worst-case error is achieved when the following conditions are met: i) data center aggregated workload is maximum ($\hat{P}_{DCT_{dyn}}^{max}$, i.e., the utilization peaks of VMs coincide at the same time); and ii) renewable energy generation is at a minimum. In such

scenario, b can be rewritten as:

$$b = max(N_s \cdot P_{s_{static}} + \hat{P}^{max}_{DCT_{dyn}} - \min_t(\hat{P}_{PV}(t)) - P^{dch}_{EES}(t),\ 0) \tag{3.27}$$

Also, as in this case $z(t) = 1$ or $z(t) = -1$, Constraint 3 can be expressed as:

$$\begin{cases} (\bar{P} + z(t)R) - \hat{P}_{Grid}(t) \leq \epsilon R \ \rightarrow \ \bar{P} + (1-\epsilon)R \leq a \\ \hat{P}_{Grid}(t) - (\bar{P} + z(t)R) \leq \epsilon R \ \rightarrow \ \bar{P} - (1-\epsilon)R \geq b \end{cases} \tag{3.28}$$

Hence, Constraints 1, 2 and 3 can be simply replaced by these two new constraints. Therefore, the final problem is as follows:

$$\min_{\bar{P},R} \quad Cost(t) = \pi^P \bar{P} - \pi^R R + \pi^R c \frac{|\hat{P}_{Grid}(t) - (\bar{P} + z(t)R)|}{R} \tag{3.29}$$

$$Subject\ to$$

$$1.\ \bar{P} + (1-\epsilon)R \leq a \tag{3.30}$$

$$2.\ \bar{P} - (1-\epsilon)R \geq b \tag{3.31}$$

$$3.\ \bar{P} \geq 0 \tag{3.32}$$

$$4.\ R \geq 0 \tag{3.33}$$

3) Solving The Worst-Case Scenario

The previous minimization is solved by using the derivative method, in order to obtain the best solution for the whole time slot (integral of the objective function, i.e., Eq. 3.29, over 1 hour). Since the statistical properties of $z(t)$ is only known but its value is unknown, the problem is solved for the worst-case scenario, which leads to two cases: i) $z(t) = -1$, and ii) $z(t) = 1$.

Worst-case scenario 1: $z(t) = -1$

When $z(t) = -1$ the absolute value of Eq. 3.29 is $\hat{P}_{Grid}(t) \geq \bar{P} + z(t)R,\ \forall t \in T$, and it can be written as:

$$\int_t^{t+T} Cost(t)dt = \int_t^{t+T} (\pi^P \bar{P} - \pi^R R)dt + \int_t^{t+T} \pi^R c(\frac{\hat{P}_{Grid}(t)}{R})dt$$
$$- \int_t^{t+T} \pi^R c(\frac{\bar{P}}{R})dt - \int_t^{t+T} \pi^R c z(t)dt \tag{3.34}$$

As $z(t)$ is the real-time broadcaster power market signal, and it takes a real number in the interval [-1, 1] with an average of zero over longer time intervals [69], the aggregated cost can

be computed for the whole 1 hour time slot as:

$$Cost^{agr} = \pi^P \bar{P} T - \pi^R R T + \pi^R c(\frac{\hat{P}^{agr}_{Grid}}{R}) - \pi^R c(\frac{\bar{P}}{R}) T \tag{3.35}$$

To find the solution, the first and second derivatives with respect to $\bar{P}$ and R are computed as follows:

$$\begin{cases} \frac{\partial Cost^{agr}}{\partial \bar{P}} = \pi^P T - \frac{\pi^R c T}{R} = 0 \;\rightarrow\; R = \frac{\pi^R c}{\pi^P} \\ \frac{\partial^2 Cost^{agr}}{\partial \bar{P}^2} = 0 \end{cases}$$
$$\begin{cases} \frac{\partial Cost^{agr}}{\partial R} = -\pi^R T - (\frac{\pi^R c \hat{P}^{agr}_{Grid} - \bar{P}\pi^R c T}{R^2}) = 0 \;\rightarrow \\ R = [c(\bar{P} - \frac{\hat{P}^{agr}_{Grid}}{T})]^{\frac{1}{2}} \;\rightarrow\; \bar{P} = c(\frac{\pi^R}{\pi^P})^2 + \frac{\hat{P}^{agr}_{Grid}}{T} \\ \frac{\partial^2 Cost^{agr}}{\partial \bar{P}^2} = \frac{2\pi^R c \hat{P}^{agr}_{Grid} - \bar{P}\pi^R c R T}{R^3} \end{cases} \tag{3.36}$$

In this case, $(\bar{P}, R)$ is a saddle point, since $\frac{\partial^2 Cost^{agr}}{\partial \bar{P}^2}\frac{\partial^2 Cost^{agr}}{\partial R^2} - (\frac{\partial^2 Cost^{agr}}{\partial \bar{P}\partial R})^2 < 0$. Therefore, it is also needed to check the boundaries to find the solution under the worst-case scenario and assumptions. According to the defined constraints, there are four different explicit solutions considering these four boundaries: 1) $\bar{P} = b + (1-\epsilon)R$, 2) $\bar{P} = a - (1-\epsilon)R$, 3) $R = 0^+$, and 4) the intersection point of boundaries 1 and 2, as follows:

Case 1: $\bar{P} = b + (1-\epsilon)R$

$$\begin{cases} R = [\frac{\pi^R c(\frac{\hat{P}^{agr}_{Grid}}{T} - b)}{(1-\epsilon)\pi^P - \pi^R}]^{\frac{1}{2}} \\ \bar{P} = b + (1-\epsilon)[\frac{\pi^R c(\frac{\hat{P}^{agr}_{Grid}}{T} - b)}{(1-\epsilon)\pi^P - \pi^R}]^{\frac{1}{2}} \end{cases} \tag{3.37}$$

Case 2: $\bar{P} = a - (1-\epsilon)R$

$$\begin{cases} R = [\frac{\pi^R c(a - \frac{\hat{P}^{agr}_{Grid}}{T})}{(1-\epsilon)\pi^P + \pi^R}]^{\frac{1}{2}} \\ \bar{P} = a - (1-\epsilon)[\frac{\pi^R c(a - \frac{\hat{P}^{agr}_{Grid}}{T})}{(1-\epsilon)\pi^P + \pi^R}]^{\frac{1}{2}} \end{cases} \tag{3.38}$$

Case 3: $R = 0^+$

$$\begin{cases} \bar{P} = a \\ R = 0 \end{cases} \tag{3.39}$$

Case 4: Finally for the intersection point of Cases 1 and 2:

$$\begin{cases} R = \frac{a-b}{2(1-\epsilon)} \\ \bar{P} = \frac{a+b}{2} \end{cases} \tag{3.40}$$

Worst-case scenario 2: $z(t) = 1$

With the same approach, the problem is also solved when $z(t) = 1$ and $\hat{P}_{Grid}(t) < \bar{P} + z(t)R$, $\forall t \in T$. In this case, three additional solutions are obtained under the aforementioned boundaries (the intersection point is the same for both situations). As in general $\pi^P \approx \pi^R$, Case 4 (Eq. 3.40) always provides the best solution, for both conditions of the absolute value. Nevertheless, this problem and solution are still valid for the case where the $\pi^P \not\approx \pi^R$. However, under these circumstances, all 7 solutions should be computed for all the possible number of active servers, as Case 4 is not guaranteed to be the best.

4) Jointly Selecting $\bar{P}$, R and N_s

$\bar{P}$ and R are functions of N_s (i.e., the number of active servers determined in the VM allocation phase).

To select N_s, I first compute $P^{ch}_{EES}(t)$, $P^{dch}_{EES}(t)$, and the minimum renewable energy ($\min_t(\hat{P}_{PV}(t))$) during the next time slot. For $z(t) = -1$ (Eq. 3.35), $\frac{\hat{P}^{agr}_{Grid}}{T}$ is considered as the maximum predicted data center power (i.e., $\max_t(\hat{P}_{DCT}(t))$) during the time slot. Similarly, when $z(t) = 1$, $\min_t(\hat{P}_{DCT}(t))$ is considered as the minimum predicted power, which is dynamically computed based on the different number of active servers and dynamic power. This assumption provides two benefits:

- Following the RS signal with better reliability and lower error due to the capability of masking the prediction error on the VMs workloads and renewable energy.
- Obtaining less QoS degradation using battery and renewable energy sources for the following regulation signal.

As the data center provider may lose its contract if the tracking error exceeds a limit, the seven solutions should be computed in the worst cases, assuming ϵ zero, to avoid maximizing R by increasing the error (see Eq. 3.28). By doing so, the tracking error can be controlled even when the prediction error on the VMs workloads and renewable energy is high (i.e., during abrupt changes). As stated before, since it is assumed $\pi^P \approx \pi^R$, Case 4 is always the best solution, and it only needs to iterate on the number of active servers, from 1 to N^{DCT}_s. Therefore, for each number of servers, the estimated power taken from the grid ($\hat{P}_{Grid}(t)$) is computed for different a and b values, together with the associated cost. Finally, the $(\bar{P}, R)$ and N_s values are chosen that minimize cost while satisfying the constraints, as shown in Algorithm 6. It is concluded that the proposed method has a time complexity of $O(N^{DCT}_s)$.

Algorithm 6 Bidding - Find $(\bar{P}, R)$ and N_s

Input: $P_{EES}^{ch}(t)$, $P_{EES}^{dch}(t)$, $\min_t(\hat{P}_{PV}(t))$, π^R, and π^R
Output: $(\bar{P}, R)$ and N_s

$\epsilon \leftarrow 0$ (Find best solution without error)
$Cost_{min}^{Tot} \leftarrow$ Maximum real value
for $i = 1 : N_s^{DCT}$ **do**
  $\hat{P}_{Grid}(t) \leftarrow$ Compute $\max_t(\hat{P}_{DCT}(t))$ and $\min_t(\hat{P}_{DCT}(t))$ for two situations separately
  Compute upper (a) and lower (b) bounds
  $Cost_{min} \leftarrow$ Minimum cost among the solutions of two situations
  (i.e., Case 4, Eq. 3.40 when $\pi^P \approx \pi^R$)
  if $Cost_{min} < Cost_{min}^{Tot}$ **then**
    $Cost_{min}^{Tot} \leftarrow Cost_{min}$
    Update $(\bar{P}, R)$
    $N_s \leftarrow i$
  end if
end for

3.6.3.2 Workload Allocation

After finding the number of turned-on servers (i.e., N_s), the same server cap ($\hat{C}_{server}$) is defined for all active servers, as follows:

$$\hat{C}_{server} = max_t(\sum_{k=1}^{N_{VM}} VMcpu_{k,t}^T / N_s) \tag{3.41}$$

where N_{VM} is the total number of VMs in the data center. Matrix $VMcpu_{k,t}^T$ contains the predicted k^{th} VM's CPU utilization at time t during the T^{th} time slot. Setting the same cap for all active servers allows a better control on the QoS. This is because server overutilization can only occur due to under-predictions on the VM usage (i.e., when the VMs require more CPU resources than predicted).

To allocate VMs to servers, a state-of-the-art correlation-aware VM allocation method is used [20]. Correlation refers to the similarity of VMs CPU utilization traces and the coincidence of their peaks. In this algorithm, the VMs are allocated to servers such that the correlation among the allocated VMs in the corresponding server is minimized, while the server does not exceed its defined cap ($\hat{C}_{server}$). This favors consolidation and leads to allocating VMs to the determined number of active servers (N_s). This correlation-aware VM allocation algorithm is periodically invoked at every time slot T.

3.6.3.3 Revising Average Power ($\bar{P}$) and Reserves (R)

The previous step tries to map the VMs (both newly arrived and already running VMs on the system) to servers considering the servers cap. However, due to the allocation error (i.e.,

potentially allocating VMs to a higher number of servers than N_s), it is needed to revise $(\bar{P}, R)$. Therefore, after computing the initial allocation and determining N_s but before sending the reserve value to ISO, the power consumption of data center is estimated using Eq. 3.19 based on the predicted VMs CPU and memory loads. Then, $\hat{P}_{Grid}(t)$ is calculated using Eq. 3.20 with respect to current battery status and predicted renewable power available in time slot T. Finally, $(\bar{P}, R)$ is once updated using the solution with the minimum cost that satisfies the constraints.

3.6.3.4 Online Regulation Signal (RS) Tracking

In this section, I describe my methodology for dynamically modulating the data center, battery and renewable power consumption to track the ISO RS signal (i.e., $P_{Grid}(t) = \bar{P} + z(t)R$) for a given $(\bar{P}, R)$ with the smallest error possible. The proposed method receives three inputs: i) $(\bar{P}, R)$ and $z(t)$ signals, ii) the real VM's CPU utilization and memory footprint, and iii) the current available renewable power. The output of the method is the CPU resource limit per VM, charge/discharge of the battery, and the renewable power usage, required to meet the QoS requirements.

Before formulating the online policy, the following additional assumptions are considered on the EES system:

- A limit is set on the discharge current of the battery, forcing it to be less than the maximum value (i.e., I_{dch}^{max}). It is also assumed that battery can be used to power the data center.

- The EES can be charged (P_{EES}^{ch}) using the power grid when the data center power consumption is less than the power provided by the power grid (i.e., P_{Grid}). In this case, SoC^{lim} is defined as the maximum battery level that can be reached by charging it using renewable power. From SoC^{lim} to full capacity, the battery is charged using the grid power.

- To charge the battery using the grid, a charge current limit (I_{ch}^{lim}) is defined as follows:

$$\begin{cases} I_{ch}^{lim} = I_{ch}^{max} & P_{EES}^{ch} \geq I_{ch}^{max} \cdot V_{EES} \\ I_{ch}^{lim} = \frac{P_{EES}^{ch}}{V_{EES}} & otherwise \end{cases} \tag{3.42}$$

  where V_{EES} indicates the battery voltage level. This guarantees that P_{EES}^{ch} is used to react to the ISO 4-second RS signal fluctuations.

In the method, as shown in Algorithm 7, I first compute the real data center power consumption ($P_{DCT}(t)$) using the real VMs CPU utilization and memory footprint. If $P_{DCT}(t) > P_{Grid}(t)$ (lines 4−17), the current renewable and battery power usage are optimized to compensate

Algorithm 7 Online RS Tracking Policy (every 4 seconds)

Input: $z(t)$, SoC^{lim}, and I_{ch}^{lim}
Output: Following $P_{Grid}(t)$ with minimum error and computing final power taken from the grid

1: Update VMs CPU utilization and memory footprint
2: Update servers utilization (U_{cpu} & U_{mem})
3: Update data center power consumption ($P_{DCT}(t)$)
4: **if** $P_{DCT}(t) > P_{Grid}(t)$ **then**
5: $[P_{PV}(t),\ P_{EES}(t),\ P_{rem}] \leftarrow$ GreenController(**S1**, $P_{DCT}(t)$, $P_{Grid}(t)$, SoC^{lim}, I_{ch}^{lim})
6: $\alpha \leftarrow \pm 1$
7: **if** $P_{rem} > 0$ **then**
8: $QoS_{deg} \leftarrow P_{DCT_{dyn}} / (P_{DCT_{dyn}} - P_{rem})$
9: **if** $QoS_{deg} < QoS_{lim}$ **then**
10: $C_j^{update} \leftarrow (1/QoS_{deg}) \cdot U_{cpu_j} \ \ \forall j \in 1...N_s$
11: **else**
12: $C_j^{update} \leftarrow (1/QoS_{lim}) \cdot U_{cpu_j} \ \ \forall j \in 1...N_s$
13: **end if**
14: Update P_{rem}
15: $P_{Grid}(t) \leftarrow P_{Grid}(t) + P_{rem}$
16: **end if**
17: **else if** $P_{DCT}(t) < P_{Grid}(t)$ **then**
18: $[P_{PV}(t),\ P_{EES}(t),\ P_{rem}] \leftarrow$ GreenController(**S2**, $P_{DCT}(t)$, $P_{Grid}(t)$, SoC^{lim}, I_{ch}^{lim})
19: $\alpha \leftarrow -1$
20: **if** $P_{rem} > 0$ **then**
21: **for** $j = 1 : N_s$ **do**
22: $P_{dyn_j}^{rem} \leftarrow (100 - U_{cpu_j}) \cdot P_{cpu}^{dyn}$
23: **if** $P_{rem} < P_{dyn_j}^{rem}$ **then**
24: $C_j^{update} \leftarrow C_j + (P_{rem} \cdot 100) / P_{cpu}^{dyn}$
25: Update P_{rem}
26: $break$
27: **else**
28: $C_j^{update} \leftarrow 100$
29: $P_{rem} \leftarrow P_{rem} - P_{dyn_j}^{rem}$
30: **end if**
31: **end for**
32: $P_{Grid}(t) \leftarrow P_{Grid}(t) - P_{rem}$
33: **end if**
34: **end if**

the excess power consumed by the data center (i.e., $P_{DCT}(t) - P_{Grid}(t)$) using the GreenController function (line 5). After using green energy, if the data center power is not completely provisioned (i.e., $P_{rem} > 0$), the VMs CPU resources are equally reduced on the servers to meet the available power with minimum tracking error, allowing QoS degradation (QoS_{deg}) until reaching its limit (lines 7–16). For the virtualized applications, the QoS constraints are defined

in terms of the maximum allowable degradation (i.e., increase in their execution time), which in the case is defined as 2x (QoS_{lim}) [161], with respect to their baseline execution time. When the QoS constraint is tight and the VMs resources cannot be further reduced, the battery can additionally be discharged (even going below the DoD) to provision the extra power needed by the data center. Finally, $P_{Grid}(t)$ is updated based on the remaining energy (line 16).

If $P_{DCT}(t) < P_{Grid}(t)$, as shown in lines 17−34, I attempt to charge the battery with the excess power (i.e., $P_{Grid}(t) - P_{DCT}(t)$) provided by the power grid using the GreenController function (line 18). The priority is to fill the battery when the VMs meet the QoS limit. After doing so, if excess power still exists ($P_{rem} > 0$), the resource limit of the VMs is equally increased on the servers, one by one, until meeting the ISO power constraint with the minimum tracking error (lines 20−33), or until no more power can be used or stored.

1) Green Controller - A Constrained Multi-Variable Optimization

To optimize the usage of renewable sources and battery power (charge/discharge), different linear and nonlinear constraints and objective functions are specified for different situations (i.e., **S1** and **S2** as shown in Algorithm 7). For all situations, vector x is optimized, which consists of SoC, I_b, I_b^{CTI}, I_{Rem}^{CTI}, I_{PV}^{CTI}, I_{DCT}^{CTI}, I_{Grid}^{CTI}, and I_{PV}^{waste}, respectively, for the defined system model in Sections 3.6.1 and 3.6.2. lb and ub indicate a set of lower and upper bounds on the design variables in x (i.e., $lb \leq x \leq up$).

S1-Discharge: this state occurs when $P_{PV}^{CTI}(t) \cdot \eta_{DCAC}\left(\rho(t)\right) \leq P_{DCT}(t) - P_{Grid}(t)$ and battery needs to be discharged ($\alpha \leftarrow +1$). The lb and ub are defined on x as follows:

$$\begin{aligned} lb &= [DoD,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0] \\ ub &= [1,\ I_{dch}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ 0] \end{aligned} \tag{3.43}$$

where I_{CTI}^{max} and I_{dch}^{max} denotes the maximum allowable current of the CTI and maximum discharge current of the battery, respectively. The linear equalities on x are indicated as $A_{eq} \cdot x = b_{eq}$, to solve the power management model at the CTI level. I simply name converters efficiency as η since this value is the same for all converters.

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & -1 & 1 & 0 \\ 0 & -V_{EES} \cdot \eta & V_{CTI} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & V_{CTI} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & V_{CTI} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & V_{CTI} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ P_{PV}^{CTI} \\ P_{DCT}^{CTI} \\ P_{Grid}^{CTI} \\ 0 \end{bmatrix} \tag{3.44}$$

The strategy aims at minimizing the following function under the aforementioned constraints, in order to maximize the EES and renewable usage:

$$\min \;\; F = x(4)^2 + (ub(2) - x(2))^2 \tag{3.45}$$

S1-Charge: takes place when $P_{PV}^{CTI}(t) \cdot \eta_{DCAC}\left(\rho(t)\right) > P_{DCT}(t) - P_{Grid}(t)$ and battery can be charged ($\alpha \leftarrow -1$) using renewable energy. Therefore, lb and ub are as follows:

$$\begin{aligned} lb &= [SoC^{cur},\ -I_{ch}^{max},\ -I_{CTI}^{max},\ 0,\ 0,\ 0,\ 0,\ 0] \\ ub &= [SoC^{lim},\ 0,\ 0,\ 0,\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max}] \end{aligned} \tag{3.46}$$

where SoC^{cur} indicates the current SoC of the EES. The linear equalities on x are represented as (changes with respect to S1-Discharge are bolded):

$$\begin{bmatrix} 0 & 0 & 1 & \mathbf{0} & 1 & -1 & 1 & -\mathbf{1} \\ 0 & -\mathbf{V_{EES}} & \mathbf{V_{CTI}} \cdot \eta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & V_{CTI} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & V_{CTI} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{0} \\ P_{DCT}^{CTI} \\ P_{Grid}^{CTI} \\ 0 \end{bmatrix} \tag{3.47}$$

Furthermore, the following linear inequality stands when renewable power cannot be fully used:

$$V_{CTI} \cdot I_{PV}^{CTI} \leq P_{PV}^{CTI} \tag{3.48}$$

The following function is also minimized under the aforementioned constraints:

$$\min \;\; F = -x(5)^2 + (lb(2) - x(2))^2 \tag{3.49}$$

S2-Charge: takes place when $P_{Grid}(t) - P_{DCT}(t) > 0$ and the generated grid power is higher than data center power consumption. Therefore, battery can be charged using the power grid ($\alpha \leftarrow -1$) and lb and ub are defined as follows:

$$\begin{aligned} lb &= [SoC^{cur},\ -I_{ch}^{lim},\ -I_{CTI}^{max},\ 0,\ 0,\ 0,\ 0,\ 0] \\ ub &= [1,\ 0,\ 0,\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max},\ I_{CTI}^{max}] \end{aligned} \tag{3.50}$$

the linear equalities on x are represented as (changes with respect to S1-Discharge are bolded):

$$\begin{bmatrix} 0 & 0 & 1 & \mathbf{-1} & \mathbf{0} & -1 & 1 & 0 \\ 0 & \mathbf{-V_{EES}} & \mathbf{V_{CTI}} \cdot \eta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & \mathbf{V_{CTI}} \\ 0 & 0 & 0 & 0 & 0 & V_{CTI} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{-1} \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ P_{PV}^{CTI} \\ P_{DCT}^{CTI} \\ \mathbf{0} \\ 0 \end{bmatrix} \tag{3.51}$$

The linear inequality on x is also shown when grid power is not completely used to supply the data center and charge the battery:

$$V_{CTI} \cdot I_{Grid}^{CTI} \leq P_{Grid} \cdot \eta \tag{3.52}$$

The minimization function is as follows:

$$\min \quad F = -x(7)^2 + (lb(2) - x(2))^2 \tag{3.53}$$

Finally, a non-linear equality ($C_{eq}(x) = 0$) is defined according to Eq. 3.3 and 3.4 used for all the situations to compute the SoC considering the available charge capacity impacted by aging as:

$$\begin{aligned} C_{eq} &= [x(1) - \frac{C_{ref} \cdot SoC - I_{eq} \cdot dt}{C_{ref}}] \\ where \quad I_{eq} &= \left(\frac{|x(2)|}{I_{ref}}\right)^{(k_b - 1)} \cdot x(2) \end{aligned} \tag{3.54}$$

To solve the optimization problem, I utilize the fmincon function [162], which is part of a non-linear programming solver, to find the minimum of the constrained non-linear multi-variable problem. Differently form the green controllers presented in Sections 3.4.1.2 and 3.5.2.2-3, neither an offline method nor a ruled-based algorithm can be used here. In this problem, the controller should follow the RS signal at run-time under the different situations caused by power market requirements, while optimizing the green energy sources usage.

3.6.4 Experimental Setup and Scenarios

In this section I present the experimental setup and introduce two scenarios to compare the proposed strategy, i.e., ECOGreen.

3.6.4.1 Experimental Setup

1) Green Data Center Configuration

I model 30 racks, each rack with 10 Intel S2600GZ servers equipped with a 6-core CPU (Intel E5-2620) and 32GB of memory (RAM). The server power consumption is modeled as in Section 3.6.2.2. Each server consumes constant 16 W and 27.2 W for disk (P_{disk}) and cooling (P_{fan}) power, respectively. A high fan speed (8000 rpm) and a low inlet temperature ($22°C$) are considered to reduce the effect of temperature-dependent leakage power. Under this condition, leakage power (P_{cpu}^{leak}) is almost constant and 3.1 W in the worst case. Idle power for CPU (P_{cpu}^{idle}) and memory (P_{mem}^{idle}) are 50 W and 4 W, respectively. The CPU and memory dynamic power (P_{cpu}^{dyn} and P_{mem}^{dyn}) range from 0 to 42.5 W and 0 to 56 W, respectively (with their maximum values occurring at 100% utilization) [96].

2) Simulation Framework

In order to consider realistic CPU and memory usage traces, I use one week of traces of Google Cluster [163], which provides the CPU and memory utilization for VMs every 5 minutes (memory utilization is varying in the range of 2% to 32%). Arrival and total time (life time) of each VM, given in time slots, are generated by poisson and exponential distributions, respectively. VM allocation and power market bidding are invoked every hour, and the online RS tracking policy is invoked every 4 seconds. The optimization problem is solved using the following values: $\pi^P = \pi^R = 0.1$ \$/kWh, and c = 1 based on typical values of today's markets [130]. The typical trajectories of $z(t)$ are used from PJM historical data [129], for a time horizon of one week.

The PV module size is tuned based on the peak power production (i.e., the number of cells and panels), defining it to 35 kWp. The irradiance forecasts are computed by implementing the algorithm presented in previous work [9]. Figure 3.19 shows the real versus forecasted PV power traces for one week. Moreover, in the simulations, a lithium-ion EES system is

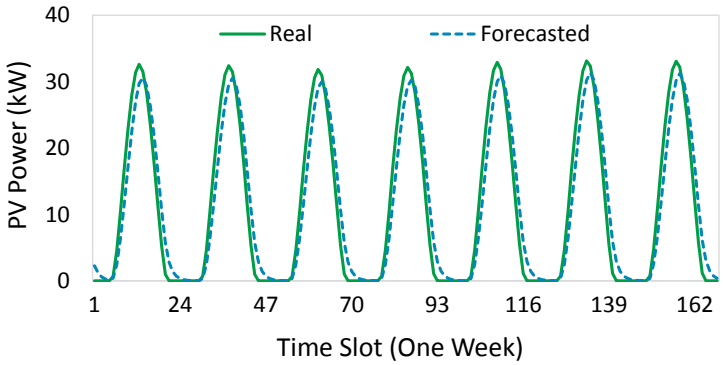

Figure 3.19 – Forecasted versus real PV power profile for a time horizon of one week.

considered with 96 kWh capacity and 70% of DoD, while keeping the remaining capacity in case of outage.

3.6.4.2 Scenarios

1) Scenario I - Bidding and RS Signal Tracking Analysis

In this scenario, I evaluate the impact of using demand-side EES and renewable energy with and without data center participation in the RS program. I also investigate the effectiveness of the proposed strategy compared to the state-of-the-art. All approaches consist of an allocation policy, and a bidding plus online tracking policy, as shown in each method name, as follows:

- PSA+DynPow [70]: this work uses a server Power-State-Aware method (PSA) to estimate the $\bar{P}$ and R, by determining the number of active servers for serving the workloads. Also, a Dynamic Power control policy (DynPow) is proposed to track the RS signal using power capping and power level adjustment using different server power modes. I adapt this method to the proposed work by considering three power modes: turned-off, idle, and active.
- PSA+DynPow w/o Bid: PSA+DynPow without bidding and RS tracking. In this case, it is assumed $R = 0$ and $\bar{P}$ as the average data center power consumption (i.e., $\bar{P} = \frac{\int_t^{t+T} P_{DCT}(t)dt}{T}$) taken from the grid during a time slot, as determined by the allocation policy.
- COAT+DynPow: for a further comparison with the aforementioned work [70], I consider the bidding and tracking solution methods proposed in this work [70] jointly with the COnsolidation-Aware allocaTion method (COAT) [20], as it is one of the best energy-aware VM consolidation strategies in the state-of-the-art.
- COAT+DynPow w/o Bid: COAT+DynPow without bidding and RS tracking, to evaluate the impact of allocation.
- ECOGreen: the proposed Electricity Cost Optimization for Green data centers.
- ECOGreen w/o Bid: ECOGreen without bidding and RS tracking, optimizing green power usage to minimize the cost and reduce grid power (i.e., $\bar{P} = \frac{\int_t^{t+T} P_{Grid}(t)dt}{T}$).

2) Scenario II - Impact of Workload Allocation Methods

To evaluate and isolate the impact of VM allocation from the effect of bidding on online tracking policies, I also compare ECOGreen against different VM allocation policies, namely:

- COAT: COnsolidation-Aware allocaTion [20].

- LB: Load Balancing strategy that aims to spread VMs across servers, reaching an average server utilization close to 50%.
- ECOGreen: the proposed optimization strategy.
- ECOGreen w/o Green: ECOGreen without renewable and battery sources (green energy).

All the above-mentioned workload allocation methods are used in conjunction with the proposed bidding, online RS signal tracking, and green (renewable and EES) controller.

Finally, in order to evaluate the efficiency of the proposed strategy under potential trade-offs between different objectives, I compute the euclidean distance for each method i in a normalized multi-dimensional space from the optimal values per each dimension (objective), i.e., vector $\mathbb{O}$. $\|\mathbf{1}\|_2$ shows the maximum distance from the normalized optimal values (here is the square root of the number of considered objectives). This efficiency metric is given by Eq. 3.55, and the higher E is, the higher the efficiency of the method:

$$E_i = \|\mathbf{1}\|_2 - \|\mathbb{M}_i - \mathbb{O}\|_2 \tag{3.55}$$

3.6.5 Experimental Results

3.6.5.1 Scenario I - Bidding and RS Signal Tracking Analysis

In this section I compare the proposed strategy, i.e., ECOGreen, in terms of bidding ($\bar{P}$ and R), monetary cost, and QoS against different state-of-the-art approaches, introduced in Section 3.6.4.2-1.

1) Bidding ($\bar{P}, R$) Analysis

The ($\bar{P}, R$) values of different approaches for a time horizon of one week are shown in Fig. 3.20 and 3.21, respectively. Approaches w/o Bid are not shown in Fig. 3.21, as $R = 0$ and they only optimize the power consumption of the data center. Due to the nature of consolidation, which packs VMs into the minimum number of servers, COAT+DynPow provides lower $\bar{P}$, but also lower R, as it has less slack to dynamically change the server resources and meet VMs requirements. On the contrary, PSA+DynPow provisions higher R due to the larger power range achievable. This is achieved at the expense of higher $\bar{P}$, as VMs are distributed among a more servers to avoid further QoS degradation. It is observed that ECOGreen provides RS reserves (R) of 35% and 76% of $\bar{P}$ in the worst case and average, respectively, over one week. In the best case (i.e., when available renewable energy is high and the load of the data center is low) ECOGreen provides 100% of $\bar{P}$ as R, drastically reducing cost when compared to other approaches. In this sense, the best solution to minimize electricity cost is not necessarily trying to achieve the lowest $\bar{P}$ or the highest R, as the best bidding varies depending on the availability of demand-side renewable energy and the EES system status.

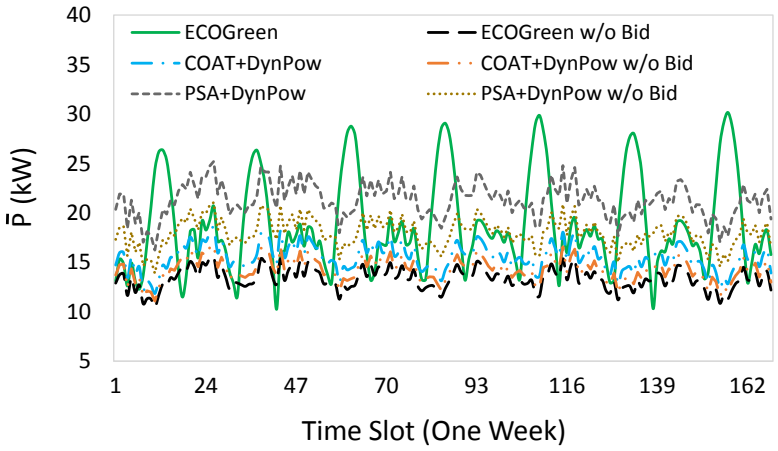

Figure 3.20 – Average power consumption ($\bar{P}$) for a time horizon of one week.

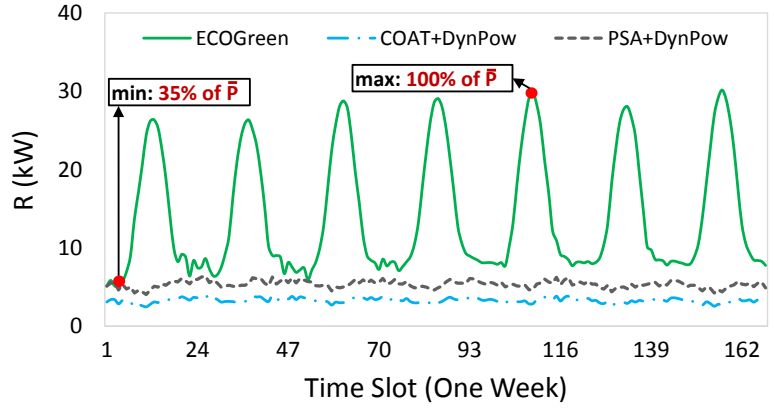

Figure 3.21 – The amount of reserves (R) for a time horizon of one week.

Also, these figures illustrate that approaches without bidding try to minimize total data center power consumption in order to reduce the electricity cost. However, the capacity reserves offered by the power grid bring the opportunity to reduce monetary costs even when increasing the average power consumption, as shown in the next section, due to the high credit obtained for the reserves.

2) Monetary Cost

The monetary cost in the RS provision case is calculated using the objective function defined in Eq. 3.29. In this calculation, installation costs of demand-side EES and renewables equipment are not taken into account for all approaches. This is because even if this installation has non-negligible costs, this research focuses on data center operational expenditure (OPEX) reduction, not on capital expenses. Without RS provision (i.e., w/o Bid), the monetary costs are calculated based solely on the power consumed by the data center during the time slot,

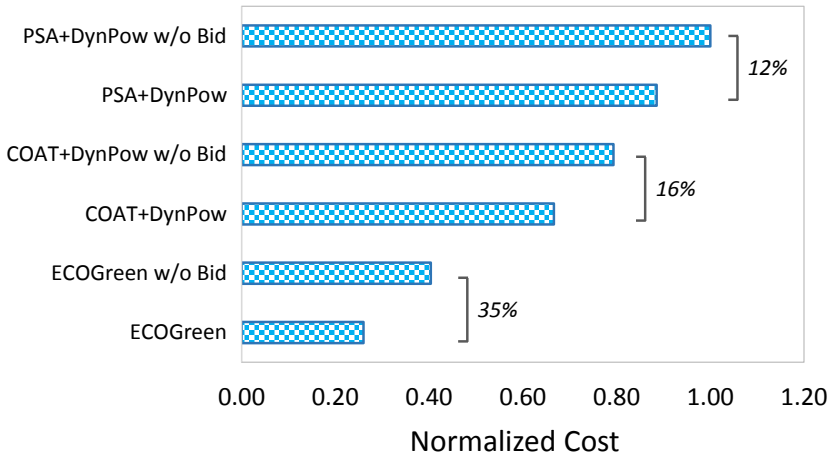

Figure 3.22 – Normalized monetary cost over a time horizon of one week.

which is $\bar{P} \cdot \pi^p \cdot T$, where $\bar{P} = \frac{\int_t^{t+T} P_{DCT}(t)dt}{T}$. To better evaluate the savings, the total monetary cost of each method is normalized to the largest value among all the methods, as shown in Fig. 3.22. All in all, ECOGreen uses the PV and EES power to optimize the bidding values and monetary costs, while ECOGreen w/o Bid only optimizes the green power usage for the data center. In contrast to COAT+DynPow and PSA+DynPow, ECOGreen tries to match $\bar{P}$ and R, enabling the green data center to provide reserves close to its average power consumption. Given that the price for average power is the same than the credit obtained for the reserves (i.e., $\pi^P \approx \pi^R$), having similar values for $\bar{P}$ and R yields the lowest cost, reduces the tracking error and efficiently utilizes renewable and EES power.

In addition, the results of ECOGreen w/o Bid show how renewables and battery power sources can save monetary costs compared to COAT+DynPow and PSA+DynPow, even without bidding. Also, COAT+DynPow and PSA+DynPow provide better results than COAT+DynPow w/o Bid and PSA+DynPow w/o Bid but with higher QoS degradation in order to regulate the data center power consumption. As a consequence, ECOGreen reduces the power costs by 35%, 61%, and 71% in comparison to ECOGreen w/o Bid, COAT+DynPow, and PSA+DynPow, respectively.

For more details, Tables A.1-A.4, Fig. A.1 and A.2 in Appendix A shows how the ECOGreen reduces cost using the different energy sources for two corner points.

3) Quality of Service (QoS)

For virtualized applications, the QoS constraint is defined in terms of the maximum allowable degradation (i.e., increase in execution time). This limit is specified as 2x (QoS_{lim}) [161], with respect to the VMs' baseline requirements. Fig. 3.23 shows the average degradation among all degraded VMs for a time horizon of one week. Degradation occurs due to miss-predictions on VMs workloads (especially during abrupt workload changes) and renewable energy production. Degradation increases when all power generation sources are not able to

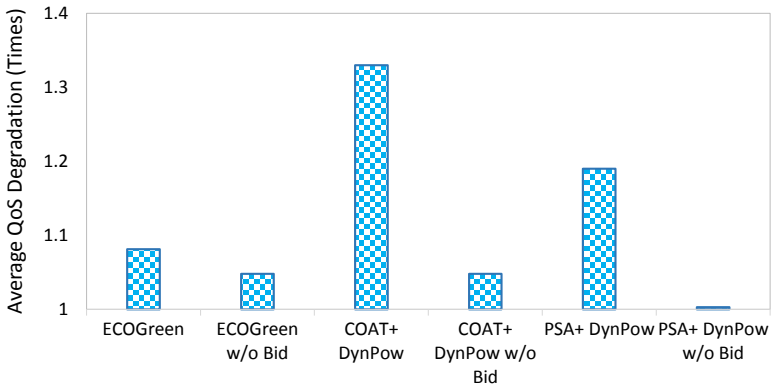

Figure 3.23 – Average QoS degradation.

provide the required power consumption of the data center.

Overall, the experimental results indicate that all the approaches meet the QoS limit in the worst case.In fact, COAT+DynPow has less control on servers overutilization during peak loads, and the behaviour worsens due to the need for tracking the RS signal. To reduce QoS degradation, PSA+DynPow uses idle servers as QoS guarantee slack for immediately serving coming workloads. Finally, ECOGreen reduces the average degradation compensating the tracking error using renewable and EES. ECOGreen w/o Bid and COAT+DynPow w/o Bid reach similar result due to using the same policy for allocating VMs to the servers. In conclusion, the proposed strategy is able to meet the QoS degradation constraint for virtualized applications in RS reserves provision market, and obtains 19% and 10% less degradation on average than COAT+DynPow and PSA+DynPow, respectively.

3.6.5.2 Scenario II - Impact of Workload Allocation Methods

In this section I compare the proposed holistic strategy in terms of monetary cost, total green data center power consumption breakdown, EES efficiency, and potential trade-offs against the different state-of-the-art VM allocation policies introduced in Section 3.6.4.2-2.

1) Monetary Cost

Figure 3.24 shows the normalized monetary costs of the green data center for a time horizon of one week. Due to the nature of consolidation, COAT reduces the average power ($\bar{P}$) by co-allocating VMs into servers until reaching the maximum server capacity. This leads to reducing the RS reserves due to the decreased flexibility in increasing the CPU resources of each VM, as well as efficiently using the renewable energy. On the contrary, by distributing the VMs to a larger number of servers, LB provides higher reserves at the expense of an increase in the average power. Finally, ECOGreen jointly optimizes the bidding power market values and the number of active servers while considering the current state of EES and the predicted

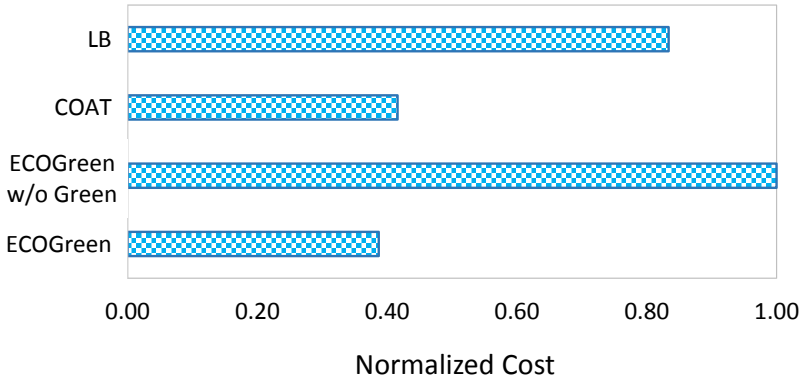

Figure 3.24 – Normalized monetary cost over a time horizon of one week.

renewable power. Therefore, results show that the lowest average power or highest reserves do not provide, by themselves only, lower monetary cost in today's RS reserves provision. On the other hand, LB can still provide better savings compared to the proposed strategy when not using demand-side EES and renewable sources (i.e., LB versus ECOGreen w/o Green) due to the optimization of green power usage.

In summary, ECOGreen obtains 7%, 53%, and 61% monetary savings compared to COAT, LB, and ECOGreen w/o Green, respectively.

2) Power Consumption Analysis

Figure 3.25 shows the total power consumption breakdown of the green data center for a time horizon of one week. As a consequence of maximizing renewable and battery power utilization, all approaches show an overall green power usage improvement higher than ECOGreen w/o Green. ECOGreen uses more grid power than COAT to provide better bidding values (i.e. higher

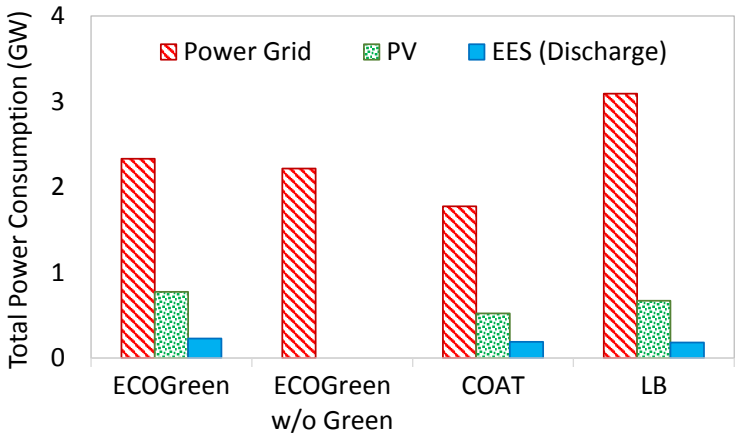

Figure 3.25 – The total power consumption breakdown of the green data center for different power supply sources for a time horizon of one week.

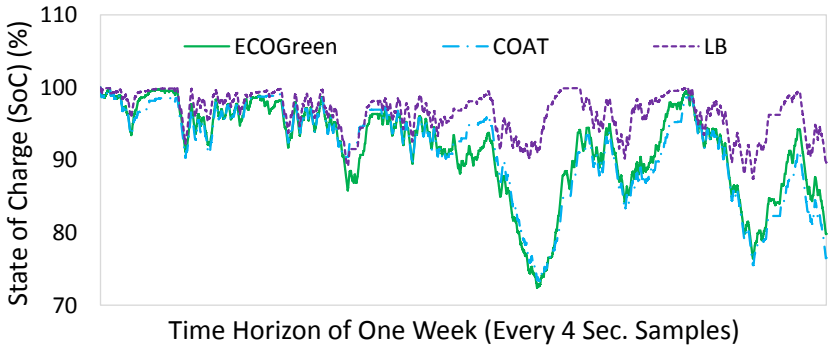

Figure 3.26 – State-of-Charge (SoC) of battery bank for a time horizon of one week.

RS reserves). ECOGreen also maximizes green energy usage, achieving 48% and 22% renewable and battery utilization improvement, respectively, compared to COAT. This situation is due to the larger reserves achieved by ECOGreen, which is a consequence of exploiting the green energy sources to track the RS signal. Furthermore, ECOGreen outperforms LB in terms of power consumption as in average it allocates VMs to a fewer number of servers. In addition, because of its lower tracking error, the battery is not significantly discharged. ECOGreen achieves up to 16% and 28% PV and battery power utilization improvements, respectively, compared to LB.

3) EES Performance Analysis

Figure 3.26 shows a one-week view of the system evolution with battery charge/discharge cycles. Some constraints on the allowed DoD are added to the battery bank. In particular, to force the battery bank to work in the optimal range of SoC, the minimum SoC is set to 70%. In this figure, it is observed how the energy buffer in RS reserves allows to use battery power to optimize the bidding values and follow the RS signal in different situations. ECOGreen better utilizes the battery power than the other approaches, especially when compensating the excess power needed by the data center. However, this leads to decreasing the SoH of the battery, in particular during discharge cycles. As ECOGreen w/o Green does not have an EES system, this method is not shown in Fig. 3.26.

4) Performance Metrics Trade-offs Analysis

The experimental results confirm that, having a holistic strategy, better overall results can be obtained by exploiting renewable and battery sources. Figures 3.27, 3.28 and 3.29 summarize the benefits of ECOGreen in comparison with other state-of-the-art techniques. First, Fig. 3.27 depicts the cost versus green power sources trade-off, showing the best performance for ECOGreen in terms of both monetary cost and green power usage. Table 3.4 summarizes the efficiency of ECOGreen compared to other methods in terms of power usage of different sources in RS reserves provision.

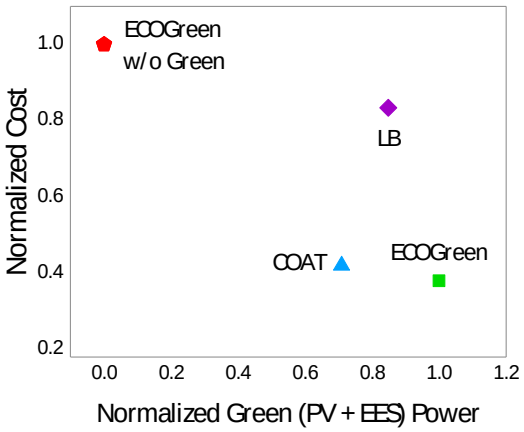

Figure 3.27 – Cost versus green power trade-off.

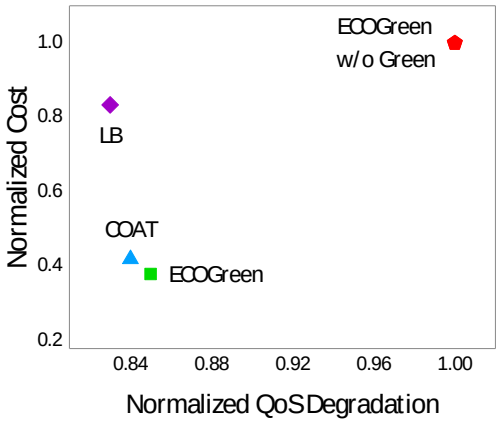

Figure 3.28 – Cost versus QoS degradation trade-off.

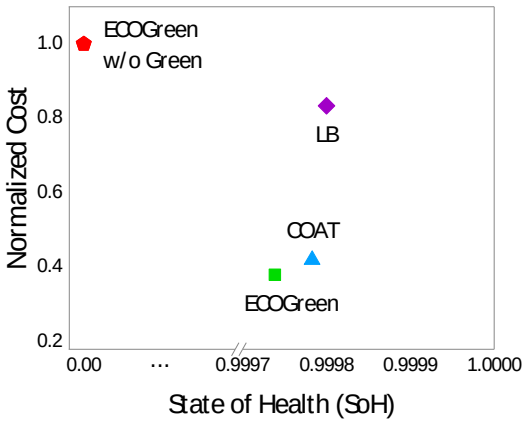

Figure 3.29 – Cost versus State-of-Health (SoH) trade-off.

Figure 3.28 shows the normalized cost versus QoS degradation trade-off, with ECOGreen providing 7%, 53%, and 61% improvements for cost compared to COAT, LB, and ECOGreen w/o Green, respectively. The results imply that all the methods can meet the QoS limit, with ECOGreen w/o Green exhibiting the highest degradation (1.3x on average). This is because ECOGreen w/o Green cannot tolerate power reduction without QoS degradation to track the

Table 3.4 – Cost, grid, PV and EES (battery) usage improvements for the proposed strategy, i.e., ECOGreen, compared to other approaches

	COAT	LB	ECOGreen w/o Green
Cost	7%	53%	61%
Grid Power	-29%	25%	-5%
PV Power	48%	16%	w/o PV
Battery Power	22%	28%	w/o EES

Table 3.5 – The overall efficiency of different methods according to Eq. 3.55

	ECOGreen	COAT	LB	ECOGreen w/o Green
Efficiency (E)	1.23	1.09	0.92	0

RS signal; while ECOGreen can use the renewable and EES to supply the additional power consumed by the data center. Compared to COAT and LB, it achieves less than 2% QoS degradation.

Finally, Fig. 3.29 depicts the cost versus SoH trade-off: the proposed algorithm results in better monetary cost at the expense of a slightly higher (0.03%) battery SoH decrease (ratio between nominal and remaining capacity, as computed in Eq. 3.1). If it is considered that batteries reach their end-of-life when working at 70% of their nominal capacity, ECOGreen enables a battery lifetime above 15 years.

5) Discussion - Efficiency of The Proposed Strategy

As this work deals with a multi-objective problem, an efficiency metric (Eq. 3.55) has been defined to evaluate ECOGreen in a holistic way. The objectives considered (i.e., vector $\mathbb{M}$ in the metric) are monetary cost, renewable usage, EES utilization, QoS and battery lifetime.

Table 3.5 shows the efficiency with respect to the worst-case distance value. Based on the equation, optimal solution value is $\|\mathbf{1}\|_2$ (here is $\sqrt{5}$, 5 is the number of objectives), while the ECOGreen w/o Green provides the minimum efficiency due to the lack of green energy sources. As a result, in the considered green data center scenario, ECOGreen achieves the best overall performance (highest efficiency value) compared to other approaches in today's emerging power markets.

3.7 Summary

Ever increasing demands for computing and growing number of clusters and servers in data centers have ramped up power consumption world-wide, which is estimated to be at 1.3% of the global usage, and growing at a yearly rate of 20% [164]. Consequently, with the increase in power usage, the electricity cost of data centers doubles every five years [61]. However, optimizing the energy and cost of a single data center powered by the grid is not enough in today's cloud computing context, where multiple data centers, built in different geographical locations, are used to deploy online services, and use renewable energy sources to reduce their carbon footprint and operational cost.

In this chapter, I have firstly presented a novel dynamic and multi-objective framework to optimize the trade-offs between energy consumption of data center, battery banks lifetime and

energy bill cost for green data centers. The Datacenter Energy Controller minimizes the total energy consumption using the state-of-the-art correlation-aware VM allocation scheme for the given VMs specifications and energy budget provided by the Green Energy Controller while improving QoS requirements. In the Green Energy Controller, I use a real-time optimization technique to maximize the lifetime of battery banks and to reduce the energy bill by managing the PV source, in price-varying scenarios, and considering the energy consumed by the data center. I also validated the effectiveness and applicability of the proposed system with the utilization traces obtained from a real data center setup. The experimental results show that the proposed framework provides up to 11.6% energy savings and up to 10.4% improvement of QoS level compared to existing conventional solutions under different number of VMs in the system, and up to 96% money saving in the electricity bill.

Secondly, I proposed a novel method to tackle the challenges of operational cost optimization and energy-performance trade-off on resource-constrained green geo-distributed data centers. I introduced a two-phase multi-objective VM placement algorithm along with a dynamic migration technique that exploit the holistic knowledge of VMs characteristics. The first phase, i.e. global controller, clusters VMs for each data center considering time-varying VMs CPU-load and data correlations and the status of data center energy sources. The second phase, i.e. local controller, allocates the VMs of each data center cluster to servers exploiting CPU-load correlation. The experimental results showed that, using the proposed method, up to 54%, 14%, and 10% improvements can be obtained for operational cost, energy consumption and performance, respectively, compared to state-of-the-art approaches.

Finally, I have introduced ECOGreen, a novel strategy to tackle the challenge of allowing green data centers to participate in RS reserves provision. I have first presented a mathematical solution to jointly find the best average power and reserve values in the bidding problem, as well as the number of active servers needed in the VM allocation phase. I have also optimized the EES and renewable power usage in a resource-constrained green data center. Then, I have proposed a runtime approach that dynamically regulates the data center power consumption following the RS signal, while also guaranteeing the QoS limit. The runtime policy utilizes VMs recourse limit control (i.e., dynamically changing the server resources allocated to VMs), EES and renewable power in decision making to minimize the signal tracking error. As a result, I have compared the holistic strategy, i.e., ECOGreen, in terms of monetary cost, total power consumption breakdown of green data center, QoS degradation, and EES efficiency analysis against different state-of-the-art approaches. The experimental results have shown that ECOGreen obtains up to 71%, 48%, and 28% monetary cost, renewable and EES utilization improvements, respectively, at the expense of battery SoH decreasing when compared to other approaches. Nonetheless, the battery aging shows a lifetime longer than 15 years. Overall, ECOGreen ensures achieving the best trade-off between different objectives when compared to all other methods.

4 Towards Next-Generation Near-Threshold Computing Data Centers

4.1 Introduction

The backbone of today's Information Technology (IT) is large-scale data centers that host a myriad of IT services, such as search and social connectivity, and operate under strict sub-second Quality-of-Service (QoS) requirements. State-of-the-art data centers deployed by IT giants, such a Microsoft and Google, host several thousands of servers, have huge acquisition costs ($100+ million), and have vast power footprints (5-20 MW) [164]. Furthermore, in a typical scale-out data center, from a software architecture perspective, requests are independently distributed across servers that do not share any state, and the performance characteristics of each server dictates the data center's overall performance [165]. In order to maximize the computational power of each server, processor and system vendors have turned to customized server architectures for data centers, identifying and eliminating the bottlenecks in conventional server processors executing workloads.

From the hardware perspective, based on Moore's law, more transistors could be integrated on a chip. However, the end of Dennard scaling unveils an era of power limited chips such that technology scaling has started to lag behind. As we penetrate into the deep sub-micron era, successive technology generations have dramatically increased the chip's power density due to the stagnation of supply voltages. This phenomenon, i.e., end of Dennard scaling, as shown in Fig. 4.1, results in an underutilization of the available transistors on the chip, a trend which is expected to continue and escalate in the future. Low-voltage operation is a well-known technique to improve energy efficiency of digital computing devices, due to lowering the supply voltage that leads to the dynamic power reduction [166]. In traditional fields of ultra-low-power applications, Near-Threshold Computing (NTC) has been demonstrated to provide up to an order of magnitude of improvement in energy efficiency [72]. NTC takes advantage of the quadratic dependence between dynamic power consumption and supply voltage, by lowering the operating voltage to a value slightly higher than the transistor threshold, increasing energy efficiency at the expense of decreased performance.

For current cloud applications, NTC allows to optimize the trade-off between performance

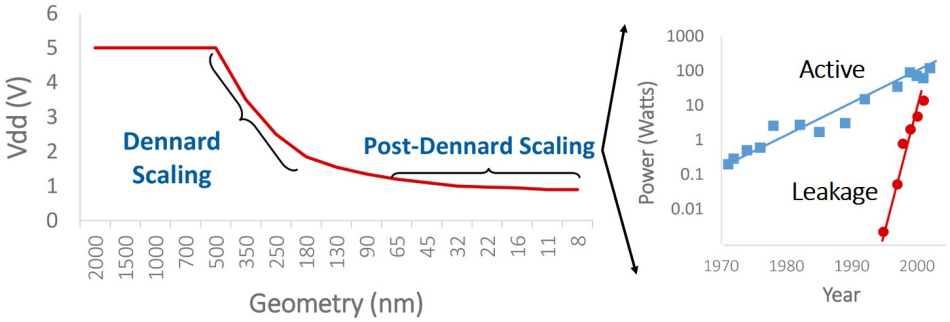

Figure 4.1 – The effect of post-Dennard scaling [167].

and power, bringing server energy-proportionality, and emerging as a promising approach to overcome the power-wall. From a technology viewpoint, Fully Depleted Silicon On Insulator (FD-SOI) is suitable for NTC due to providing a significantly increased voltage range and even higher performance for the same energy thanks to the better behavior of transistors at low voltage [73]. Interestingly, none of the prior work has considered near-threshold operation for servers, due to the strict QoS requirements of cloud computing workloads. However, this work shows the energy efficiency of NTC servers improved by reducing frequency while meeting the strict QoS requirements of data center workloads.

4.1.1 Contributions

In this chapter, I first show how efficiency, i.e., performance per power ratio, can be achieved by using the FD-SOI technology when running traditional virtualized applications on two different server architectures: i) scale-out processors, and ii) ARM-based server. For this purposed, appropriate voltage and frequency are set to these near-threshold servers. Then, at the data center level, I evaluate the impact of these new architectures and technologies on traditional energy-aware VM consolidation techniques deployed in conventional servers. Virtual Machine (VM) consolidation [14] has represented for years the most widely used technique to minimize energy consumption. However, the emergence of energy-proportional NTC servers brought by the FD-SOI technology, with drastically reduced static power, together with the advent of applications able to work at reduced frequencies, changes the underlying assumptions that made consolidation best for energy efficiency. Finally, the results conclude how consolidation and server turn-off policies no longer yield the best energy efficiency in these scenarios, and how the proposed policy can provide significant energy savings.

4.2 State-of-the-art on Technology, Architecture, and System-Level Energy Efficiency Management

In this section, I classify previous studies on the area of energy efficiency in different levels, namely, technology, server architecture and data center workload allocation methods.

4.2.1 Technology and Architecture

With further technology scaling, the capability of arranging hundreds of homogeneous, heterogeneous, or hybrid cores into one chip (i.e., chip multiprocessors) has been increased [168]. However, ever increasing the number of cores leads to rapid increase of the on-chip power density. Hence, according to the dark silicon phenomenon, a part of chip (i.e., 50% of chip size) must be powered off due to the power budget limitation and thermal management, in particular at lower technology (e.g., 8nm technology) [169, 170]. Many works have addressed the dark silicon phenomenon challenge [171, 172, 173, 174]. For instance, Swaminathan *et al.* [172] introduced the concept of dim silicon to power on a larger fraction of the chip by reducing the supply voltage near to the threshold voltage. Lyons *et al.* [173] and Cong *et al.* [174] used energy-efficient hardware accelerators, instead of power-hungry processors, to mitigate the dark silicon phenomenon.

In the recent past, the dedicated Application-Specific Integrated Circuit (ASIC) design has been extensively used in different areas of low-power systems. While all these systems obtain remarkable energy efficiency, their performance is not scalable as they are designed for a specific scenario [175]. Recent work in the area of energy-efficient server design focuses on presently-shipping enterprise servers, with traditional x86 architectures [161]. These servers had traditionally been designed to meet performance goals, without energy efficiency as a design constraint. Only recently, with the stagnation of Dennard scaling [71], and the resulting power-limited servers, NTC turned into a key technology to improve energy efficiency. With respect to traditional bulk technology, FD-SOI enables the NTC giving an increased voltage range and even higher performance for the same energy thanks to the better behavior of transistors at low voltage [176].

Previous work on near-threshold many-cores mainly focused on single voltage domain and multiple frequency domain architectures [177]. However, other recent works on processors in FD-SOI demonstrated the near-threshold capabilities of the technology, capable to run a dual-core Cortex A9 processor at 1GHz at the supply voltage of 0.6V [73]. Moreover, in FD-SOI, back-bias voltage can be varied from -3V Reverse Body Biasing (RBB) using conventional-well transistors up to +3V Forward Body Biasing (FBB) using flip-well transistors [73]. Applying such a strong bias has a significant impact on the leakage-performance trade-off as the threshold voltage of transistors varies by 85mV when the bias voltage value is changed by 1V.

In order to increase the performance of application, from the architecture viewpoint, one way is to use Non-Uniform Cache Architectures (NUCA) to reduce access time to the Last-Level Cache (LLC) [178]. In order to overcome the interconnect delays, richly-connected topology is one of the promising solutions [179]. However, this solution imposes significant area and energy overheads on many-core chip architectures [180]. Hence, it is essential to design a chip with a simple crossbar interconnect and a small-sized LLC, mitigating the inefficiency of NUCA.

Prior works tried to optimize the chip multiprocessors design either focused on providing

the optimal core micro-architecture for a specific application [181], or on finding the optimal cache architecture for a given number of cores [182, 183]. Most of these works assume non-data center applications. Hence, Lotfi-Kamran *et al.* [165] presented scalable and efficient scale-out server processors designed for data center workloads. With the consideration of performance-density, they proposed optimal multi-core configurations, called PODs (clusters). Each POD behaves as an complete server, coupling a number of cores to a small LLC with a fast interconnect. This design avoids the inter-POD interconnect cost and delay. Moreover, replicating the POD to fill the die area results in a maximum per-chip throughput (performance density), lower design complexity, and technology scalability.

This work is the first one proposing the usage of NTC servers in Ultra-Thin Body and Buried Oxide (UTBB) FD-SOI technology for virtualized applications based on the modified scale-out processors. To assess the energy efficiency, a detailed power characterization of the core and uncore components has been considered based on the FD-SOI technology. Moreover, the target Cavium ThunderX servers [184] (a commercial ARM-based server in the market, implemented in 28nm HKMG technology) have been validated for virtualized applications based on 28nm FD-SOI technology process.

4.2.2 Energy-aware VM Allocation

Today's data centers benefit from virtualization technology [185], hosting multiple VMs on one server to achieve higher server utilization and felexibility [186]. Research in the area of energy efficiency in cloud computing usually focuses on consolidation-based VM allocation techniques to decrease power while meeting a certain QoS [187]. The VM allocation problem have been solved by different bin-packing algorithms to minimize the total number of turned-on servers, preventing the servers from being underutilized or overutilized [81]. Hadji *et al.* [188] addressed the VM allocation problem in a dynamic workloads scenario. This problem also formulated for a multi-objective optimization, aiming at maximizing application performance, and minimizing data center power consumption and operational cost [186, 189].

For allocating VMs to physical servers, several works only check that the total size of VMs' load does not exceed the maximum server's capacity [14, 79], or their peak, off-peak, and average utilization of VMs [12, 83]. However, the dynamic nature of cloud workloads results in the CPU-load correlation across VMs (i.e., the similarity of CPU utilization traces and the coincidence of their peaks). In this context, a few studies [20, 22, 28] consider CPU-load correlation to achieve further energy savings, keeping the total co-located VMs demand near to the maximum server capacity during a long time. Ruan *et al.* [87] propose a dynamic migration-based VM allocation method to achieve the optimal balance between server utilization and energy consumption. Garg *et al.* [190] tackle the allocation problem for different types of applications to maximize the resource utilization and profit. Nevertheless, having considered the traditional x86 server architectures, these approaches assume a linear power-frequency relation for a given workload, and large static server power. However, this is not compatible with novel server architectures,

and in particular at data center level. Lin *et al.* [191] addressed the VM allocation problem for server chips based on the 7nm Fin Field-Effect Transistor (FinFET) technology to reduce the memory coherence and inter-core communication overhead and improve parallelism. To optimized the energy consumption, they maintained the total workload of a server around a certain utilization level (a fixed server utilization cap, i.e., 70%, to host the VMs), as the most energy-efficient state of the server. However, in the dynamic scenario when the VMs requirements change, the optimal utilization level of the server significantly varies due to the effect of non-energy proportionality of the sever components on power consumption.

The exploration of the new trade-offs and impact on energy-aware VM allocation, brought by new server architectures (in particular NTC servers), remains today an open issue. In this chapter, I propose a novel dynamic VM allocation method that exploits the knowledge of VMs characteristics and uses the power model based on the FD-SOI technology and server architecture information to increase the energy proportionality of next-generation NTC-based data centers, while guaranteeing the QoS. The results demonstrate the inefficiency of the latest workload consolidation techniques for new NTC-based data center designs.

4.3 Overview of The System

4.3.1 Process Technology

This work exploits the capabilities of UTBB FD-SOI technology at low voltage in near-threshold servers for running cloud computing applications. In addition, body biasing can give an extra advantage, as shown in Fig. 4.2. RBB provides minimum leakage power for idle or sleep mode; while, FBB brings higher performance for operating mode, but at the expense of higher leakage power. In the context of cloud applications, the described capabilities of UTBB FD-SOI technology can be exploited to:

- Operate at the best energy efficiency point for a given performance target. By exploiting FBB, it is possible to reduce the supply voltage of a device to achieve the best energy point, at the cost of increased leakage, improving energy efficiency in dynamic-power dominated operating regions.
- Manage spikes of computation. FBB allows to temporarily boost the operating frequency of processors. With respect to voltage scaling, FBB speed-ups transitions between the normal and boost modes. For example, the back-bias voltage of a $5mm^2$ Cortex A9 processor can switch between 0V and 1.3V in less than $1\mu s$ [73].
- Achieve state-retentive leakage management. With RBB, processor can temporarily enter low-leakage sleep mode, reducing leakage power by up to an order of magnitude [73]. Since the contents of the processor register files and buffers are not lost and restored on wakeup, this is referred to as state-retentive power gating of register files. Compared to traditional power gating techniques, body biasing allows faster transitions between

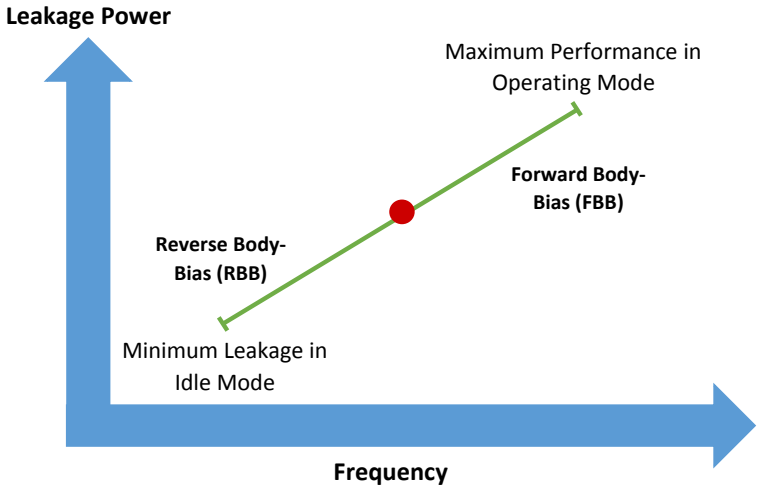

Figure 4.2 – Body-biasing power/performance trade-off [192].

the two modes (i.e., from sleep to operating mode), and is intrinsically state-retentive. This helps to avoid losing the contents of the processor register files and buffers, and restoring data on wakeup.

- Manage variations. In different environmental conditions, the control of variations is essential to maintain the nominal level of performance and energy efficiency. Typically, the temperature variations are larger than process variations. Thanks to the extended body bias range of FD-SOI technology, part of the body bias range can be used to mitigate the effect of process variations, while a wider range should be used to compensate temperature variations that are magnified in near-threshold operation [175].

The 28nm FD-SOI technology process is currently employed for mass production by Samsung and ST Microelectronics; the 20nm technology is being produced by GlobalFoundries while the 12nm node is on the strategic roadmap [74]. In the context of this work, a flip-well (LVT) implementation of 28nm UTBB FD-SOI technology is considered, able to provide higher frequencies than the conventional-well flavor, and featuring FBB in the range 0V-3V, suitable for high-performance applications.

4.3.2 Server and Data Center Architecture

In this section, I first introduce two considered NTC server architectures: i) the scale-out architecture optimized for scale-out data center workloads and used for overall evaluation of the benefits of NTC servers, and ii) a commercial ARM-based architecture tuned to virtualized applications (the data center configuration is based on this server architecture).

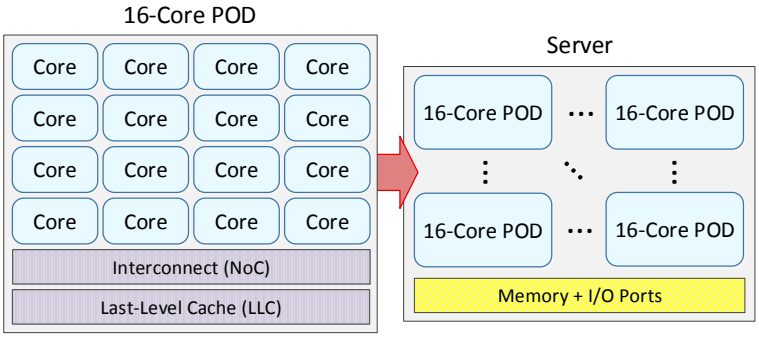

Figure 4.3 – Server architecture with 16-Core Clusters (PODs).

4.3.2.1 Server Architecture Based On Scale-Out Server Platform

The scale-out server has been designed and optimized for data center workloads. In this architecture, a number of cores is coupled to a small LLC with a fast interconnect. For this platform, chips with an area of $300mm^2$, and a power budget of 100W are considered [165]. In the chip, the cores are considered as Cortex A57, a 3-way Out-of-Order (OoO) core, resembling those used in specialized many-cores for servers [184, 193]. Following the scale-out processor methodology [165], the chip is organized as a set of clusters (PODs), which exhibit an optimal ratio between core count and cache size. Each POD behaves as an complete server, and it is replicated to fill the die area. Although the optimal ratio is calculated as a 16-core cluster (POD) with a 4MB LLC (Fig. 4.3), 4-core clusters are modeled due to a lower simulation turnaround time, verified that the cluster's core count does not affect the trends of results. Each cluster features a cache-coherent crossbar interconnect and runs its own OS image. Besides the cores, caches, and interconnects, the chip features a set of IO peripherals along the chip's edge, which are modeled using McPAT [194] following a Sun UltraSparc T2 configuration.

The server comprises four DDR4 memory channels clocked at 1600MHz with a peak bandwidth of 25.6GB/s per channel. Four ranks per channel and 8x 4Gbit DRAM chips are modeled following Micron's specifications [195]. As a result, the server's total memory capacity is 64GB.

4.3.2.2 Server and Data Center Architecture Based On Modified Commercial Cavium ThunderX Platform

In this section servers are modeled with multi-core processors, DDR4 memory, memory controllers, IO, peripherals, interconnect, and motherboard, as shown in Fig. 4.4a. As a starting point for the server architecture, the Cavium ThunderX platform is chosen [184]. However, for the target applications, the Cavium performance was slower (from 1.5x to 3.5x) than the x86 platform with similar characteristics, and unable to meet QoS constraints, as shown in Table 4.2 in Section 4.6.3. This was due to an inappropriate memory subsystem design for the target applications considered and the choice of in-order cores. Hence, the original architecture has been modified and ARMv8 Cortex A57 OoO cores have been used,

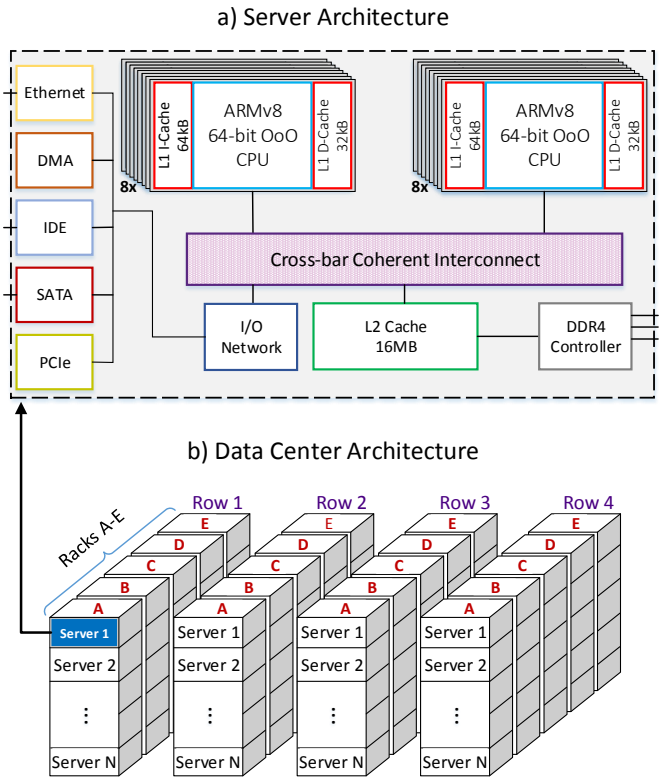

Figure 4.4 – a) Server and b) data center architecture.

instead of the in-order Cortex A53 processor.

A 16-core CPU (instead of 48 in ThunderX) is modeled to achieve a lower simulation turnaround time, as is experimentally observed that the model linearly scaled up for larger servers. The memory subsystem was also updated, by including L1 instruction cache (I-cache) and data cache (D-cache) of 64KB and 32KB, respectively. A LLC of 16MB is modeled. A total memory size of 16GB is considered using a DDR4 memory model with memory controller [195]. DDR4 is clocked at 2400MHz with a peak bandwidth of 19.2GB/s.

A data center typically comprises a hierarchical structure from top to down: rows, racks, and servers, as shown in Fig. 4.4b. Each row contains several racks and each rack encompasses several servers. Without loss of generality, for this exploration, a data center with 600 NTC servers (60 racks and 10 servers per rack) is simulated, each NTC server with its dedicated power supply, fans and disks.

4.3.3 Application Description

The considered applications consist of VMs, virtualized via Linux LXC containers, and running synthetically generated workloads that resemble batches of real banking applications, as

reported by our industry partners. For realistic CPU and memory usage patterns, I use one week of traces of Google Cluster [163], which provide CPU and memory utilization for over 600 VMs, reported every 5 minutes. Memory utilization is varying in the range of 2% to 32%, while CPU utilization range is between 2% and 100% with a higher variance than memory footprints. Therefore, for profiling purposes, the workloads are split in three categories, according to the per-VM memory utilization: i) low-mem for average memory usage of 70MB (7%), ii) mid-mem for 255MB (25%) and iii) high-mem for 435MB (43%). Moreover, in order to run the experiments in worst-case scenarios, the workloads are tuned to maximum CPU utilization.

4.3.4 QoS Degradation Constraint for VMs

Because banking applications are virtualized batch jobs, their QoS constraints are defined in terms of the maximum allowable degradation (i.e., increase in their execution time). According to our industrial partners, the minimum degradation observed in their production data centers is 2x, while the maximum degradation can reach values as high as 4x [161].

To evaluate the impact of the Voltage/frequency (V/f) points on the QoS for NTC-based scale-out servers, the performance degradation of the workloads is computed taking as a baseline the 2GHz frequency (maximum frequency). For modified Cavium ThunderX platform, the performance degradation of the workloads is computed with respect to a baseline execution in a 16-core Intel Xeon X5650 running at 2.6GHz, with 12MB LLC and 128GB of RAM clocked at 1333MHz, in which one LXC container (VM) is run per core.

4.4 Server and Data Center Power Models

The overall NTC server power model has been extracted by combining direct measurements on a commercial ARM-v8 based server [196] with power measurements of real prototypes implemented in 28nm FD-SOI technology and operating in near-threshold regime [73, 175], allowing for a very accurate system power estimation for all the operating conditions investigated in this work. Four main contributors to the overall power consumption of the server are considered: i) the core region composed of the A57 cores logic and the L1/L2 caches, ii) the LLC, iii) the memory controller, peripherals, IO subsystem and motherboard, and iv) the DRAM banks.

4.4.1 Cores

In this research, the 28nm FD-SOI power and performance model of a recent Cortex A9 implementation of STM in 28nm bulk and FD-SOI are combined, considering the differences in pipeline length ratio and critical path between Cortex A57 and Cortex A9. These parameters are extracted by comparing the different voltage to frequency ratio (extracted via the CPUFreq Linux driver) present in the Samsung Exynos processor family. The Cortex A57 is 1.17x faster

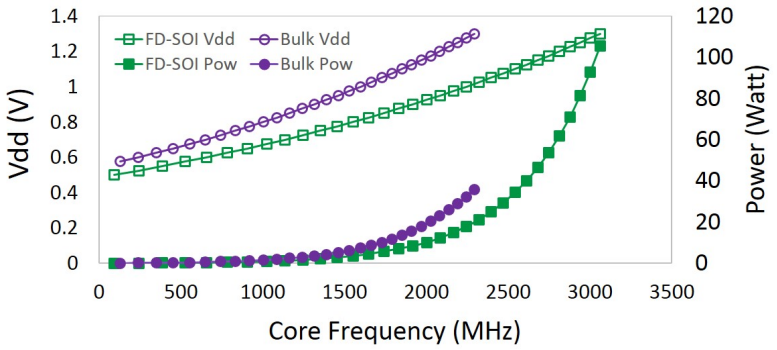

Figure 4.5 – A57 performance and power model in bulk and FD-SOI technology.

(i.e., it reaches a higher frequency) than the Cortex A9. This information is combined with the active and static energy per clock cycle at the different Dynamic Voltage and Frequency Scaling (DVFS) levels from the Samsung Exynos 5433 processors to scale its energy figures to the STM 28nm FD-SOI technology by using the trends reported in prior work [73]. These numbers also account for the L1 and L2 cache power consumption.

When in active state the non-filled marked lines of Fig. 4.5 show the voltage level required at each frequency for each technology. It is observed that the use of the FD-SOI technology increases the frequency with respect to pure bulk. While pure bulk A57 has timing issues when operating in the low voltage region (0.5V), the FD-SOI implementation reaches almost 100MHz. In this figure, the filled marked lines show that FD-SOI by itself leads to a significant reduction in the power consumption at the same frequency with respect to bulk silicon technology. This power gain increases as the voltage supply reduces, producing the maximum benefits in the near-threshold operating region.

When in Wait-For-Memory (WFM) state the core region consumes 24% less power than when active. This number has been measured empirically on an Intel Xeon v3 processor. Then, the performance and power model is extended to the NTC region fitting a template extracted from measurements of a 28nm UTBB FD-SOI near-threshold parallel processor [175].

4.4.2 Last-Level-Cache (LLC)

The LLC power model was extracted by measuring the leakage power for a 256KB SRAM block in 28nm UTBB FD-SOI (ranging from 32 to 444 mW/16MB) and read and write energy [pJ/Access] for 128-bit wide accesses (ranging from 0.8 to 5.6 pJ/byte). All these values have been obtained for different voltage levels.

4.4.3 Memory Controller, Peripherals, IO and Motherboard

The memory controller, peripherals and IO subsystem power consumption overhead of an Intel Xeon v3 CPU has empirically been measured. This power consumption is split in two parts:

i) a constant component which accounts for the static and fix dynamic power cost needed to keep these subsystems on, and ii) a component proportional to the operating condition. The constant part causes a 11.84W overhead in all operating points, while the proportional one ranges from 1.6W to 9W in the operational range. Finally, the same motherboard power consumption is assumed than in the Cavium ThunderX server, which is of 15W for a low fan speed, and with 1 SSD disk.

4.4.4 DRAM

The DRAM power has been modeled with direct measurement on a real server platform based on Intel Xeon v3 architecture. During a large variety of workloads, the total DRAM read and write accesses have been measured in windows of 1 second and, accordingly measuring the power of the DRAM banks. Afterwards, the empirical measurement has been interpolated with a linear power model. The final model contains the empirical measurement of an idle power value of 15.5mW/GB per GB of DRAM, which increases to 155mW/GB when the banks are activated. On top of this static power, an energy consumption of 800pJ/Byte is reported per byte read.

4.4.5 Overall Data Center Power

All these power consumption values have been inserted in the GEM5 simulator to estimate the power consumption of each server node under real workload. Total data center power consumption (P_{DC}) ia taken into account as the sum of power consumed by servers (P_s).

4.5 Assessment of Energy-Performance Trade-offs in The NTC-Based Scale-Out Server Architecture

Following the architecture organization described in Section 4.3.2.1, the processor die can accommodate 9 clusters before hitting the area limit. Each cluster contains 4 Cortex A57, 3-way OoO, with an instruction window of 128 instructions. Each core integrates a 32KB 2-way L1-Instruction (L1-I) and L1-Data (L1-D) cache. Each cluster hosts a unified 4MB 16-way LLC with 4 banks. The cores and the LLC banks are interconnected through a crossbar. The chip features a total of 36 cores and uses the 28nm FD-SOI process technology.

This section explores the trade-offs in energy and performance when running cloud applications (i.e., virtualized banking applications). This study demonstrates the benefits of near-threshold operation and proposes several directions to synergistically increase the energy proportionality of a near-threshold server.

4.5.1 Setup

For the experiments, the *Flexus* full-system cycle-accurate simulation infrastructure [197] is used. Flexus models the SPARCv9 Instruction Set Architecture (ISA) and runs an unmodified Solaris 10 operating system. Flexus extends the *Simics* functional simulator with timing models of OoO cores, caches, on-chip protocol controllers, interconnects, and DRAM. DRAM is modeled by integrating DRAMSim2 [198] directly into Flexus. DRAMSim2 is configured following Micron's DDR4 specifications [195].

To enable virtualization in Solaris 10, I employ Solaris containers (a.k.a., Solaris Zones), which are integrated with the operating system. One Solaris container instance is run on each of the cores of the cluster. Each container runs one instance of a synthetic banking application, that is tuned to obtain various CPU and memory stress levels for the containers.

To accelerate simulations, the statistical sampling methodology of the Flexus simulator is employed [199]. The samples are drawn over an interval of 10 seconds of simulated time. For each measurement, the simulations are launched from checkpoints with warmed caches and branch predictors, and run 100K cycles to achieve a steady state of detailed cycle-accurate simulation prior to collecting measurements for the subsequent 50K cycles. To measure performance, the ratio of the aggregated number of application instructions committed to the total number of cycles (including cycles spent executing operating system code) is used; this metric, User Instructions Per Cycle (UIPC), or User Instructions Per Second (UIPS), has been shown to reflect system throughput [197]. Performance is measured at a 95% confidence level and an average error below 2%.

4.5.2 Quality-of-Service (QoS) and Energy Assessment

4.5.2.1 Quality-of-Service (QoS)

NTC allows to operate in a wide range of V/f points to improve energy efficiency. However, the strict QoS requirements of applications make it unclear whether this technology is suitable for server processors. In order to understand the effects of the V/f points on the QoS for virtualized applications, the performance degradation of the banking workloads (i.e., low-mem and high-mem as defined in Section 4.3.3) is computed taking as a baseline the 2GHz frequency (i.e., the increase in execution time as frequency decreases). As shown in Table 4.1, by assuming the maximum boundary of 4x degradation [161], frequency can be decreased down to 500MHz for both low-mem and high-mem. Even by limiting the maximum degradation to 2x, frequency could still be reduced to 1GHz. Therefore, the frequency of the cores can significantly be reduced while meeting the strict QoS requirements (acceptable degradation) of banking applications.

Table 4.1 – QoS analysis, i.e., Billion UIPS of virtualized applications on ARM-based scale-out processor under different frequency levels

Application	2GHz	1.5GHz	1GHz	500 MHz	200 MHz	100 MHz	**QoS limit (2x)**
low-mem	5.13	3.92	2.63	1.33	0.53	0.27	**2.56**
high-mem	9.17	7.05	4.72	2.36	0.94	0.47	**4.58**

4.5.2.2 Energy Efficiency

1) Cores

To understand the efficiency benefits of reducing the V/f points, Fig. 4.6 indicates the total number of UIPS at the chip level divided by the total power consumption of the A57 cores. As expected, due to the cubic relation between frequency and power, and the linear relation between throughput and frequency, the lower the frequency, the higher the energy efficiency. However, there is a voltage point (i.e., 0.5V), where cores become non-functional due to the L1 cache, before entering a low-frequency region where leakage brings efficiency down. In conclusion, the most energy-efficient design is the one that operates at the lowest V/f point. Hence, maximum energy-efficiency at low power operating point has the advantage of reducing the overall system Thermal Design Power (TDP)—easing the thermal design [200], and dark-silicon effects.

In the context of virtualized applications, the UIPS of high-mem is higher than low-mem, as apart from increasing memory usage high-mem also increase CPU boundness when compared to low-mem. Depending on the limit imposed to degradation (i.e., 2x or 4x), the best frequency ranges from 500MHz to 1GHz.

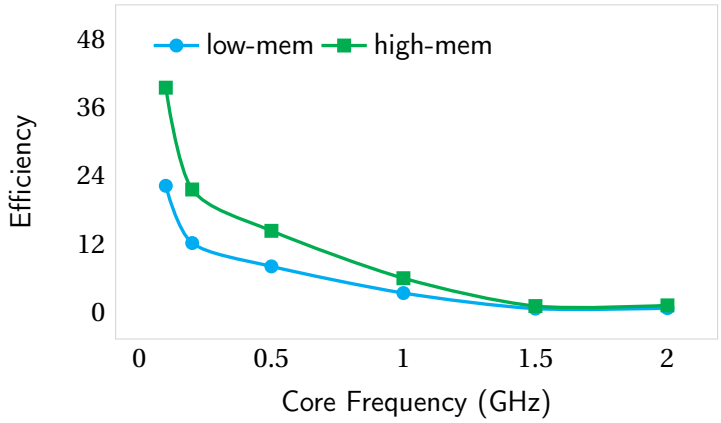

Figure 4.6 – Efficiency of the cores calculated as UIPS/Watt as the core frequency varies for the virtualized applications.

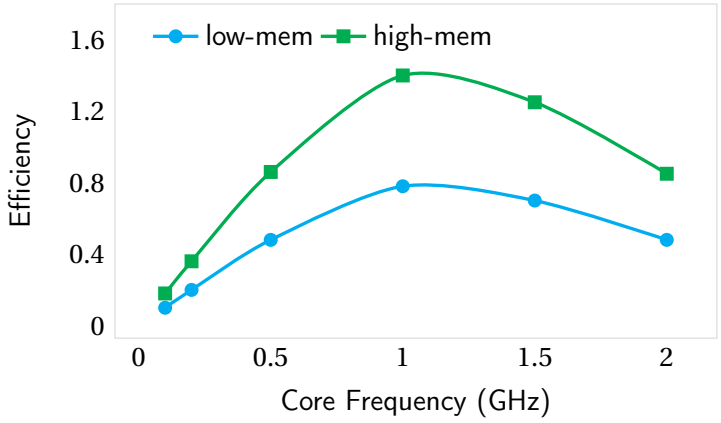

Figure 4.7 – Efficiency of the System on Chip (SoC) calculated as UIPS/Watt as the core frequency varies for the virtualized applications.

2) System on Chip (SoC)

Previously, core power has been considered to compute efficiency. However, there are other components in the processor die, i.e., System on Chip (SoC), that dissipate power. Each cluster has a LLC and a crossbar interconnect that operate at a different V/f point than that of the cores. Additionally, there is a set of IO peripherals along the chip's edge that consume power regardless of the state of the cores. Figure 4.7, considers UIPS at the chip level, divided by the total power consumption of the SoC, for VMs. As can be seen, the most energy-efficient point is not the lowest core frequency. The reason is that there is a point at which the reduction in throughput is not compensated by the power reduction in the cores, as the power of the remaining chip components dominate. This constant power at the chip level pushes the most energy-efficient point to 1GHz of core frequency.

3) Server

Besides the processor, the memory subsystem is one of the most important contributors to overall server power [201]. Although the dynamic power consumption of the memory scales with the frequency of the cores, the background power consumption (static power) remains constant in particular when the cores issue fewer references per unit of time. Figure 4.8, considers UIPS at the chip level, divided by the total power consumption of the server, which considers the power of the SoC and the memory subsystem for the VMs. As expected, the optimal efficiency point moves to around 1.2GHz. As a result, the optimal efficiency point moves further to the right as other system components, which are not energy proportional, are taken into account.

Overall, the aforementioned results shows significant potential for the NTC process technology for servers. However, in order to achieve significant improvements in energy efficiency, not only for the cores, but also for the rest of the components of the processor, all remaining server

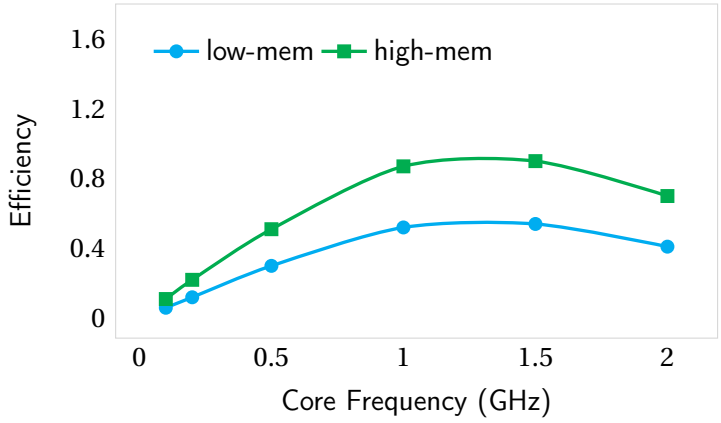

Figure 4.8 – Efficiency of the server calculated as UIPS/Watt as the core frequency varies for the virtualized applications.

components of the system (e.g., motherboard, memory controller, IO peripherals, network, disk, etc.) need to be energy proportional [202].

4.5.2.3 Discussion

The results shown in this section unveil some interesting trade-offs to increase the energy efficiency of servers. First, server workloads tolerate low core frequency, enabling near-threshold operating voltage. However, not all SoC components scale with the core voltage, shifting the most energy-efficient point to a higher frequency, i.e., 1GHz. Hence, there are interesting challenges and opportunities to increase the energy-efficiency at the SoC level, by making the uncore components and DRAM more energy proportional, instead of optimizing the design for the TDP. More specifically, when operating at near-threshold operation, the server is still energy-bound instead of power/thermal bound. While power/thermal bound is fundamental and can be addressed with cutting-edge cooling technology [203, 204], energy optimizations can be achieved at the circuit, architecture, and control level. In this perspective, FD-SOI provides effective knobs to improve energy proportionality to reduce leakage. Additionally, this technology is applicable not only for the cores, but also for the uncore components. Moreover, the background power of the memory dominates the total server power as the power consumption of the SoC decreases. Therefore, memory technologies that exhibit lower background power than DDR4, such as mobile DRAM (LPDDR4), could be used to increase the energy proportionality of the servers [205].

Finally, given that the core frequency can be greatly reduced, application consolidation should be possible in these scenarios at server level. Specially, under the more relaxed latency constraints of the public cloud environments, where servers are usually oversubscribed, the optimal energy efficiency point could be adjusted to accommodate more workloads on the same server.

4.6 Proposed Optimization Method for NTC-Based Data Centers

4.6.1 Data Center Scenario and Motivational Example

As a motivational example, Fig. 4.9a shows the worst-case data center power consumption in an NTC-based data center when servers run at different frequencies (the same frequency for all the servers) for various data center CPU utilization rates. The CPU utilization rate is defined as the ratio of required CPU resources in MHz to the total CPU resources in the data center (i.e., the number of servers multiplied by the maximum CPU resources of one server). In the following scenario and setup, I first shows the effect of NTC-server frequency levels on the data center power consumption, when running a CPU-bounded workload (i.e., dynamic memory power is close to zero). Then, I investigate the impact of memory system on power consumption and energy-efficient strategy.

In the setup, 80 servers are considered with a maximum frequency (F_{max}) of 3.1GHz. As CPU utilization rate increases, it is needed to either turn on more servers, or set higher frequencies to the turned-on servers. A traditional consolidation approach minimizes the amount of active servers and runs them at the highest frequency possible. However, in NTC-based data centers, the optimal frequency of servers (F_{opt}^{NTC}) is around 1.9GHz, instead of 3.1GHz, in terms of CPU power consumption due to the non-linear behavior of CPU power with voltage and frequency. For a utilization rate higher than 50%, the optimal frequency is the minimum possible that meets the workload demand. On the contrary, Fig. 4.9b shows the power consumption of a non-NTC-based data center (equipped with 6-core Intel E5-2620 servers), where consolidation is the most energy-efficient strategy due to the high static power of the servers.

On the other hand, in the power model (Section 4.4.4), memory power consumption is a linear function of the number of memory accesses per second. Thus, from the memory power perspective, to minimize energy consumption VMs should be consolidated as memory capacity allows, and keep the number of active servers to a minimum. Hence, in NTC-based data centers, CPU and memory bounded workloads exhibit opposite behaviour in terms of efficiency. Therefore, neither VM consolidation nor load balancing are the best options, as the optimal server frequency and workload allocation strategy dynamically change depending on the data center workload.

4.6.2 EPACT: Proposed Energy Proportionality-Aware DynamiC AllocaTion Method

Given the previous analysis, I propose the Energy Proportionality-Aware dynamiC allocaTion (EPACT) method to allocate the total number of VMs available in the data center (N_{VM}) to servers every time slot T, while trying to make servers work at the most energy-efficient frequency (F_{opt}^{T}) in each sampled value 1..n (one sample every 5 minutes) during time slot T (considered as 1 hour).

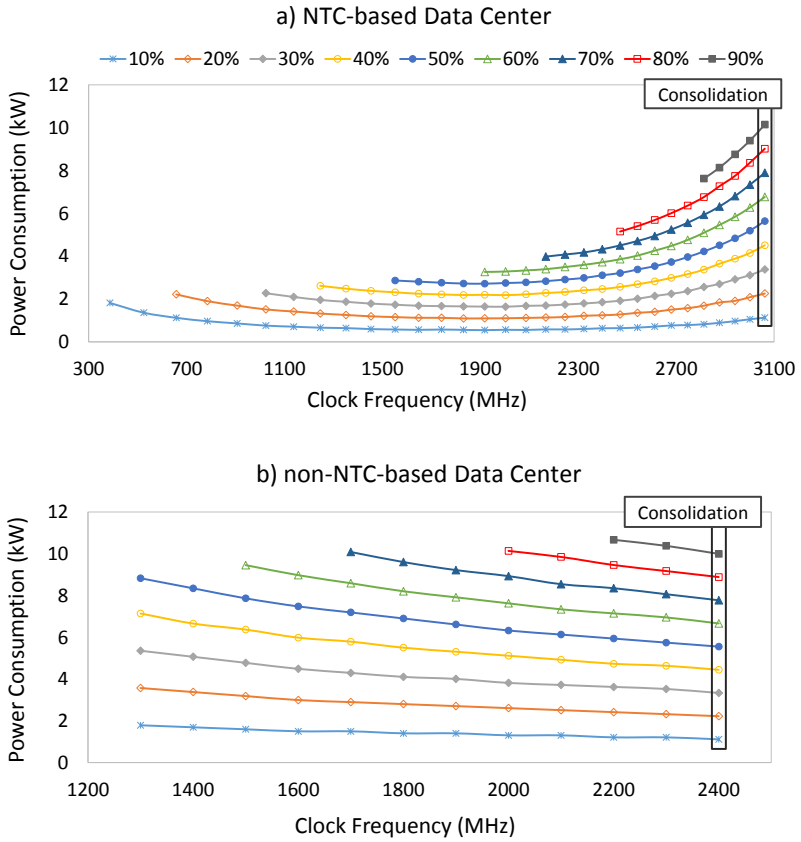

Figure 4.9 – Power consumption under different data center utilization for CPU-bounded tasks (no dynamic memory power) for a) NTC-based and b) non-NTC-based data center.

The proposed method requires predicting, at the beginning of T, the per-VM CPU and memory utilization patterns ($\tilde{U}_{cpu}$ and $\tilde{U}_{mem}$). Given the daily periodicity observed in the VMs of Google Cluster traces, the Autoregressive Integrated Moving Average (ARIMA) prediction model is used [159]. ARIMA considers the CPU and memory utilization from the previous week and forecasts the next-day traces per VM. The worst-case prediction error is around 10%.

Given these predictions, the optimal number of turned-on servers is first determined from the CPU and memory perspective, independently:

$$\hat{N}^{cpu}_{server} = \frac{max_n(\sum_{k=1}^{N_{VM}} \tilde{U}^{k,n}_{cpu}) \cdot F_{max}}{F^{NTC}_{opt} \cdot 100}$$
$$\hat{N}^{mem}_{server} = \frac{max_n(\sum_{k=1}^{N_{VM}} \tilde{U}^{k,n}_{mem})}{100} \tag{4.1}$$

From the CPU viewpoint the number of servers is chosen that allows to set a frequency as close to $F^{NTC}_{opt} = 1.9GHz$ as possible, and from the memory standpoint the VMs are consolidated

Algorithm 8 Finding Optimal Server Frequency Level

Input: $\tilde{U}_{cpu}$, $\tilde{U}_{mem}$, $\hat{N}^{cpu}_{server}$, $\hat{N}^{mem}_{server}$, and F_{max}
Output: Finding and updating F_{opt}
1: **if** $\hat{N}^{cpu}_{server} > \hat{N}^{mem}_{server}$ **then**
2: $P^{i}_{DC} \leftarrow MAX\_REAL$
3: **for** $i = \hat{N}^{mem}_{server} : \hat{N}^{cpu}_{server}$ **do**
4: $F^{i}_{opt} \leftarrow \frac{max_n(\sum_{k=1}^{N_{VM}} \tilde{U}^{k,n}_{cpu}) \cdot F_{max}}{i \cdot 100}$
5: **if** $F_{opt} \leq F_{max}$ **then**
6: $P^{i}_{DC} \leftarrow$ Compute total data center power consumption
7: **if** $P^{i}_{DC} \leq P_{DC}$ **then**
8: $F_{opt} \leftarrow F^{i}_{opt}$
9: $P_{DC} \leftarrow P^{i}_{DC}$
10: **end if**
11: **end if**
12: **end for**
13: Launch Algorithm 9
14: **else if** $\hat{N}^{cpu}_{server} \leq \hat{N}^{mem}_{server}$ **then**
15: $F_{opt} \leftarrow \frac{max_n(\sum_{k=1}^{N_{VM}} \tilde{U}^{k,n}_{cpu}) \cdot F_{max}}{\hat{N}^{mem}_{server} \cdot 100}$
16: Launch Algorithm 10
17: **end if**

until the maximum server memory capacity (i.e., memory cap) is hit.

The definition of $\hat{N}^{cpu}_{server}$ and $\hat{N}^{mem}_{server}$ results in two cases, as described in Algorithm 8:
1) If $\hat{N}^{cpu}_{server} > \hat{N}^{mem}_{server}$, all the number of turned-on servers between these two values is exhaustively explored, until finding the F^{T}_{opt} that exhibits the lowest data center power consumption. Then, as described in Algorithm 9, the best VMs fit into servers are found by using the First-Fit-Decreasing algorithm, only taking into account the CPU utilization, as they drive QoS. Thus, one server is selected (ID_s, line 1). If the server is empty, the first unallocated VM is selected from the pool of VMs, allocating it to the corresponding server. The server load patterns (both $Patt_{ID_s,cpu}$ and $Patt_{ID_s,mem}$, lines 4-6) are updated, as shown in Fig. 4.10-step 1. Otherwise, first, the complementary utilization pattern of server ($Patt^{Com}_{s,cpu}$) is computed with respect to its current maximum load (line 8), Fig. 4.10-step 2. Then, one VM is selected from the pool of VMs and allocated, which has the maximum similarity (ϕ, defined as the Pearson Correlation) to the pattern such that the maximum aggregated load of server ($\max([Patt_{ID_s,cpu} + \tilde{U}^{ID_{VM}}_{cpu}]/100).F_{max}$) is less than F^{T}_{opt} (simply named F_{opt}), Fig. 4.10-step 3. Otherwise, another server is turned on (lines 9-17).

2) If $\hat{N}^{cpu}_{server} \leq \hat{N}^{mem}_{server}$, memory dominates and the optimal frequency is defined based on $F_{opt} = \frac{max_n(\sum_{k=1}^{N_{VM}} \tilde{U}^{k,n}_{cpu}) \cdot F_{max}}{\hat{N}^{mem}_{server} \cdot 100}$. In this case, the allocation phase needs to take into account both the CPU utilization and memory footprint patterns to find the best VMs fit into the servers based on CPU cap (i.e, $Cap_{cpu} = (F_{opt} \cdot 100)/F_{max}$) and memory cap (i.e., $Cap_{mem} = 100\%$).

Algorithm 9 The Proposed 1D VM Allocation Algorithm

Input: $\tilde{U}_{cpu}$, $\tilde{U}_{mem}$, F_{opt}, and F_{max}
Output: Allocating VMs to servers

$ID_s \leftarrow 1$
while All VMs not allocated **do**
  if Server ID_s is empty **then**
    $ID_{VM} \leftarrow$ First unallocated VM
    $Patt_{ID_s,cpu} \leftarrow Patt_{ID_s,cpu} + \tilde{U}_{cpu}^{ID_{VM}}$
    $Patt_{ID_s,mem} \leftarrow Patt_{ID_s,mem} + \tilde{U}_{mem}^{ID_{VM}}$
  else if Server ID_s is not empty **then**
    $Patt_{ID_s,cpu}^{Com} \leftarrow \max(Patt_{ID_s,cpu}) - Patt_{ID_s,cpu}$
    for $i = 1$: Number of unallocated VMs **do**
      $\phi_i \leftarrow$ PearsonCorrelation($Patt_{ID_s,cpu}^{Com}$,$\tilde{U}_{cpu}^{i}$)
    end for
    Find VM (ID_{VM}) with maximum ϕ & $\max(Patt_{ID_s,cpu} + \tilde{U}_{cpu}^{ID_{VM}}).F_{max} \leq F_{opt}$
    if $ID_{VM} == Null$ **then**
      $ID_s \leftarrow ID_s + 1$
    else
      Allocate VM ID_{VM} to server ID_s
    end if
  end if
end while

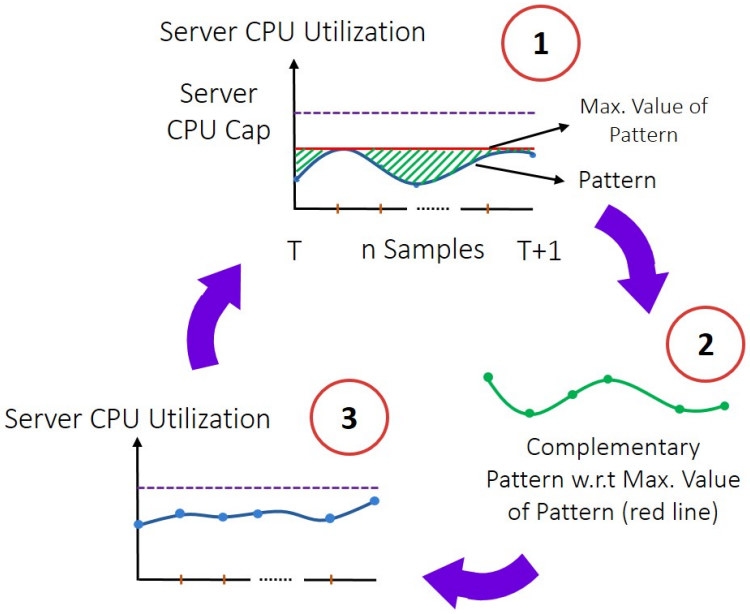

Figure 4.10 – VM selection steps for allocating to servers.

Having chosen the number of servers, the best server is found for each VM, maximizing the

following merit function:

$$\mathscr{M}_j^i = \omega_{cpu} \cdot \frac{\phi_{cpu}^{j,i}}{Dist_{cpu}^{j,i}} + \omega_{mem} \cdot \frac{\phi_{mem}^{j,i}}{Dist_{mem}^{j,i}}$$
$$\omega_{cpu} = \frac{Cap_{cpu}}{Cap_{cpu} + Cap_{mem}} \;\&\; \omega_{mem} = \frac{Cap_{mem}}{Cap_{cpu} + Cap_{mem}} \tag{4.2}$$

where $\phi_{cpu}^{j,i}$ and $\phi_{mem}^{j,i}$ exhibit the similarity of i^{th} VM's CPU utilization and memory footprint patterns with complementary CPU utilization and memory patterns of j^{th} server, respectively. To define this metric I use the Pearson Correlation, which is effective to judge the similarity on the shape of the patterns. However, as the Pearson Correlation cannot reflect the closeness of VM's CPU and memory patterns to the server CPU and memory cap, respectively, the euclidean distance ($Dist_{cpu}^{j,i}$ and $Dist_{mem}^{j,i}$) is incorporated into the metric. As a result, Eq. 4.2 demonstrates that $\mathscr{M}_j^i$ is high when i^{th} VM has both the same shape and lower distance to j^{th} server's caps. ω_{cpu} and ω_{mem} are weighting factors that need to be set with respect to the determined CPU and memory cap for filling up the server resources with the same importance.

Algorithm 10 The Proposed 2D VM Allocation Algorithm

Input: $\tilde{U}_{cpu}$, $\tilde{U}_{mem}$, Cap_{cpu}, and Cap_{mem}
Output: Allocating VMs to servers

for $i = 1 : N_{VM}$ **do**
  for $j = 1 : N_s$ **do**
    if $max_n(\tilde{U}_{cpu}^{i,n} + S_{cpu}^{j,n}) \leq Cap_{cpu}$ & $max_n(\tilde{U}_{mem}^{i,n} + S_{mem}^{j,n}) \leq Cap_{mem}$ **then**
      \\ CPU
      $Patt_{j,cpu}^{Com} \leftarrow \max(S_{cpu}^j) - S_{cpu}^j$
      $\phi_{cpu}^{j,i} \leftarrow$ PearsonCorrelation($Patt_{j,cpu}^{Com}$,$\tilde{U}_{cpu}^i$)
      $S_{rem,cpu}^j \leftarrow Cap_{cpu} - S_{cpu}^j$
      $Dist_{cpu}^{j,i} \leftarrow \|\tilde{U}_{cpu}^i - S_{rem,cpu}^j\|_2$
      \\ Memory
      $Patt_{j,mem}^{Com} \leftarrow \max(S_{mem}^j) - S_{mem}^j$
      $\phi_{mem}^{j,i} \leftarrow$ PearsonCorrelation($Patt_{j,mem}^{Com}$,$\tilde{U}_{mem}^i$)
      $S_{rem,mem}^j \leftarrow Cap_{mem} - S_{mem}^j$
      $Dist_{mem}^{j,i} \leftarrow \|\tilde{U}_{mem}^i - S_{rem,mem}^j\|_2$
      $\mathscr{M}_j^i \leftarrow$ Compute efficiency using Eq. 4.2
    end if
  end for
  $ID_s \leftarrow$ Find server with max $\mathscr{M}^i$
  Allocate VM i to server ID_s and update server's resources
end for

As described in Algorithm 10, first, one VM (i^{th} VM) is selected and then the best server is tried to be found among all (N_s) for it. For each candidate server (j^{th} server), it is checked whether the server has enough resources for hosting the VM at each sample in time slot T. If the server has sufficient remaining CPU and memory capacity ($S^j_{rem,cpu}$ and $S^j_{rem,mem}$), $\mathscr{M}^i_j$ is computed for the server. Finally, the VM is allocated to the server which has the maximum $\mathscr{M}^i$, and the target server's resources are updated (lines 17 and 18).

After allocation, for both cases, based on the real VMs CPU utilization, the best frequency level is online set for each server per sample to guarantee QoS.

4.6.3 Simulation Framework Validation

The GEM5 cycle-accurate simulator [206] is used to simulate the server architecture described in Section 4.3.2.2. In order to understand the effect of DVFS on performance, the QoS degradation is computed taking as a baseline the execution time on the x86 server discussed in Section 4.3.4. Then, the virtualized applications are simulated in GEM5 for different frequency levels ranging from 2.5GHz down to 100MHz.

To validate the correctness of the results provided by the GEM5 simulator, the applications are run on two real hardware platforms based on x86 and ARM. The execution times of Cavium ThunderX are compared with the ones obtained via GEM5 while matching the exact same architectural configuration. The error obtained was below 10%, showing that GEM5 is able to accurately simulate the workloads. The execution time for each workload, on all three platforms are shown in Table 4.2. The QoS limit is a 2x degradation of the execution time on x86 based platform, as already discussed. The Cavium server exhibits the worst execution time. After the modifications undertaken, the NTC server architecture outperforms Cavium by a factor of 1.25x to 1.76x. These results are due to the improved memory sub-system and the incorporation of the OoO processor in the presented architecture.

4.6.4 Experimental Results for The Server and Data Center Based On The NTC-Based Modified Cavium ThunderX Architecture

This section first explores the energy-performance trade-offs on NTC server. Then, the effectiveness of the proposed method is investigated at data center level.

Table 4.2 – NTC server and Cavium ThunderX QoS analysis

Application	Intel x86 @2.66 GHz	2x Degrad. Intel (QoS limit)	Cavium @2GHz	NTC Server @2GHz
low-mem	0.437	0.873	0.733	0.582
mid-mem	1.564	3.127	5.035	2.926
high-mem	3.455	6.909	11.943	6.765

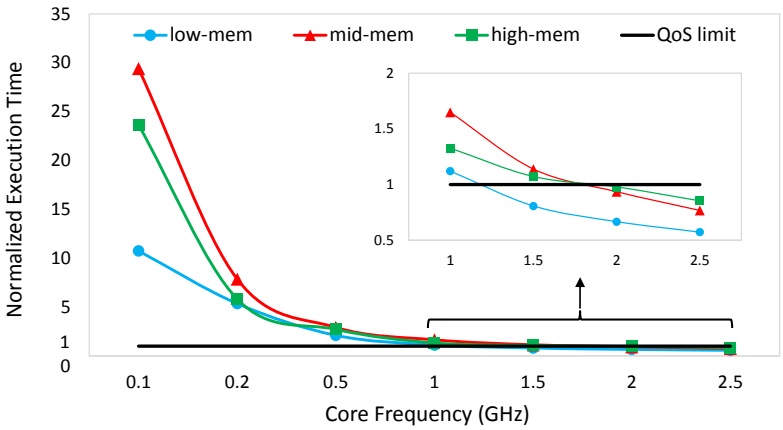

Figure 4.11 – Execution time normalized to QoS limit for different workloads.

4.6.4.1 Server-Level Results

1) Quality-of-Service (QoS)

QoS requirements of virtualized applications make it unclear whether this technology is suitable for server processors. To check for QoS requirements being met for VM workloads on NTC server, normalized execution time to QoS limit is shown in Fig. 4.11. It can be seen that high-mem and mid-mem workloads meet QoS requirement till a minimum frequency of 1.8GHz, whereas low-mem can scale down to 1.2GHz. In conclusion, the frequency of the cores can be reduced until meeting the 2x degradation constraint for virtualized applications on new architecture.

2) Energy Efficiency

Fig. 4.12 shows the benefits of reducing DVFS on server energy efficiency (i.e., the total number of UIPS at the chip level, divided by the total power consumption of the server, as defined in the previous section for NTC-based scale-out server). The optimal efficiency point is around 1.2GHz for high-mem, and around 1.5GHz for low-mem and mid-mem. The energy efficiency decreases with increasing memory utilization, firstly, because of higher active memory power, and secondly, because more memory accesses increase the amount of stalls and the WFM cycles.

3) Trade-offs Discussion

As shown in Section 4.5.2, workloads can tolerate low frequencies if only core power is considered, thus enabling NTC operation to reduce core power consumption. However, not all server components scale with the core voltage, shifting the most energy-efficient point to a higher frequency. The results showed that frequency can be reduced to 1.2GHz for high-mem and 1.5GHz for low-mem and mid-mem. But, to guarantee the QoS requirements, the fre-

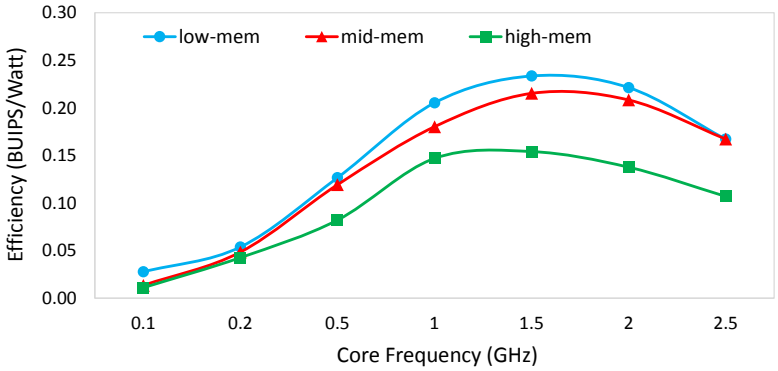

Figure 4.12 – Server efficiency as UIPS/Watt under different core frequencies on new NTC-based architecture.

quency level should be scaled up to 1.8GHz for mid-mem and high-mem; while for low-mem (CPU-bounded tasks) the optimal frequency (i.e., 1.5GHz) still meets the QoS limit.

4.6.4.2 Data Center-Level Results

At the data center level, I compare the proposed EPACT policy against two other energy-aware methods:

- COAT: COnsolidation-Aware allocaTion [20].
- COAT-OPT: COAT with an OPTimal fixed cap (optimal server frequency) when the worst-case data center power consumption is minimum.

and the proposed approach is evaluated in terms of Service-Level Agreement (SLA) violations and overall energy consumption.

1) Service-Level Agreement (SLA) Violation

Figure 4.13 shows violation, defined as the number of overutilized servers (i.e., the aggregated CPU or memory utilization among co-located VMs is beyond the CPU and memory cap), during each time slot for a time horizon of one week. Violations can only occur due to miss-prediction on the VM usage, especially during abrupt workload changes. EPACT provides a drastic reduction of the violations compared to COAT and COAT-OPT. This is because, in EPACT, the servers are not filled up to their maximum capacity, and there is some slack to increase frequency and compensate for violations. On the contrary, COAT and COAT-OPT have less control on violations during peak loads using a fixed cap.

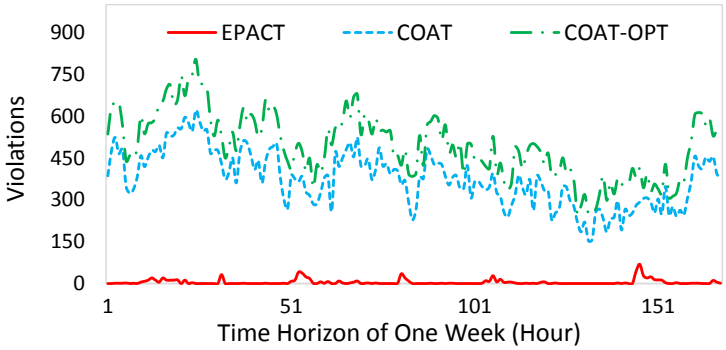

Figure 4.13 – Violations per time slot for a time horizon of one week.

2) Energy Consumption Analysis

Figure 4.14 shows the number of active servers per time slot for a time horizon of one week. COAT, being consolidation-based, reduces the number of active servers by 37% on average compared to EPACT. Despite this fact, EPACT achieves 45% and 10% energy savings in the best

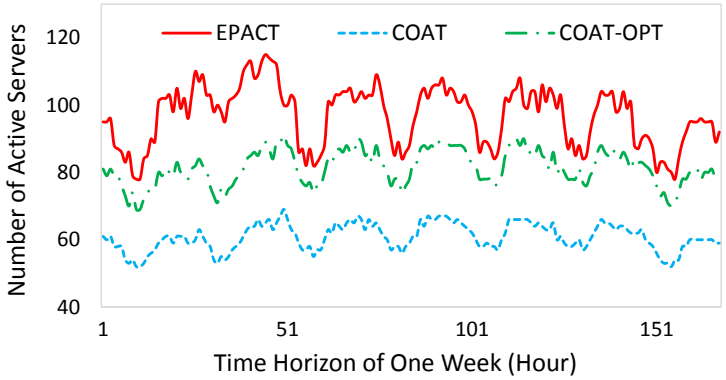

Figure 4.14 – Number of active servers for a time horizon of one week.

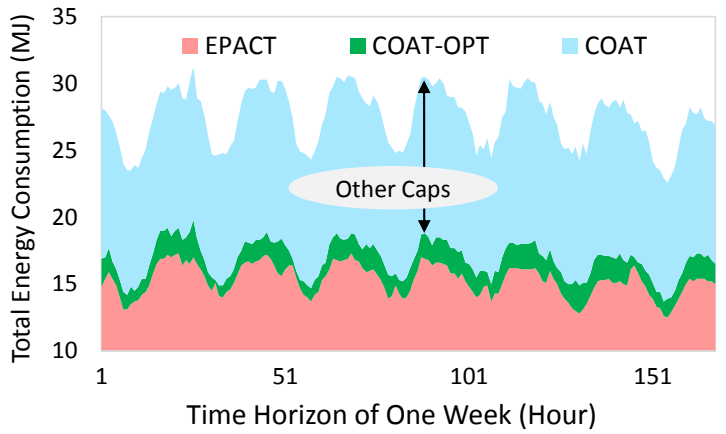

Figure 4.15 – Energy consumed by data center for a time horizon of one week.

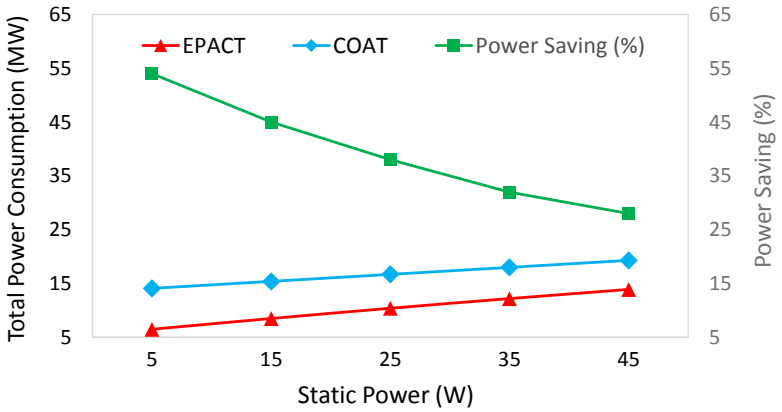

Figure 4.16 – Efficiency of proposed method under different static power.

and worst case compared to COAT and COAT-OPT, respectively (Fig. 4.15). This is because the optimal frequency is dinamycally found with respect to the time-varying data center CPU utilization and memory footprint, thus showing the inefficiency of consolidation-based techniques for NTC-based data centers.

3) Different Amount of Static Power Analysis

Figure 4.16 represents the effectiveness of the proposed algorithm (i.e., EPACT) compared to consolidation-based technique with maximum cap (COAT) when the static power (motherboard, fan, disk, etc.) increases from an efficient to a traditional power-hungry one. For higher static power consumption, optimal server frequency should be increased leading to higher CPU cap and lower number of active servers. These results prove that EPACT will be even more effective in future technologies, where static power is expected to decrease further.

4.7 Summary

As Moore's law continues to integrate more transistors on a chip, the end of Dennard scaling is unveiling an era of power-limited chips. NTC is a well-known voltage-scaling technique to reduce the energy consumption of the transistors. In this chapter, I first shed light on NTC in the context of server processors, demonstrating that significant improvements in energy efficiency can be achieved, while meeting the strict QoS requirements of workloads. Additionally, I showed that in order to substantially increase the energy efficiency of a server, all the server components of the system, not only the cores, need to be energy proportional.

Then, I explored the existing energy versus performance trade-offs using an accurate power modeling for the presented NTC servers based on the FD-SOI process technology, when VMs with different CPU utilization and memory footprint characteristics are executed. Finally, I proposed EPACT, a novel dynamic VM allocation method exploiting the given holistic knowledge of VMs characteristics and the power model to increase the energy proportionality of

next-generation NTC-based data centers while guaranteeing their QoS requirements. The proposed method has provided up to 45% energy savings when compared to conventional consolidation-based approach. Thus, the results demonstrate that the new NTC servers have created a completely new and promising (from an energy-efficiency viewpoint) research space on novel workload allocation techniques for next-generation data centers.

5 Conclusions and Future Work

To conclude this thesis, hereafter I summarize the key contributions of my research work and highlight its impacts on the academia (in particular in the computer engineering and computer architecture communities) and the industry (for both data center users and service providers). Finally, at the end of this chapter, I provide important items for future research in the same direction based on the results obtained in this thesis.

5.1 Summary and Contributions

In this thesis I have proposed a set of system-level techniques to improve the energy efficiency of servers and cloud data centers. The following list provides a more detailed summary of the contributions introduced in the different chapters, and discusses the results of this thesis as follows:

- **Efficient Workload Allocation in Single Data Center:** In Chapter 2, I have proposed a multi-objective two-phase greedy heuristic and a multi-objective Machine Learning (ML)-based Virtual Machine (VM) allocation method for various data center scenarios, and compare them in terms of energy, Quality-of-Service (QoS), network traffic, migrations, and scalability. Both approaches exploit CPU-load and data correlations as key factors of workload characteristics, together with information about data center network topology. The strategies consolidate VMs into the minimum number of servers and racks, and set Dynamic Voltage and Frequency Scaling (DVFS) appropriately. Then, I have presented, for the first time in literature, a novel hyper-heuristic algorithm that exploits the benefits of both methods by dynamically finding the best algorithm, while allowing users to decide on the importance of each metric (objective). That is, this approach integrates the strengths of both heuristic and ML methods in highly dynamic environment. For optimality assessment, I have formulated an Integer Linear Programming (ILP)-based VM allocation method to minimize energy consumption and data communication in a single data center, which obtains optimal results, but is impractical at run-time.

Finally, I have provided an evaluation of the flexibility, scalability, benefits and drawbacks of heuristic versus ML methods for the highly dynamic and complex VM allocation problem. The experimental results have shown that the heuristic, ML, and hyper-heuristic methods reach almost similar results in terms of energy consumption (< 2% difference), consuming only up to 6% more energy than the optimal solution. However, the ML method improves server-to-server network traffic by up to 24% and reduces execution time by up to 480x when compared to conventional approaches for large-scale problems. On the other hand, the heuristic algorithm results in better QoS and lower traffic in the upper layers of the network structure, due to its fine-tuning capabilities. Also, the hyper-heuristic obtains better trade-offs for different objectives between solution quality and computational overhead. These results demonstrate the benefits of the proposed approach when compared to other state-of-the-art techniques.

Publication: The work presented in this chapter, including all the proposed allocation methods, has been published in the journal *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* [207].

- **Multi-Objective Optimization in Green Data Centers:** In Chapter 3, I have first presented a multi-level and multi-objective framework and a system model used in green virtualized data centers exploiting different battery technologies, management schemes, and implementation policies. This system introduces a Hybrid Electric Systems (HES) architecture to replace standard Uninterruptible Power Supply (UPS) systems, which allows an active management and the full exploitation of the energy buffers for the locally-generated renewable energy, to minimize the operational cost. I have also designed a dedicated control loop which connects the VMs allocation scheme to the HES manager and optimizes the resources in real-time. At the same time, the proposed modular structure allows to use both general purpose models for performance evaluation and feasibility analysis.

  Second, I have presented a novel method to tackle the challenges of operational cost optimization and energy-performance trade-off on resource-constrained green geo-distributed data centers on the introduced framework and system model. I have proposed a two-phase multi-objective VM placement (i.e., clustering and allocation) algorithm along with a dynamic migration technique. The first phase, i.e., global controller, creates clusters of VMs that will be sent to each data center exploiting both CPU-load and data correlations (VMs characteristics) based on data centers status (current electricity price, battery information, renewable energy forecast and VMs utilization prediction). The second phase, i.e. local controller which is used in each data center, allocates VMs to servers considering CPU-load correlation. The proposed algorithm minimizes the operational costs, data center energy consumption, network traffic and response time while maximizing the renewable energy and battery usage. Since the whole problem is optimized to find the best solution based on load and renewable forecast information, a low-complexity rule-based green controller is adopted to compensate the difference between real and forecasted information. Experimental results have shown that, using

the proposed method, up to 54%, 14%, and 10% improvements can be obtained for operational cost, energy consumption and performance, respectively, compared to state-of-the-art approaches.

Third, I have introduced ECOGreen, a novel strategy to tackle the challenge of green data centers participation in Regulation Service (RS) reserves provision (one of the most interesting power markets programs from the customer's perspective), considering the demand-side renewable and Electrical Energy Storage (EES) power. I have first presented, for the first time in the literature, a mathematical solution to jointly find the best average power and reserve values in the bidding problem, together with the number of active servers needed in the VM allocation phase. The EES utilization and renewable power usage are optimized in a resource-constrained green data center based on the introduced fine-grained system model. Then, I have proposed a runtime approach that dynamically regulates the data center power consumption following the RS signal, while also guaranteeing the QoS limit. The runtime policy utilizes VMs recourse limit control (i.e., dynamically changing the server resources allocated to VMs), EES and renewable power in decision making to minimize the signal tracking error. In my experiments, I have compared my proposed holistic strategy, i.e., ECOGreen, in terms of monetary cost, total power consumption breakdown of green data center, QoS degradation, and EES efficiency analysis against different state-of-the-art approaches. The experimental results have demonstrated that ECOGreen enables the green data center to provide 76% of its power consumption on average to the power market due to largely operating on renewable energy and EES. This amount of reserves helps the data center to save up to 71% in electricity cost compared to the state-of-the-art approaches. Moreover, the proposed strategy reduces monetary cost by 35% and 61% when compared to the same approach but without participation in RS reserves, and without the use of green energy (renewables and EES), respectively. ECOGreen has also obtained up to 48% and 28% renewable and EES utilization improvements, respectively, at the expense of battery State-of-Health (SoH) decreasing when compared to other approaches. Nonetheless, the battery ageing shows a lifetime longer than 15 years. Overall, ECOGreen ensures achieving the best trade-off between different objectives when compared to all other methods.

Publications: The green data center system model and framework have been published in *Springer Handbook of Hardware/Software Codesign* [208]. The second part of this work, i.e., multi-objective VM placement for geo-distributed data centers, was accepted for publication in *Design Automation and Test in Europe (DATE)* [104]. The last part, i.e., electricity cost optimization for green data centers in emerging power markets has recently been submitted to the journal *IEEE Transactions on Sustainable Computing (T-SUSC)*.

- **Towards Next-Generation Near-Threshold Data Centers:** As an effect of post Dennard scaling, computing servers have become power-limited, and techniques such as NTC together with new system-level approaches must be used to improve their energy effi-

ciency. In Chapter 4, I have explored the existing energy versus performance trade-offs using an accurate power modeling for NTC servers based on the Fully Depleted Silicon On Insulator (FD-SOI) process technology, considering two different architectures: i) the scale-out architecture optimized for scale-out data center workloads and used for overall evaluation of the benefits of NTC servers, and ii) the ARM-based Cavium ThunderX architecture tuned to virtualized applications, when VMs with different CPU utilization and memory footprint characteristics are executed. The results have demonstrated that significant improvements in energy efficiency can be achieved, while meeting the QoS requirements of workloads. Additionally, I have shown that in order to substantially increase the energy efficiency of a server, all its components, not only the cores, need to be energy proportional. Also, results have shown that NTC servers create a completely new and promising (from an energy-efficiency viewpoint) research space on novel workload allocation techniques for next-generation data centers. Therefore, I have proposed EPACT, a novel dynamic VM allocation method exploiting the given holistic knowledge of VMs characteristics and the power model to increase the energy proportionality of next-generation NTC-based data centers while guaranteeing their QoS requirements. The proposed method has provided up to 45% energy savings when compared to conventional consolidation-based approach, indicating the inefficiency of consolidation-based techniques for NTC-based data centers.

Publications: This work has been highly appreciated by the community and has lead to two publications at the *Design Automation and Test in Europe (DATE)* Conference. More precisely, the first version of the approach at server-level on scale-out architecture was published in 2016 [209], as a result of the collaboration between the PARSA lab at EPFL (who provided the scale-out architecture model) and ETHZ (who provided the power model). The second publication, which proposed a modified Cavium ThunderX platform as well as the data center-level policies in 2018 [210], was result of the collaboration with another PhD student at ESL (who developed the NTC server model) and also ETHZ (who provided accurate and detailed power model).

5.2 Future Work

Based on my research findings and achievements revealed in this thesis, in this section I specify the future directions that can be taken in the field to improve the efficiency of data centers. For this purpose, I categorize and highlight the future lines that can derive from my research work into short- and long-term lines.

In the following, a set of short-term research lines is presented that continues the proposed approach to manage the modern data centers efficiently.

- **Electricity Cost Management in Emerging Power Markets:** In Chapter 3, I focused on the electricity cost optimization for a green data center. However, an interesting area of research is to manage green geo-distributed data centers in emerging power markets. In

this context, the goal is to optimize multiple data centers cost that belong to a provider. For this purpose, a novel centralized-distributed method is required to determine the average power consumption and reserves values per data center considering the data centers loads, renewable and battery energy, while minimizing the total electricity cost of data centers and maximizing the provider's revenue. The interaction between data centers (i.e., migrating workloads) helps data center owners to provide higher amount of reserves to the power market due to largely operating on the temporal and regional diversity of renewable energy. Also, a distributed algorithm is needed for each data center to regulate its power with respect to power market requirements. Moreover, one interesting approach could be investigating stochastic methods to find near-optimal solutions, instead of taking into account the worst-case scenario, when the RS signal is predictable or its statistical information is available for a certain time-ahead period.

- **Heterogeneous Architectures and Servers in Data Centers:** In Chapter 4, a NTC-based data center was modeled and we showed how new system-level approaches must be used to increase energy proportionality. However, data centers are usually equipped with heterogeneous servers (i.e., different number and types of servers). Two main reasons have led to this situation today. The first one is due to the maintenance and evolution of the components: different generations of servers are commonly used in data centers since the owners are not replacing all the older systems at each update. The second reason is driven by the idea that heterogeneity might be the key to achieving energy-proportional computing, since it opens the opportunity to assign the workloads to the right systems and further improve the energy efficiency for both Information Technology (IT) equipment and cooling systems at data center scale.

  On the other hand, due to the heterogeneity of the servers in the data center, different scheduling strategies may lead to different scenarios to tackle the performance and energy efficiency trade-off by selecting the right servers. While a specific scheduling strategy may optimize one objective, the other objectives can be conflicting and make the optimization problem difficult. In this case, I showed the inefficiency of consolidation-based techniques for NTC-based data centers to improve the energy efficiency, while this technique is the most suitable for x86 server architectures designed to meet performance goals. Therefore, developing new methods for heterogeneous data centers for different workloads with different performance constraints is needed.

- **Server Failure Uncertainty Model (Performance Variability):** Although most data center optimization algorithms assume that all the parameters of the VM allocation problem are fixed and known (e.g., the load of servers, the servers availability, etc), they are actually dominated by uncertainties. Application performance variability is one of the important uncertainties in data centers, and may originate from both internal and external sources (e.g., aged or failing hardware, thermal control, orphan processes or operating system issues). Most of these anomalies lead to performance degradation during host operation. Hence, improving the energy efficiency of data centers while guaranteeing QoS, together with detecting performance variability of servers caused

by either hardware or software failures, are two of the major challenges for efficient resource management of large-scale cloud infrastructures. Previous works in the area of dynamic VM consolidation are mostly focused on addressing the energy challenge, but fall short in proposing comprehensive, scalable, and low-overhead approaches that jointly tackle energy efficiency and performance variability. Therefore, as data centers are very complex and highly dynamic systems in reality, real online low-overhead energy-efficient management must cope with unforeseen events (e.g., a server may be damaged or may need to be restarted because of an urgent overheating and eventually failing servers).

To address the aforementioned challenge, we proposed a multi-agent ML-based approach in collaboration with a visiting PhD student at ESL. This work has been submitted to the journal *IEEE Transactions on Services Computing*, and a revision has been requested.

In the following, long-term research lines are presented to manage the hybrid cloud data centers and fog computing as a new paradigm to reduce the energy consumption, communication and computation time overhead of cloud data centers.

- **Co-Allocating Workloads with Different Characteristics:** Cloud applications are primarily used for data-intensive and latency-sensitive jobs, search engines, business processing, social-media networking, data warehousing and big-data analytics, and they are characterized by their dataset size, memory-access pattern, and service model applied. On the other side, High Performance Computing (HPC) applications such as scientific computing loads typically occupy many server nodes, run for a long time, and include heavy data exchange and communication among the threads of the application. These HPC applications can run with relaxed QoS constraints (less sensitive to their required resources) whereas interactive workloads are highly sensitive to latency. As users can access these cloud services and applications anytime and anywhere, exploiting the mixed nature of workloads behaviors (e.g., scale-out applications mixed with virtualized batch jobs) on different server architectures is needed to utilize the server and data center resources efficiently. For instance, scale-out applications (e.g., web search, web serving, data analytics, etc.) requests are independently distributed across a servers cluster, and the load of servers is dependent on the number of clients/queries. Hence, during the low client requests, the traditional virtualized workloads can be migrated and co-located with scale-out applications to maximize the servers utilization. In addition, investigating the mixed nature of workloads behaviors associated with next-generation NTC-based data centers is another interesting direction.
- **Resource Provisioning in Fog-Cloud Computing:** Fog computing as an extension of cloud computing has been introduced as a new paradigm to distribute computation at the edge of the network. In fog computing, the goal is to move decision and computation closer to the data sources, for instance to network switches and mobile service stations,

to reduce response time and latency. While fog computing brings several benefits like highly mobility support, low latency, distributed real-time applications (e.g., Internet of Things (IoT) applications and services in healthcare [211, 212, 213, 214, 215, 216, 217, 218]), and heterogeneity, it also provides various challenges. To maximize the utilization rate of resources, several works have proposed load balancing on different components of the fog computing nodes and environment. In this case, the idle period of computational nodes as well as network traffic become minimized [219, 220]. However, keeping the computational nodes powered-on for a long period increases the power consumption and electricity cost, when the nodes are not efficiently utilized.

On the other hand, in order to mange the cloud data centers power usage, fog computing infrastructure can host several power-hungry applications to accordingly improve the cloud data centers energy efficiency. This process by fog computing devices also helps to avoid network congestion, high latency to deliver user requests, and QoS degradation. Therefore, developing new policies to improve the efficiency of the fog-cloud interaction is an interesting and challenging future direction.

A Appendix: Data Center Monetary Cost Evaluation in Power Markets

This appendix provides additional information on the experimental results presented in the Section 3.6.5.1 of Chapter 3.

Figure A.1 shows two corner points on determining the average power consumption ($\bar{P}$) and the amount of reserves (R) based on the estimated data center power consumption, renewable energy, and Electrical Energy Storage (EES) status. Point 1 exhibits high R because it has low workload (results in ~25% resource utilization) and high Photovoltaic (PV) availability; while Point 2 depicts low R because it has higher workload (~40% utilization) and without PV energy

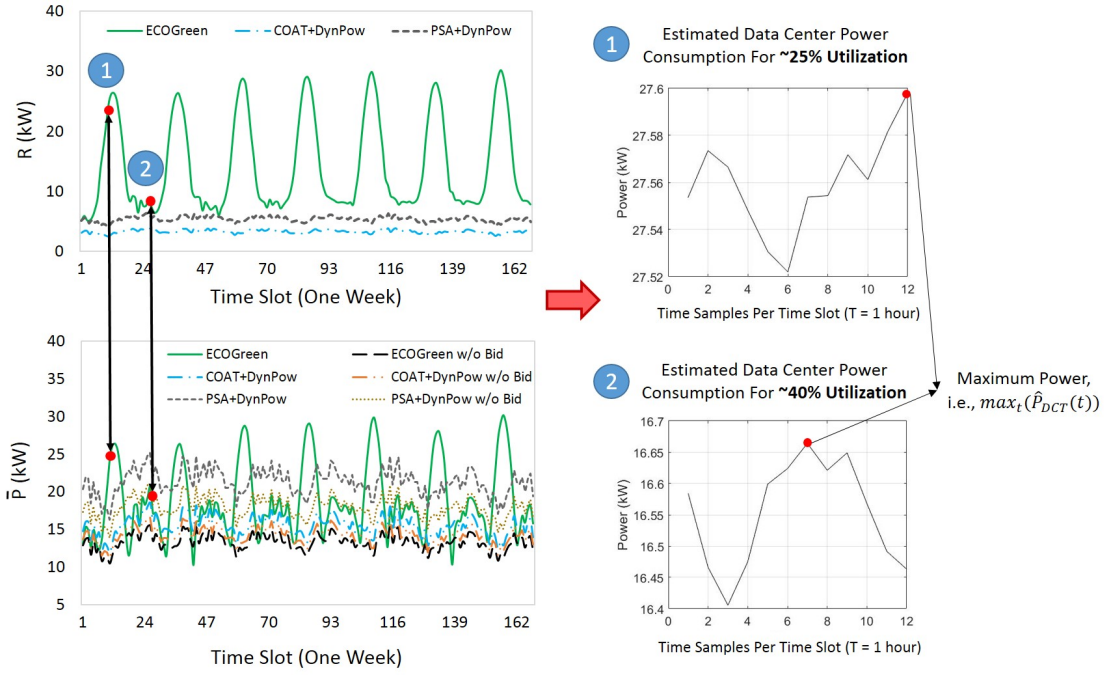

Figure A.1 – $\bar{P}$ and R for two corner cases (two time slots): 1) the estimated data center power consumption when its utilization is ~25% during the time slot, and 2) for ~40% data center utilization.

availability.

Tables A.1-A.4 shows a specific time sample in the time slot of points 1 and 2, representing the solution of the proposed algorithm (i.e., ECOGreen) to follow the Regulation Service (RS) signal ($z(t)$) based on the current data center situation.

According to the Eq. 3.21, the best solution is to try to have the closest $\bar{P}$ and R, since this minimizes cost. Moreover, for more cost reduction, the error term should be minimized. Therefore, the largest the R, the lesser error. Hence, In Table A.1, the ECOGreen drastically increases $\bar{P}$ and R (i.e., number of active servers), even for a low utilization. This is because the available renewable energy is high, which allows the policy to drastically reduce cost, while also reducing energy consumption from the power grid. On the contrary, Table A.3 shows a low R when the available renewable energy is low. In this case, $\bar{P}$ should be decreased to reduce the cost (getting close to R as possible).

Figure A.2 indicates that all policies "with bidding" draw more energy from the power grid than policies "w/o bidding". This is because of reducing the cost by potentially equalizing the $\bar{P}$ and R (not only reducing $\bar{P}$). As a result, in order to avoid the high usage of the power grid (i.e., making the data center "greener"), the proposed algorithm (i.e., EcoGreen) can easily be restricted to "not bid" in the power market when the data center can be self-sustained by green energy sources.

Table A.1 – Time slot of point 1 - the proposed algorithm solution and results to follow the RS signal using energy sources and server resources allocated to workloads for one time sample (every 4 sec.) in the time slot under the current situation as data center utilization: ≈25%, battery charge: 98%, and real PV available: 26.9 kW

	ECOGreen	**COAT_DynPow**	**PSA_DynPow**
$\bar{P}$ (kW)	25	11.9	16.2
R (kW)	24.5	2.5	4.1
Number of Active Servers	239	74	102
Grid Power (kW) - P_{Grid}	10.06	10.403	13.86
Used PV (kW) - P_{PV}	17.59	0	0
Used Bat (kW) - P_{EES}	0	0	0
P_{DCT} - Data Center Power (kW) (***before*** *RS tracking*)	27.65	11.102	13.91
P_{DCT} - Data Center Power (kW) (***with*** *RS tracking*)	27.65 *no change*	10.403 *workload resource reduction to follow RS*	13.86 *workload resource reduction to follow RS*
QoS Degradation (times) (*1x is no degradation*)	1x *no degradation*	1.24x *QoS degradation*	1.013x *QoS degradation*
Data Center Cost ($) (*for 4 sec, i.e., 1 point*)	0.000051	0.941	1.22

Table A.2 – Extra time sample with low renewable energy in time slot of point 1 - the proposed algorithm solution and results to follow the RS signal under the current situation as data center utilization: ≈28%, battery charge: 99%, and real PV available: 0 kW

	ECOGreen	**COAT_DynPow**	**PSA_DynPow**
$\bar{P}$ (kW)	12.601	13.424	18.428
R (kW)	5.089	2.843	4.635
Number of Active Servers	84	84	116
Grid Power (kW) - P_{Grid}	9.995	11.968	16.052
Used PV (kW) - P_{PV}	0	0	0
Used Bat (kW) - P_{EES}	2.499 (*discharge*)	0	0
P_{DCT} - Data Center Power (kW) (***before*** *RS tracking*)	12.494	12.494	15.703
P_{DCT} - Data Center Power (kW) (***with*** *RS tracking*)	12.494 *no change*	11.969 *workload resource reduction to follow RS*	16.052 *workload resource reduction to follow RS*
QoS Degradation (times) (*1x is no degradation*)	1x *no degradation*	1.15x *QoS degradation*	1x *no degradation*
Data Center Cost ($) (*for 4 sec, i.e., 1 point*)	0.752	1.059	0.138

Table A.3 – Time slot of point 2 - the proposed algorithm solution and results to follow the RS signal under the current situation as data center utilization: ≈41%, battery charge: 92%, and real PV available: 0 kW

	ECOGreen	**COAT_DynPow**	**PSA_DynPow**
$\bar{P}$ (kW)	20.6	18.5	25.2
R (kW)	8.6	4	6.4
Number of Active Servers	116	116	159
Grid Power (kW) - P_{Grid}	19.990	18.178	24.706
Used PV (kW) - P_{PV}	0	0	0
Used Bat (kW) - P_{EES}	-2.748 (*charge*)	0	0
P_{DCT} - Data Center Power (kW) (***before*** *RS tracking*)	16.744	16.744	21.056
P_{DCT} - Data Center Power (kW) (***with*** *RS tracking*)	17.242 *workload resource reduction to follow RS*	18.178 *workload resource reduction to follow RS*	24.706 *workload resource reduction to follow RS*
QoS Degradation (times) (*1x is no degradation*)	1x *no degradation*	1x *no degradation*	1x *no degradation*
Data Center Cost ($) (*for 4 sec, i.e., 1 point*)	1.203	1.459	1.881

Table A.4 – Extra time sample with high renewable energy in time slot of point 2 - the proposed algorithm solution and results to follow the RS signal under the current situation as data center utilization: ≈39%, battery charge: 98%, and real PV available: 29 kW

	ECOGreen	COAT_DynPow	PSA_DynPow
$\bar{P}$ (kW)	25.025	17.908	24.628
R (kW)	25.025	3.778	6.185
Number of Active Servers	244	112	155
Grid Power (kW) - P_{Grid}	5.347	14.937	19.764
Used PV (kW) - P_{PV}	24.336	0	0
Used Bat (kW) - P_{EES}	0	0	0
P_{DCT} - Data Center Power (kW) (***before*** *RS tracking*)	29.683	16.443	20.756
P_{DCT} - Data Center Power (kW) (***with*** *RS tracking*)	29.683 *no change*	14.937 *workload resource reduction to follow RS*	19.765 *workload resource reduction to follow RS*
QoS Degradation (times) (*1x is no degradation*)	1 *no degradation*	1.41x *QoS degradation*	1.26x *QoS degradation*
Data Center Cost ($) (*for 4 sec, i.e., 1 point*)	0	1.413	1.848

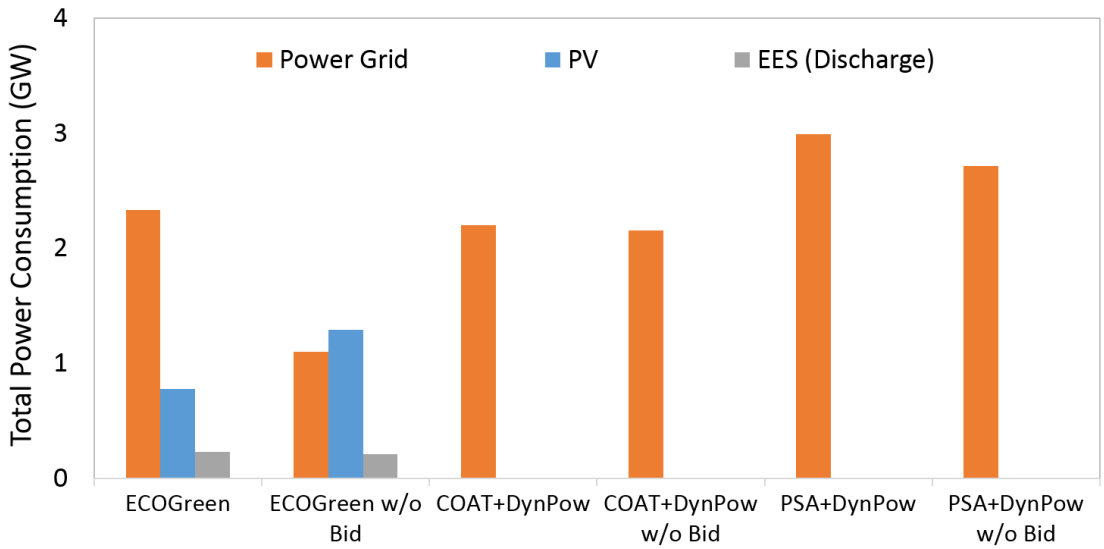

Figure A.2 – Average power consumption ($\bar{P}$) and reserves (R) for two corner cases (two time slots): 1) the estimated data center power consumption when its utilization is ~25% during the time slot, and 2) for ~40% data center utilization.

Bibliography

[1] R. H. Katz, "Tech titans building boom," *IEEE Spectrum*, vol. 46, no. 2, pp. 40–54, Feb 2009.

[2] A. Qureshi, "Power-demand routing in massive geo-distributed systems," in *Ph.D. dissertation, MIT*, 2010.

[3] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2012, pp. 211–222.

[4] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *Springer Middleware*, 2011, pp. 143–164.

[5] J. Clark, "It now 10 percent of world's electricity consumption, http://www.theregister.co.uk/2013/08/16/it_electricity_use_worse_than_you_thought/." Report Finds, The Register, August 2013.

[6] G. Cook, "How clean is your cloud?" in *Greenpeace International Technical Report*, 2012.

[7] Greenpeace's clean energy report, https://techcrunch.com/2017/01/10/apple-facebook-and-google-top-greenpeaces-clean-energy-report/.

[8] C. Stewart and K. Shen, "Some joules are more precious than others: Managing renewable energy in the datacenter," in *Workshop on Power Aware Computing and Systems*, 2009.

[9] C. Bergonzini, D. Brunelli, and L. Benini, "Comparison of energy intake prediction algorithms for systems powered by photovoltaic harvesters," *Elsevier Microelectronics Journal*, vol. 41, no. 11, pp. 766–777, Nov. 2010.

[10] M. Rossi and D. Brunelli, "Electricity demand forecasting of single residential units," in *IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS)*, 2013, pp. 1–6.

[11] M. Ghasemi, M. Mohaqeqi, and M. Kargahi, "Joint management of processing and cooling power based on inaccurate thermal information in a stochastic real-time system,"

in *ACM International Conference on Real Time and Networks Systems (RTNS)*, 2015, pp. 45–54.

[12] E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2009, pp. 145–150.

[13] A. Verma, P. Ahuja, and A. Neogi, "pmapper: Power and migration cost aware application placement in virtualized systems," in *Springer Middleware*, 2008, pp. 243–264.

[14] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1366–1379, July 2013.

[15] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proceedings of the IEEE Conference on Information Communications (INFOCOM)*, 2010, pp. 1154–1162.

[16] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware vm placement for cloud systems," in *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2012, pp. 498–506.

[17] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Clearing the clouds: a study of emerging scale-out workloads on modern hardware," *ACM SIGARCH Comput. Archit. News*, vol. 40, no. 1, pp. 37–48, 2012.

[18] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[19] F. Research, "The future of data center wide-area networking," 2010.

[20] J. Kim, M. Ruggiero, D. Atienza, and M. Lederberger, "Correlation-aware virtual machine allocation for energy-efficient datacenters," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013, pp. 1345–1350.

[21] D. de Oliveira, K. A. C. S. Ocaña, F. Baião, and M. Mattoso, "A provenance-based adaptive scheduling heuristic for parallel scientific workflows in clouds," *Springer Journal of Grid Computing*, vol. 10, no. 3, pp. 521–552, Sep 2012.

[22] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari, "Server workload analysis for power minimization using consolidation," in *Proceedings of the Conference on USENIX Annual Technical Conference*, 2009.

[23] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in

IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, pp. 400–407.

[24] Z. A. Mann, "Allocation of virtual machines in cloud data centers–a survey of problem models and optimization algorithms," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 11:1–11:34, 2015.

[25] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *Proceedings of the ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2012, pp. 175–186.

[26] A. Pahlavan, M. Momtazpour, and M. Goudarzi, "Power reduction in hpc data centers: a joint server placement and chassis consolidation approach," *Springer Journal of Supercomputing*, vol. 70, pp. 845–879, 2014.

[27] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *ACM Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, 2010.

[28] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient resource provisioning in compute clouds via vm multiplexing," in *ACM Proceedings of the International Conference on Autonomic Computing (ICAC)*, 2010, pp. 11–20.

[29] K. Halder, U. Bellur, and P. Kulkarni, "Risk aware provisioning and resource aggregation based consolidation of virtual machines," in *IEEE International Conference on Cloud Computing (CLOUD)*, 2012, pp. 598–605.

[30] A. Amokrane, M. F. Zhani, R. Langar, R. Boutaba, and G. Pujolle, "Greenhead: Virtual data center embedding across distributed infrastructures," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 36–49, Jan 2013.

[31] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 909–928, 2013.

[32] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: A data center network virtualization architecture with bandwidth guarantees," in *ACM Proceedings of the International COnference (Co-NEXT)*, 2010, pp. 15:1–15:12.

[33] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *Proceedings of the ACM SIGCOMM Conference*, 2011, pp. 242–253.

[34] M. F. Zhani, Q. Zhang, G. Simona, and R. Boutaba, "Vdc planner: Dynamic migration-aware virtual data center embedding for clouds," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2013, pp. 18–25.

[35] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," in *Advances in Computers, Marvin V. Zelkowitz (editor)*, vol. 82. Academic Press, 2011, pp. 47–111.

[36] . Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres, and R. Bianchini, "Greenslot: Scheduling energy consumption in green datacenters," in *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov 2011, pp. 1–11.

[37] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: A profit maximization approach," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 1017–1025, June 2013.

[38] N. Sharma, S. Barker, D. Irwin, and P. Shenoy, "Blink: Managing server clusters on intermittent power," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 1, pp. 185–198, Mar. 2011.

[39] . Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and greenswitch: Managing datacenters powered by renewable energy," in *ACM Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2013, pp. 51–64.

[40] ——, "Designing and managing data centers powered by renewable energy," *IEEE Micro*, vol. 34, no. 3, pp. 8–16, May 2014.

[41] L. Gu, D. Zeng, S. Guo, and B. Ye, "Joint optimization of vm placement and request distribution for electricity cost cut in geo-distributed data centers," in *International Conference on Computing, Networking and Communications (ICNC)*, Feb 2015, pp. 717–721.

[42] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Proceedings of the IEEE Conference on Information Communications (INFOCOM)*, March 2010, pp. 1–9.

[43] J. Li, Z. Li, K. Ren, and X. Liu, "Towards optimal electric demand management for internet data centers," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 183–192, March 2012.

[44] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen, "Reducing electricity cost through virtual machine placement in high performance computing clouds," in *ACM Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2011, pp. 22:1–22:12.

[45] L. Gu, D. Zeng, S. Guo, and S. Yu, "Type-aware task placement in geo-distributed data centers with low opex using data center resizing," in *International Conference on Computing, Networking and Communications (ICNC)*, Feb 2014, pp. 211–215.

[46] L. Gu, D. Zeng, A. Barnawi, S. Guo, and I. Stojmenovic, "Optimal task placement with qos constraints in geo-distributed data centers using dvfs," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 2049–2059, July 2015.

[47] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. C. M. Lau, "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers," in *IEEE Conference on Computer Communications (INFOCOM)*, April 2014, pp. 118–126.

[48] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds," in *IEEE International Conference on Distributed Computing Systems*, June 2012, pp. 526–535.

[49] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.

[50] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, and A. Wolman, "Volley: Automated data placement for geo-distributed cloud services," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2010.

[51] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi, "Embedding virtual topologies in networked clouds," in *Proceedings of the ACM International Conference on Future Internet Technologies (CFI)*, 2011, pp. 26–29.

[52] N. Cordeschi, M. Shojafar, D. Amendola, and E. Baccarelli, "Energy-efficient adaptive networked datacenters for the qos support of real-time applications," *The Journal of Supercomputing*, vol. 71, no. 2, pp. 448–478, Feb 2015.

[53] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, Aug. 2009.

[54] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Geographical load balancing with renewables," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 39, no. 3, pp. 62–66, Dec. 2011.

[55] Z. Abbasi, M. Pore, and S. K. S. Gupta, "Online server and workload management for joint optimization of electricity cost and carbon footprint across data centers," in *IEEE 28th International Parallel and Distributed Processing Symposium*, May 2014, pp. 317–326.

[56] M. Ghamkhari and H. Mohsenian-Rad, "Optimal integration of renewable energy resources in data centers with behind-the-meter renewable generator," in *IEEE International Conference on Communications (ICC)*, June 2012, pp. 3340–3344.

[57] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, 2011, pp. 221–232.

[58] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. K. Fathy, "Energy storage in datacenters: What, where, and how much?" in *Proceedings of the ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, 2012, pp. 187–198.

[59] X. Deng, D. Wu, J. Shen, and J. He, "Eco-aware online power management and load scheduling for green cloud datacenters," *IEEE Systems Journal*, vol. 10, no. 1, pp. 78–87, March 2016.

[60] X. Xiang, C. Lin, F. Chen, and X. Chen, "Greening geo-distributed data centers by joint optimization of request routing and virtual machine scheduling," in *IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, Dec 2014, pp. 1–10.

[61] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[62] C. Bhringer, A. Lschel, U. Moslener, and T. F. Rutherford, "Eu climate policy up to 2020: An economic impact assessment," *Energy Economics*, vol. 31, pp. 295 – 305, 2009.

[63] EIA, "Annual energy outlook," http://www.eia.gov/forecasts/aeo, 2014.

[64] A. L. Ott, "Experience with pjm market operation, system design, and implementation," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 528–534, 2003.

[65] NYISO, "Manual 2: Ancillary services manual, v3.26," http://www.eia.gov/forecasts/aeo, 2013.

[66] D. Aikema, R. Simmonds, and H. Zareipour, "Data centres in the ancillary services market," in *IEEE International Green Computing Conference (IGCC)*, 2012, pp. 1–10.

[67] H. Chen, A. K. Coskun, and M. C. Caramanis, "Real-time power control of data centers for providing regulation service," in *IEEE Conference on Decision and Control (CDC)*, 2013, pp. 4314 – 4321.

[68] B. Kirpes and S. Klingert, "Evaluation process of demand response compensation models for data centers," in *ACM Proceedings of the International Workshop on Energy Efficient Data Centres (E2DC)*, 2016, pp. 4:1–4:6.

[69] H. Chen, C. Hankendi, M. C. Caramanis, and A. K. Coskun, "Dynamic server power capping for enabling data center participation in power markets," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 122–129.

[70] H. Chen, M. C. Caramanis, and A. K. Coskun, "The data center as a grid load stabilizer," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014, pp. 105–112.

[71] M. Shafique, S. Garg, T. Mitra, S. Parameswaran, and J. Henkel, "Dark silicon as a challenge for hardware/software co-design," in *International Conference on Hardware/-Software Codesign and System Synthesis (CODES+ISSS)*, Oct 2014, pp. 1–10.

[72] D. Markovic, C. C. Wang, L. P. Alarcon, T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, Feb 2010.

[73] D. Jacquet, F. Hasbani, P. Flatresse, R. Wilson, F. Arnaud, G. Cesana, T. D. Gilio, C. Lecocq, T. Roy, A. Chhabra, C. Grover, O. Minez, J. Uginet, G. Durieu, C. Adobati, D. Casalotto, F. Nyer, P. Menut, A. Cathelin, I. Vongsavady, and P. Magarshack, "A 3 ghz dual core processor arm cortex tm -a9 in 28 nm utbb fd-soi cmos with ultra-wide voltage range and energy efficiency optimization," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 812–826, April 2014.

[74] P. Clarke, "Globalfoundries preps 12nm fdsoi process," http://www.eenewsanalog.com/news/globalfoundries-preps-12nm-fdsoi-process, 2016.

[75] L. Gwennap, "Fd-soi offers alternative to finfet," Posted at https://www.globalfoundries.com/sites/default/files/fd-soi-offers-alternative-to-finfet.pdf, 2016.

[76] A. Substrates, "Fd-soi keeps moor's law on track," Feb 2014.

[77] A. Iranfar, S. N. Shahsavani, M. Kamal, and A. Afzali-Kusha, "A heuristic machine learning-based algorithm for power and thermal management of heterogeneous mpsocs," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, July 2015, pp. 291–296.

[78] P. Cowling, G. Kendall, and E. Soubeiga, "A hyperheuristic approach to scheduling a sales summit," in *Springer Practice and Theory of Automated Timetabling III*, 2001, pp. 176–190.

[79] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, "Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing," *Elsevier Computers & Electrical Engineering*, vol. 42, pp. 74 – 89, 2015.

[80] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Elsevier Future Generation Computer Systems*, vol. 28, pp. 755 – 768, 2012.

[81] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proceedings of the Conference on Power Aware Computing and Systems (HotPower)*. USENIX Association, 2008.

[82] M. Cardosa, M. R. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in *IFIP/IEEE International Symposium on Integrated Network Management*, 2009, pp. 327–334.

[83] D. Meisner, C. M. Sadler, L. A. Barroso, W. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *ACM/IEEE Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, 2011, pp. 319–330.

[84] E. Ahvar, S. Ahvar, Z. . Mann, N. Crespi, J. Garcia-Alfaro, and R. Glitho, "Cacev: A cost and carbon emission-efficient virtual machine placement method for green distributed clouds," in *IEEE International Conference on Services Computing (SCC)*, June 2016, pp. 275–282.

[85] W. Lin, S. Xu, J. Li, L. Xu, and Z. Peng, "Design and theoretical analysis of virtual machine placement algorithm based on peak workload characteristics," *Springer Soft Computing*, vol. 21, no. 5, pp. 1301–1314, Mar 2017.

[86] M. Chen, H. Zhang, Y. Y. Su, X. Wang, G. Jiang, and K. Yoshihira, "Effective vm sizing in virtualized data centers," in *IFIP/IEEE International Symposium on Integrated Network Management and Workshops*, 2011, pp. 594–601.

[87] X. Ruan and H. Chen, "Performance-to-power ratio aware virtual machine (vm) allocation in energy-efficient clouds," in *IEEE International Conference on Cluster Computing (CLUSTER)*, 2015, pp. 264–273.

[88] J. V. Wang, K. Y. Fok, C. T. Cheng, and C. K. Tse, "A stable matching-based virtual machine allocation mechanism for cloud data centers," in *IEEE World Congress on Services (SERVICES)*, 2016, pp. 103–106.

[89] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, and H. Tenhunen, "Multi-agent based architecture for dynamic vm consolidation in cloud data centers," in *EUROMICRO Conference on Software Engineering and Advanced Applications*, Aug 2014, pp. 111–118.

[90] S. S. Masoumzadeh and H. Hlavacs, "A cooperative multi agent learning approach to manage physical host nodes for dynamic consolidation of virtual machines," in *IEEE Symposium on Network Cloud Computing and Applications (NCCA)*, June 2015, pp. 43–50.

[91] ——, "Integrating vm selection criteria in distributed dynamic vm consolidation using fuzzy q-learning," in *Proceedings of the International Conference on Network and Service Management (CNSM)*, Oct 2013, pp. 332–338.

[92] V. Ravi and H. S. Hamead, "Reinforcement learning based service provisioning for a greener cloud," in *International Conference on Eco-friendly Computing and Communication Systems (ICECCS)*, Dec 2014, pp. 85–90.

[93] L. Chen and H. Shen, "Consolidating complementary vms with spatial/temporal-awareness in cloud datacenters," in *IEEE Conference on Computer Communications*, 2014, pp. 1033–1041.

[94] A. Pahlavan, M. Momtazpour, and M. Goudarzi, "Variation-aware server placement and task assignment for data center power minimization," in *IEEE International Symposium on Parallel and Distributed Processing with Applications*, July 2012, pp. 158–165.

[95] ——, "Data center power reduction by heuristic variation-aware server placement and chassis consolidation," in *International Symposium on Computer Architecture and Digital Systems (CADS)*, May 2012, pp. 150–155.

[96] J. C. Salinas-Hilburg, M. Zapater, J. L. Risco-Martín, J. M. Moya, and J. L. Ayala, "Unsupervised power modeling of co-allocated workloads for energy efficiency in data centers," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2016, pp. 1345–1350.

[97] M. Zapater, O. Tuncer, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun, "Leakage-aware cooling management for improving server energy efficiency," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 26, no. 10, pp. 2764–2777, Oct 2015.

[98] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," in *International Conference on Autonomic Computing*, June 2008, pp. 3–12.

[99] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, "Virtual machine consolidation for datacenter energy improvement," *CoRR, Distributed, Parallel, and Cluster Computing*, 2013.

[100] J. Kim, M. Ruggiero, and D. Atienza, "Free cooling-aware dynamic power management for green datacenters," in *International Conference on High Performance Computing and Simulation (HPCS)*, 2012, pp. 140–146.

[101] S. Shen, V. v. Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters," in *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2015, pp. 465–474.

[102] D. S. Dias and L. H. M. K. Costa, "Online traffic-aware virtual machine placement in data center networks," in *Global Information Infrastructure and Networking Symposium (GIIS)*, Dec 2012, pp. 1–8.

[103] M. H. Ferdaus, M. Murshed, R. N. Calheiros, and R. Buyya, *Network-aware virtual machine placement and migration in cloud data centers*, ser. Bagchi S (ed) Emerging research in cloud distributed computing systems, Chap 2. Information Science Reference, Hershey PA, 2015.

[104] A. Pahlevan, P. G. D. Valle, and D. Atienza, "Exploiting cpu-load and data correlations in multi-objective vm placement for geo-distributed data centers," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2016, pp. 1333–1338.

[105] L. Costero, A. Iranfar, M. Zapater, F. D. Igual, K. Olcoz, and D. Atienza, "Multi-agent reinforcement learning for efficient real-time multi-user video transcoding," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019.

[106] I. Mesecan and I. Ö. Bucak, "Searching the effects of image scaling for underground object detection using kmeans and knn," in *European Modelling Symposium (EMS)*, 2014, pp. 180–184.

[107] A. Iranfar, M. Zapater, and D. Atienza, "Machine learning-based quality-aware power and thermal management of multistream hevc encoding on multicore servers," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 29, no. 10, pp. 2268–2281, 2018.

[108] R. E. Rosenthal, "Gams-a user's guide," in *GAMS Development Corporation*, 2016.

[109] Intel, "Intel xeon processor e5 v4 product family datasheet, volume one: Electrical," https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/xeon-e5-v4-datasheet-vol-1.pdf, 2016.

[110] I. Narayanan, A. Kansal, and A. Sivasubramaniam, "Right-sizing geo-distributed data centers for availability and latency," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 230–240.

[111] D. Fooladivanda, C. Rosenberg, and S. Garg, "An analysis of energy storage and regulation," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2014, pp. 91–96.

[112] Y. Kim, V. Raghunathan, and A. Raghunathan, "Design and management of battery-supercapacitor hybrid electrical energy storage systems for regulation services," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 1, pp. 12–24, Jan 2017.

[113] H. Chen, Z. Liu, A. K. Coskun, and A. Wierman, "Optimizing energy storage participation in emerging power markets," in *International Green and Sustainable Computing Conference (IGSC)*, 2015, pp. 1–6.

[114] V. S. K. M. Balijepalli, V. Pradhan, S. A. Khaparde, and R. M. Shereef, "Review of demand response under smart grid paradigm," in *IEEE PES Innovative Smart Grid Technologies*, Dec 2011, pp. 236–243.

[115] G. Carpinelli, G. Celli, S. Mocci, F. Mottola, F. Pilo, and D. Proto, "Optimal integration of distributed energy storage devices in smart grids," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 985–995, 2013.

[116] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.

[117] Y. Wang, X. Lin, Y. Kim, Q. Xie, M. Pedram, and N. Chang, "Single-source, single-destination charge migration in hybrid electrical energy storage systems," *IEEE Transactions on Very Large Scale Integration Systems (T-VLSI)*, vol. 22, no. 12, pp. 2752–2765, 2014.

[118] N. Mukherjee and D. Strickland, "Control of cascaded dc-dc converter-based hybrid battery energy storage systems - part i: Stability issue," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2340–2349, 2016.

[119] L. Y. Wang, C. Wang, G. Yin, F. Lin, M. P. Polis, C. Zhang, and J. Jiang, "Balanced control strategies for interconnected heterogeneous battery systems," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 189–199, 2016.

[120] M. Rossi, A. Toppano, and D. Brunelli, "Real-time optimization of the battery banks lifetime in hybrid residential electrical systems," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2014, pp. 1–6.

[121] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," in *Proceedings of the USENIX Annual Technical Conference*. USENIX Association, 2007, pp. 29:1–29:6.

[122] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, , and C. Kozyrakis, "Power management of datacenter workloads using per-core power gating," *Computer Architecture Letter*, vol. 8, no. 2, pp. 48–51, September 2009.

[123] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 19, no. 11, pp. 1458–1472, Nov 2008.

[124] Y. Chen, D. Gmach, C. Hyser, Z. Wang, C. Bash, C. Hoover, and S. Singhal, "Integrated management of application performance, power and cooling in data centers," in *IEEE Network Operations and Management Symposium (NOMS)*, April 2010, pp. 615–622.

[125] L. Parolini, B. Sinopoli, B. H. Krogh, and W. Zhikui, "A cyber-physical systems approach to data center modeling and control for energy efficiency," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 254–268, Jan 2012.

[126] "Operating practices, procedures, and tools," *North American Electrical Reliability Corporation*, 2011, http://www.nerc.com/docs/pc/ivgtf/IVGTF2-4.pdf.

[127] "Ancillary services and balancing authority area solutions to integrate variable generation," *North American Electrical Reliability Corporation*, 2011, http://www.nerc.com/docs/pc/ivgtf/IVGTF2-3_Ancillary_Service.pdf.

[128] B. Kranz, R. Pike, and E. Hirst, "Integrated electricity markets in new york," *The Electricity Journal*, vol. 16, no. 2, pp. 54 – 65, 2003.

[129] PJM, "Description of regulation signals," www.pjm.com, 2014.

[130] ——, "Pjm manual 12: Balancing operations," www.pjm.com, 2018.

[131] M. Ghasemi-Gol, Y. Wang, and M. Pedram, "An optimization framework for data centers to minimize electric bill under day-ahead dynamic energy prices while providing regulation services," in *International Green Computing Conference (IGCC)*, Nov 2014, pp. 1–9.

[132] B. Aksanli and T. Rosing, "Providing regulation services and managing data center peak power budgets," in *Design Automation Test in Europe Conference Exhibition (DATE)*, 2014, pp. 1–4.

[133] Z. Liu, I. Liu, S. Low, and A. Wierman, "Pricing data center demand response," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 111–123, 2014.

[134] H. Dou, Y. Qi, W. Wei, and H. Song, "Carbon-aware electricity cost minimization for sustainable data centers," *IEEE Transactions on Sustainable Computing (T-SUSC)*, vol. 2, no. 2, pp. 211–223, 2017.

[135] N. Deng, C. Stewart, and J. Li, "Concentrating renewable energy in grid-tied datacenters," in *Proceedings of the IEEE International Symposium on Sustainable Systems and Technology (ISSST)*, May 2011, pp. 1–6.

[136] Y. Wang, Y. Kim, Q. Xie, N. Chang, and M. Pedram, "Charge migration efficiency optimization in hybrid electrical energy storage (hees) systems," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Aug 2011, pp. 103–108.

[137] Y. Riffonneau, S. Bacha, F. Barruel, and A. Delaille, "Energy flow management in grid connected pv systems with storage - a deterministic approach," in *IEEE International Conference on Industrial Technology*, Feb 2009, pp. 1–6.

[138] Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, "Optimal power flow management for grid connected pv systems with batteries," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 3, pp. 309–320, 2011.

[139] I. Alsaidan, A. Khodaei, and W. Gao, "Determination of optimal size and depth of discharge for battery energy storage in standalone microgrids," in *North American Power Symposium (NAPS)*, Sep. 2016, pp. 1–6.

[140] VARTA professional dual power as lead-acid battery, http://www.varta-automotive.com/en-gb/products/industrial/industrial-professional-dual-purpose.

[141] StarkPower UltraEnergy as lithium-ion battery, http://www.starkpower.com/spnews/energystoragebatt.

[142] PV device characteristics, http://www.enfsolar.com/pv/cell-datasheet/429.

[143] Real sun irradiance, http://www.soda-is.com/eng/services/services_radiation_free_eng.php.

[144] Temperature profiles, http://www.tutiempo.net/en/Climate.

[145] P. Q. . E. D. P. Eaton Electrical, "Energy storage and pv architecture design," in *GreenDataNet (Seventh Framework Program)*, 2015.

[146] T. Cui, S. Chen, Y. Wang, S. N. Massoud, and Pedram, "Optimal control of pevs with a charging aggregator considering regulation service provisioning," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 4, pp. 23:1–23:23, Aug. 2017.

[147] Converter efficiency, http://www.schaeferpower.de/cms/en/produkte.html.

[148] Y. Wang, X. Lin, M. Pedram, S. Park, and N. Chang, "Optimal control of a grid-connected hybrid electrical energy storage system for homes," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2013, pp. 881–886.

[149] M. Pedram and I. Hwang, "Power and performance modeling in a virtualized server system," in *Proceedings of the IEEE International Conference on Parallel Processing Workshops (ICPPW)*, 2010, pp. 520–526.

[150] T. Benson, A. Anand, A. A. M., and Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 92–99, Jan. 2010.

[151] Electricity market price tariff, http://www.strompreis.elcom.admin.ch/PriceDetail.aspx?placeNumber=261&OpID=565&-Period=2014&CatID=12.

[152] X. Zhao, V. Vusirikala, B. Koley, V. Kamalov, and T. Hofmeister, "The prospect of inter-data-center optical networks," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 32–38, Sep. 2013.

[153] A. Sadasivarao, S. Syed, P. Pan, C. Liou, I. Monga, C. Guok, and A. Lake, "Bursting data between data centers: Case for transport sdn," in *IEEE Annual Symposium on High-Performance Interconnects*, Aug 2013, pp. 87–90.

[154] H. Rodrigues, I. Monga, A. Sadasivarao, S. Syed, C. Guok, E. Pouyoul, C. Liou, and T. Rosing, "Traffic optimization in multi-layered wans using sdn," in *IEEE Annual Symposium on High-Performance Interconnects*, Aug 2014, pp. 71–78.

[155] C. P. Guok, D. W. Robertson, E. Chaniotakis, M. R. Thompson, W. Johnston, and B. Tierney, "A user driven dynamic circuit network implementation," in *IEEE Globecom Workshops*, Nov 2008, pp. 1–5.

[156] B. Aksanli, T. S. Rosing, and I. Monga, "Benefits of green energy and proportionality in high speed wide area networks connecting data centers," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2012, pp. 175–180.

[157] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *NETWORKING, Springer Berlin Heidelberg*, 2009, pp. 795–808.

[158] A. K. Somani and T. Wu, "Monitoring and detecting attacks in all-optical networks," in *Information Assurance*, ser. The Morgan Kaufmann Series in Networking. Morgan Kaufmann, 2008, pp. 307 – 347.

[159] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis, Forecasting and Control*. Forth Edition, Hoboken, NJ, USA: Wiley, 2013.

[160] H. Chen, Y. Zhang, M. Caramanis, and A. K. Coskun, "EnergyQARE: Qos-aware data center participation in smart grid regulation service reserve provision," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2018.

[161] C. Delimitro and C. Kozyrakis, "Optimizing resource provisioning in shared cloud systems," Standford University, Tech. Rep., 2014.

[162] Matlab fmincon function, http://www.mathworks.com/help/optim/ug/fmincon.html.

[163] J. Wilkes, "More google cluster data," *Google research blog, November*, 2011.

[164] J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Oakland, CA, Tech. Rep., 2011.

[165] P. Lotfi-Kamran, B. Grot, M. Ferdman, S. Volos, O. Kocberber, J. Picorel, A. Adileh, D. Jevdjic, S. Idgunji, E. Ozer, and B. Falsafi, "Scale-out processors," in *ACM/IEEE Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, 2012, pp. 500–511.

[166] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[167] E. Alon, K. Asanovic, J. Bachrach, J. Demmel, A. Fox, K. Keutzer, B. Nikolic, D. Patterson, K. Sen, and J. Wawrzynek, "Algorithms and specializers for provably optimal implementations with resilience and efficiency," *Research Projects, UC Berkeley*, https://www2.eecs.berkeley.edu/bears/2013/Presentations/asanovic.pdf.

[168] S. Borkar, "Thousand core chips: a technology perspective," in *ACM Proceedings of the annual Design Automation Conference (DAC)*, 2007, pp. 746–749.

[169] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May 2012.

[170] A. M. Fard, M. Ghasemi, and M. Kargahi, "Response-time minimization in soft real-time systems with temperature-affected reliability constraint," in *CSI Symposium on Real-Time and Embedded Systems and Technologies (RTEST)*, Oct 2015, pp. 1–8.

[171] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "Impact: Imprecise adders for low-power approximate computing," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, Aug 2011, pp. 409–414.

[172] K. Swaminathan, E. Kultursay, V. Saripalli, V. Narayanan, M. T. Kandemir, and S. Datta, "Steep-slope devices: From dark to dim silicon," *IEEE Micro*, vol. 33, no. 5, pp. 50–59, Sep. 2013.

[173] M. Lyons, M. Hempstead, G. Wei, and D. Brooks, "The accelerator store framework for high-performance, low-power accelerator-based systems," *IEEE Computer Architecture Letters*, vol. 9, no. 2, pp. 53–56, Feb 2010.

[174] J. Cong and B. Xiao, "Optimization of interconnects between accelerators and shared memories in dark silicon," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2013, pp. 630–637.

[175] D. Rossi, A. Pullini, I. Loi, M. Gautschi, F. K. Gurkaynak, A. Bartolini, P. Flatresse, and L. Benini, "A 60 GOPS/W, -1.8 v to 0.9 v body bias ULP cluster in 28 nm UTBB fd-soi technology," *Solid-State Electronics*, vol. 117, pp. 170 – 184, 2016.

[176] N. Planes, O. Weber, V. Barral, S. Haendler, D. Noblet, D. Croain, M. Bocat, P. Sassoulas, X. Federspiel, A. Cros, A. Bajolet, E. Richard, B. Dumont, P. Perreau, D. Petit, D. Golanski, C. Fenouillet-Béranger, N. Guillot, M. Rafik, V. Huard, S. Puget, X. Montagner, M. Jaud, O. Rozeau, O. Saxod, F. Wacquant, F. Monsieur, D. Barge, L. Pinzelli, M. Mellier, F. Boeuf, F. Arnaud, and M. Haond, "28nm fdsoi technology platform for high-speed low-voltage digital applications," in *IEEE Symposium on VLSI Technology (VLSIT)*, June 2012, pp. 133–134.

[177] U. R. Karpuzcu, A. Sinkar, N. S. Kim, and J. Torrellas, "Energysmart: Toward energy-efficient manycores for near-threshold computing," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2013, pp. 542–553.

[178] C. Kim, D. Burger, and S. W. Keckler, "An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches," in *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Oct 2002.

[179] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *Annual International Symposium on Computer Architecture (ISCA)*, June 2007.

[180] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Kilo-noc: A heterogeneous network-on-chip architecture for scalability and service guarantees," in *Annual International Symposium on Computer Architecture (ISCA)*, June 2011, pp. 401–412.

[181] M. Ekman and P. Stenstrom, "Performance and power impact of issue-width in chip-multiprocessor cores," in *International Conference on Parallel Processing*, Oct 2003, pp. 359–368.

[182] N. P. Jouppi and S. J. E. Wilton, "Tradeoffs in two-level on-chip caching," in *Annual International Symposium on Computer Architecture (ISCA)*, April 1994, pp. 34–45.

[183] R. Iyer, S. Makineni, J. Moses, R. Illikkal, and D. Newell, "Performance , area and bandwidth implications on large-scale cmp cache design," in *Proceedings of the Workshop on Chip Multiprocessor Memory Systems and Interconnects*, Feb 2007.

[184] L. Gwennap, "Thunderx rattles server market," *Microprocessor Report*, vol. 29, no. 6, pp. 1–4, 2014.

[185] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *ACM SIGOPS Operating Systems Review (SOSP)*, vol. 37, no. 5, 2003, pp. 164–177.

[186] J. Xu and J. A. B. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *IEEE/ACM International Conference on Green Computing and Communications International Conference on Cyber, Physical and Social Computing*, Dec 2010, pp. 179–188.

[187] G. V. Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *IEEE International Conference on Cluster Computing and Workshops*, Aug 2009, pp. 1–10.

[188] M. Hadji and D. Zeghlache, "Minimum cost maximum flow algorithm for dynamic resource allocation in clouds," in *IEEE International Conference on Cloud Computing*, June 2012, pp. 876–882.

[189] H. N. Van, F. D. Tran, and J. Menaud, "Performance and power management for cloud infrastructures," in *IEEE International Conference on Cloud Computing*, July 2010, pp. 329–336.

[190] S. K. Garg, A. N. Toosi, S. K. Gopalaiyengar, and R. Buyya, "Sla-based virtual machine management for heterogeneous workloads in a cloud datacenter," *J. Netw. Comput. Appl.*, vol. 45, no. C, pp. 108–120, Oct. 2014.

[191] X. Lin, Y. Xue, P. Bogdan, Y. Wang, S. Garg, and M. Pedram, "Power-aware virtual machine mapping in the data-center-on-a-chip paradigm," in *IEEE International Conference on Computer Design (ICCD)*, Oct 2016, pp. 241–248.

[192] R. Srinivasan and T. Ragheb, "Body-bias scaling for globalfoundries 22fdx technology: New dimension to explore the design," in *GLOBALFOUNDRIES, Synopsys Users Group (SNUG) Silicon Valley*, March 2016.

[193] M. Ferdman, A. Adileh, Y. O. Koçberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "A case for specialized processors for scale-out workloads," *IEEE Micro*, vol. 34, no. 3, pp. 31–42, 2014.

[194] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2009, pp. 469–480.

[195] Micron, "4gb: x4, x8, x16 ddr4 sdram features," https://www.micron.com/\/media/documents/products/data-sheet/dram/ddr4/4gb_ddr4_sdram.pdf,2014.

[196] D. Bortolotti, S. Tinti, P. Altoé, and A. Bartolini, "User-space apis for dynamic power management in many-core armv8 computing nodes," in *International Conference on High Performance Computing Simulation (HPCS)*, July 2016, pp. 675–681.

[197] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe, "Simflex: Statistical sampling of computer system simulation," *IEEE Micro*, vol. 26, no. 4, pp. 18–31, 2006.

[198] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "Dramsim2: A cycle accurate memory system simulator," *IEEE Computer Architecture Letters*, vol. 10, no. 1, pp. 16–19, Jan 2011.

[199] R. E. Wunderlich, T. F. Wenisch, B. Falsafi, and J. C. Hoe, "Smarts: accelerating microarchitecture simulation via rigorous statistical sampling," in *ACM/IEEE Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, June 2003, pp. 84–95.

[200] A. Iranfar, M. Kamal, A. Afzali-Kusha, M. Pedram, and D. Atienza, "Thespot: Thermal stress-aware power and temperature management for multiprocessor systems-on-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1532–1545, 2018.

[201] L. A. Barroso, J. Clidaras, and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*, ser. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2013, vol. 8(3).

[202] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, Dec 2007.

[203] A. Seuret, A. Iranfar, M. Zapater, J. Thome, and D. Atienza, "Design of a two-phase gravity-driven micro-scale thermosyphon cooling system for high-performance computing data centers," in *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, 2018, pp. 587–595.

[204] A. Iranfar, A. Pahlevan, M. Zapater, and D. Atienza, "Enhancing two-phase cooling efficiency through thermal-aware workload mapping for power-hungry servers," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019.

[205] K. T. Malladi, F. A. Nothaft, K. Periyathambi, B. C. Lee, C. Kozyrakis, and M. Horowitz, "Towards energy-proportional datacenter memory with mobile dram," in *ACM/IEEE Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, June 2012, pp. 37–48.

[206] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and

D. A. Wood, "The gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[207] A. Pahlevan, X. Qu, M. Zapater, and D. Atienza, "Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1667–1680, Aug 2018.

[208] A. Pahlevan, M. Rossi, P. G. D. Valle, D. Brunelli, and D. Atienza, "Joint computing and electric systems optimization for green datacenters," *Handbook of Hardware/Software Codesign, Springer*, pp. 1–21, 2017.

[209] A. Pahlevan, J. Picorel, A. P. Zarandi, D. Rossi, M. Zapater, A. Bartolini, P. G. D. Valle, D. Atienza, L. Benini, and B. Falsafi, "Towards near-threshold server processors," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2016, pp. 7–12.

[210] A. Pahlevan, Y. M. Qureshi, M. Zapater, A. Bartolini, D. Rossi, L. Benini, and D. Atienza, "Energy proportionality in near-threshold computing servers and cloud data centers: Consolidating or not?" in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2018, pp. 147–152.

[211] A. Iranfar, A. Pahlevan, M. Zapater, M. Zagar, M. Kovac, and D. Atienza, "Online efficient bio-medical video transcoding on mpsocs through content-aware workload allocation," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2018, pp. 949–954.

[212] E. D. Giovanni, A. Aminifar, A. Luca, S. Yazdani, J.-M. Vesin, and D. Atienza, "A patient-specific methodology for prediction of paroxysmal atrial fibrillation onset," in *Computing in Cardiology (CinC)*, 2017.

[213] G. Surrel, A. Aminifar, F. Rincon, S. Murali, and D. Atienza, "Online obstructive sleep apnea detection on medical wearable sensors," in *IEEE Transactions on Biomedical Circuits and Systems*, 2018.

[214] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza, "Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems," in *IEEE Transactions on Biomedical Circuits and Systems*, 2018.

[215] F. Forooghifar, A. Aminifar, and D. Atienza, "Self-aware wearable systems in epileptic seizure detection," in *Euromicro Conference on Digital System Design (DSD)*, 2018.

[216] D. Pascual, A. Aminifar, and D. Atienza, "A self-learning methodology for epileptic seizure detection with minimally-supervised edge labeling," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019.

[217] L. Ferretti, G. Ansaloni, L. Pozzi, A. Aminifar, D. Atienza, L. Cammoun, and P. Ryvlin, "Control-quality driven design of cyber-physical systems with robustness guarantees," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019.

[218] V. M. Canovas, F. I. T. Dell'Agnola, A. A. Valdes, A. Aminifar, and D. Atienza, "Multimodal acute stress recognition using off-the-shelf wearable devices," in *International Engineering in Medicine and Biology Conference (EMBC)*, 2019.

[219] R. Mahmud, R. Kotagiri, and R. Buyya, *Fog Computing: A Taxonomy, Survey and Future Directions*. Springer Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives, 2018, pp. 103–130.

[220] P. Zhang, J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, and X. Luo, "A survey on access control in fog computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 144–149, Feb 2018.

Ali **PAHLEVAN**

Cloud Computing | Data Center Efficiency | Machine Learning | Hardware Engineering | Embedded Systems Design

@ ali.pahlevan@epfl.ch
linkedin.com/in/ali-pahlevan
scholar.google.com/ali-pahlevan

Education

Apr. 2019 – Sep. 2014 | **Ph.D. in Electrical Engineering**
Swiss Federal Institutes of Technology (EPFL), Switzerland

Mar. 2013 – Sep. 2010 | **M.Sc. in Computer Engineering, Computer Architecture**
Sharif University of Technology (SUT), Iran

July 2010 – Sep. 2006 | **B.Sc. in Computer Engineering, Hardware Engineering**
Ferdowsi University of Mashhad (FUM), Iran

Professional Experience

Present – Sep. 2014 | **Doctoral Assistant | Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland**
Project : Multi-Objective System-Level Management of Modern Green Data Centers

- Developing multi-objective heuristic and machine learning-based workload allocation methods
- Presenting hyper-heuristic method to integrate the strengths of heuristic and machine learning algorithms
- Energy and cost management of green geo-distributed data centers considering renewable and battery sources
- Data center cost optimization in emerging power markets
- Increasing energy proportionality in Near-Threshold Computing (NTC) servers and cloud data centers w.r.t. FD-SOI technology

Machine Learning | Hyper-Heuristic | Green Data Centers | Energy Efficiency | Renewables | Near-Threshold Computing (NTC) | FD-SOI Technology

Sep. 2018 – Feb. 2016 | **Project Co-Supervision | Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland**

1. Visiting Student Project : Multi-Agent Machine Learning-Based Approach for Energy Efficient Dynamic Consolidation in Data Centers

- Presenting a centralized-distributed low-overhead failure-aware dynamic workload allocation strategy using a multi-agent machine learning algorithm to minimize energy consumption

2. Student Intern Project : Efficient Workload Allocation Method in Data Centers

- Developing multi-objective machine learning-based workload allocation method using value-iteration algorithm (reinforcement learning) to optimize network traffic and energy consumption

Multi-Agent Machine Learning | Reinforcement Learning

Sep. 2014 – Mar. 2014 | **Intern | Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland**

- Developing a multi-level discrete-time framework for green data centers implemented in C++ programming language
- System-level optimization of renewable energy and hybrid energy storage systems (joint lead-acid and lithium-ion technology)

Multi-Level Framework | Lead-Acid and Lithium-Ion Batteries | System-Level Optimization | C++

Mar. 2013 – Sep. 2010 | **Research Assistant | Energy- and Environment-Aware Systems laboratorY (EASY), SUT, Tehran, Iran**

Project : Thermal- and Process Variation-Aware Data Center Energy Reduction

- Investigating process variation (inability to precisely control the transistors or system parameters) effect on data center power reduction techniques
- Proposing different heuristic- and Integer Linear Programming (ILP)-based methods and techniques in system-level power management of data centers
- Optimizing cooling system operation under process variation effect on servers power consumption

Heuristic | Integer Linear Programming (ILP) | Process Variation

Project Experiences

2017 – 2014 | **European Project : GreenDataNet | European Commission FP7, STREP Project**

Green and smart data centres network design

- **Description :** design, validate and demonstrate a system-level optimization solution for urban data centers to improve their energy and performance
- **Output :** demo link on Internet Explorer : http ://gdn.perf-it.eu/webhmi/homeexternal.htm
- In cooperation with Eaton Industries (FR), Nissan International SA (CH), CEA (FR), ICTRoom (NL), Univ. of Trento (IT) and Credit Suisse SA (CH) (supervision of Prof. Atienza)

2017 – 2015 | **Nano-Tera | YINS RTD Project**

Thermal- and Power-Aware Design Approach for Next Generation Data Centers

- In cooperation with EPFL, ETHZ, Eaton, BrainServe and Credit Suisse (supervision of Prof. Atienza)

2017 | **Face Recognition**

- Using Neural Network, SVM, LDA, and K-means

2015 | **Building Climate Control**

- Using a robust Model Predictive Control (MPC)

Publications

Journal Articles and Book Chapter :

- A. Pahlevan, M. Zapater, A. K. Coskun and D. Atienza, "**Electricity Cost Optimization for Green Data Centers in Smart Grid Regulation Service**," *In IEEE Transactions on Sustainable Computing (TSUSC)*, 2019 (Submitted)
- K. Haghshenas, A. Pahlevan, M. Zapater, S. Mohammadi and D. Atienza, "**A Distributed-Centralized Low Overhead Approach for Energy Efficient Resource Management of Cloud Data Centers**," *In IEEE Transactions on Services Computing (TSC)*, 2018 (Preparing the major revision version)
- A. Pahlevan, X. Qu, M. Zapater and D. Atienza, "**Integrating Heuristic and Machine-Learning Methods for Efficient Virtual Machine Allocation in Data Centers**," *In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2017 (Link)
- A. Pahlevan, M. Rossi, P. G. Del Valle, D. Brunelli and D. Atienza, "**Joint Computing and Electric Systems Optimization for Green Datacenters**," *In Handbook of Hardware/ Software Codesign*, Springer, 2017 (Link)
- S. Esfandiarpoor, A. Pahlavan and M. Goudarzi, "**Structure-Aware Online Virtual Machine Consolidation for Datacenter Energy Improvement in Cloud Computing**," *In Computers & Electrical Engineering*, Elsevier, 2015 (Link)
- A. Pahlavan, M. Momtazpour and M. Goudarzi, "**Power Reduction in HPC Data Centers : A Joint Server Placement and Chassis Consolidation Approach**," *In Journal of Supercomputing*, Springer, 2014 (Link)

Conference and Technical Papers

- A. Iranfar, A. Pahlevan, M. Zapater and D. Atienza, "**Enhancing Two-Phase Cooling Efficiency through Thermal-Aware Workload Mapping for Power-Hungry Servers**," *In Design, Automation & Test in Europe Conference Exhibition (DATE)*, Accepted 2019 (Link)
- A. Pahlevan, Y. M. Qureshi, M. Zapater, A. Bartolini, D. Rossi, L. Benini and D. Atienza, "**Energy Proportionality in Near-Threshold Computing Servers and Cloud Data Centers : Consolidating or Not ?**," *In Design, Automation & Test in Europe Conference Exhibition (DATE)*, 2018 (Link)
- A. Iranfar, A. Pahlevan, M. Zapater, M. Zager, M. Kovac and D. Atienza, "**Online Efficient Bio-Medical Video Transcoding on MPSoCs Through Content-Aware Workload Allocation**," *In Design, Automation & Test in Europe Conference Exhibition (DATE)*, 2018 (Link)
- A. Pahlevan, J. Picorel, A. Pourhabibi, D. Rossi, M. Zapater, A. Bartolini, P. G. Del Valle, D. Atienza, L. Benini and B. Falsafi, "**Towards Near-Threshold Server Processors**," *In Design, Automation & Test in Europe Conference Exhibition (DATE)*, 2016 (Link)
- A. Pahlevan, P. G. Del Valle and D. Atienza, "**Exploiting CPU-Load and Data Correlations In Multi-Objective VM Placement for Geo-Distributed Data Centers**," *In Design, Automation & Test in Europe Conference Exhibition (DATE)*, 2016 (Link)
- A. Pahlavan, M. Momtazpour and M. Goudarzi, "**Variation-Aware Server Placement and Task Assignment for Data Center Power Minimization**," *In IEEE Int'l Symp. On Parallel and Distributed Processing with Applications (ISPA)*, 2012 (Link)
- A. Pahlavan, M. Momtazpour and M. Goudarzi, "**Data Center Power Reduction By Heuristic Variation-Aware Server Placement and Chassis Consolidation**," *In IEEE CSI Int'l Symp. on Computer Architecture and Digital Systems (CADS)*, 2012 (Link)
- A. Pahlevan, M. Zapater, P. G. Del Valle and D. Atienza, "**Multi-Objective System-Level Management of Geo-Distributed Data Centers**," *In YINS project, nano-tera & EcoCloud*, 2016 and 2017
- S. Esfandiarpoor, A. Pahlavan and M. Goudarzi, "**Virtual Machine Consolidation for Datacenter Energy Improvement**," *In Distributed, Parallel, and Cluster Computing (cs.DC), arXiv*, 2013 (Link)

Teaching Experiences

2018	Digital Systems Design (B.Sc. Course), EPFL
2017	Digital Systems Design (B.Sc. Course), EPFL
2012	Electronic System Level Design (M.Sc. Course), Sharif University of Technology (SUT)
2012	Digital Electronics Laboratory (B.Sc. Course), Sharif University of Technology (SUT)
2011	Digital Systems Design (B.Sc. Course), Sharif University of Technology (SUT)

Honors and Awards

2014	Awarded internship grant from Embedded Systems Laboratory (ESL) at EPFL, Lausanne, Switzerland
2013	Honorary admitted as a distinguished student in Ph.D. program, department of computer engineering, Sharif University of Technology (SUT), Tehran, Iran
2012	Ranked 1^{st} among M.Sc. students in computer engineering
2010	Ranked 23^{th} in national university students olympiad in computer engineering, Tehran, Iran
2010	Ranked 58^{th} in national university entrance exam for M.Sc. degree in computer Engineering among about 20,000 participants
2010	Ranked 1^{st} among B.Sc. students in computer engineering
2007-2009	Honorable mention in ACM Asia programming contest, Tehran Site
2007-2009	Ranked 1^{st}, 3^{rd}, and 4^{th} in internal ACM programming contest in Ferdowsi University of Mashhad (FUM), Mashhad, Iran

Management and Organization Experiences

2018 **Tackling The Job Market Successfully Seminar :** assessing the career, writing a results-oriented job application, and prepare ourselves for the job process, EPFL, Lasuanne, Switzerland

2017 A member of organization committee in Design, Automation & Test in Europe Conference Exhibition (DATE), EFPL, Lausanne, Switzerland

Professional Activities

Member IEEE student member

Reviewer IEEE Transactions on Computers (TC), IEEE Transactions on Sustainable Computing (TSUSC), ESWEEK, International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS), IEEE International Conference on Computer Design (ICCD)

Professional Skills

Programming C/C++, Matlab, Python(familiar), Verilog, System Verilog, SystemC, CUDA Programming, Assembly

Tools Proteus, Hspice, Pspice, Modelsim, Synopsys Design Compiler, Gams and Lingo (for linear and non-linear programming), Synopsys DFT Compiler, Quartus II, XPS and ISE Xilinx, Code Composer Studio

Editing Softwares Microsoft Office and LaTeX

**Ce document a été imprimé au Centre d'impression EPFL,
imprimerie climatiquement neutre, certifiée myClimate.**