Power-Thermal Modeling and Control of Energy-Efficient Servers and Datacenters

Jungsoo Kim, Mohamed M. Sabry, Martino Ruggiero and David Atienza

1 Introduction

This continuous growth in demand for computing has resulted in larger collections of servers machines, referred to as clusters or server farms, being hosted in denser datacenters thus having a higher computational and storage capability per occupied unit volume. While projections indicate a continued scaling of server density and manufacturing cost for another decade, the semiconductor manufacturing industry has already renounced following *Dennard scaling*[1] and almost reached the physical limits of voltage scaling in Complementary Metal-Oxide-Semiconductor (CMOS) technologies, which results in an energy-scalability wall that makes transistor power

J. Kim was also affiliated with ESL-EPFL during the period this research was developed.

[1] The scaling theory he and his colleagues formulated in 1974 postulated that MOSFETs continue to function as voltage-controlled switches while all key figures of merit (such as layout density, operating speed, and energy efficiency improve provided geometric dimensions, voltages, and doping concentrations) are consistently scaled to maintain the same electric field. This property underlies the achievement of Moore's Law and the evolution of microelectronics over the last few decades.

J. Kim (✉)
DMC Research Center, Samsung Electronics, Suwon, Republic of Korea
e-mail: jungsoo9.kim@samsung.com

M. M. Sabry · M. Ruggiero · D. Atienza
Embedded Systems Laboratory, EPFL, Lausanne, Switzerland
e-mail: mohamed.sabry@epfl.ch

M. Ruggiero
e-mail: martino.ruggiero@epfl.ch

D. Atienza
e-mail: david.atienza@epfl.ch

S. U. Khan, A. Y. Zomaya (eds.), *Handbook on Data Centers,*
DOI 10.1007/978-1-4939-2092-1_29

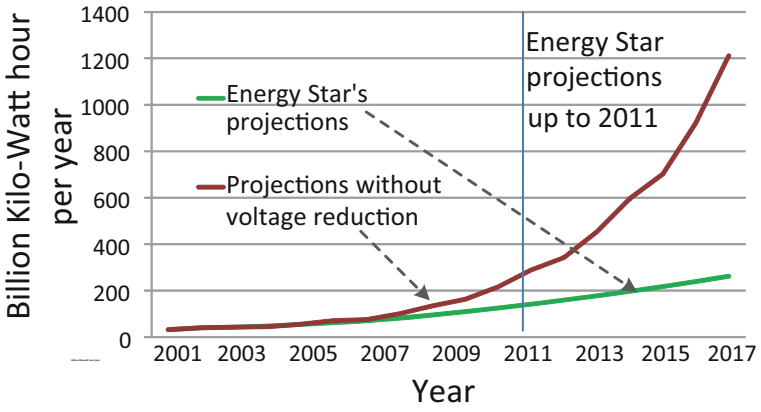

Fig. 1 Datacenters current energy use and projection [2]

consumption increase with further increases in density. At a large-scale, this "economic meltdown trend of Moore's law" for servers and datacenters [1], translates in a dramatic increase in computation and cooling electricity costs.

Energy-efficiency constraints have therefore become the dominant limiting factor for datacenters because their growing size and electrical power demands cannot be met with state-of-the-art design practices and their electricity bill is skyrocketing, as Fig. 1 shows. This figure depicts the *Energy Star*s [2] electricity usage measured and projected up to 2011. If we extrapolate these values linearly up to 2017, as voltages stop scaling down according to the current *International Technology Roadmap for Semiconductors* (ITRS) projections, the electricity use would exponentially increase. Moreover, the expected increase in energy prices would only exacerbate the cost of using datacenters. Thus, datacenter operation will require more money per year on energy costs than on IT equipment replacement. In 2007, datacenters in Western Europe consumed an estimated total of 56 terawatt-hours (TWh) of power per year. The European Union (EU) estimates that this figure is likely to reach 124 TWh by 2020 [2].

Power and thermal monitoring and control play a key role to reduce the power consumption of datacenters while maintaining the performance requirements and the maximum temperature constraints by manipulating multiple control knobs in the systems. As monitoring and control solutions are developed by being tightly coupled with hardware architecture and workload characteristics running on datacenters, we first revisit the datacenter structures (Sect.1.1) and the workload characteristics running on current datacenters (Sect. 1.2). Then, we present an energy efficiency figure of state-of-the-art datacenters (Sect. 1.3), which motivates us to develop effective power and thermal monitoring and control solutions by manipulating multiple control knobs to achieve further global/holistic energy savings in datacenters.

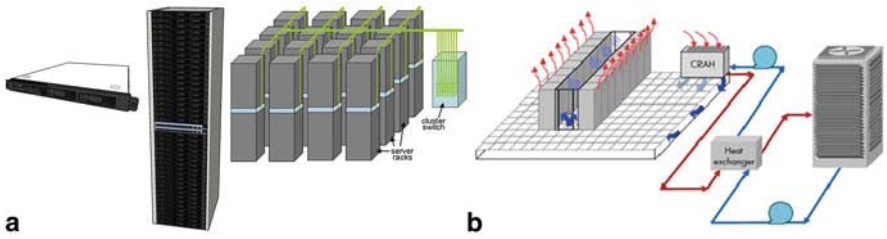

Fig. 2 Organization of datacenters: computing and cooling systems (**a**) and server organization (**b**) [3]

1.1 Overall Datacenter Architecture

A datacenter can largely be decomposed into three parts: (1) IT, i.e., aggregation of servers, (2) cooling, and (3) power distribution units. Servers are the key constituent of datacenters and produce a significant amount of heat as they provide the capability of data manipulation and processing. In a server room, there is a large number of servers to sustain performance requirements. Figure 2a shows an example of typical server organization in a server room with a typical 1 U[2]. Server are typically placed in 42 U racks such that the servers are interconnected with local rack Ethernet switch, and then, connected to cluster-level Ethernet switches, which can potentially span more than ten thousand individual servers [3].

Datacenter cooling systems are deployed to remove heat generated by the servers along with additional amount of heat inside a server room, which needs to be removed as well. Power is delivered to servers through power distribution units (PDUs) and stored in un-interruptible power supply (UPS) systems to cope with power blackout. In this chapter, we focus on IT and cooling parts of datacenters. As shown in Fig. 2b, in a typical datacenter, a cooling system consists of computer room air conditioning/handler (CRAC/CRAH) in a server room and heat exchanger (namely, chiller) and cooling tower outside the server room. CRAC/CRAH provides cold air, such that the air condition of server rooms maintains safe operating temperature and humidity through the exchange of hot air exhausted by servers in the room with cold air (or water) provided from a chiller. According to the *American Society of Heating, Refrigerating and Air-Conditioning* (ASHRAE) 2009 recommendation, it is recommended to maintain the server room air condition as follows:

- Temperature: 64.4–80.6 °F
- Humidity: 41.9 °F at dew point (DP) to 60 % RH and 59 °F DP.

However, these values are quite conservative as they are determined by assuming that servers in a server room are fully utilized, which rarely happens as will be explained

[2] A rack unit, **U** or **RU**, is a unit of measure to describe the height of rack-mount servers placed in 19-in. or a 23-in. rack, where 1U corresponds to 1.75 in. (44.45 mm) high.

in Sect. 1.2. Due to the over-provisioning of cooling capability to server rooms, huge amount of power are now wasted in datacenters, which motivates us to develop an efficient system control solution that adaptively adjusts cooling configurations along with existing power and thermal management solutions developed for servers to achieve further energy savings. The effective control solution is only obtained through accurate-yet-efficient monitoring of power consumption and temperature of multiple points of datacenters, which urges to develop an efficient monitoring system for datacenters.

1.2 Datacenter Workload Characteristics

Many types of applications are running on datacenters, ranging from high-performance computing (HPC) to large-scale services, e.g., web search, streaming service, etc. Recently, due to the big advancements on cloud service providers (e.g., Amazon, Microsoft, Google, etc.), it becomes easier to deploy large-scale services, which leads to the drastic increase on servers hosting large-scale applications. The common characteristics of the large-scale services are that they are unprecedentedly parallel as it uses big chunk of data by splitting into small chunk. Figure 3 illustrates the overall operation which manipulates big chunk of dataset. In [4], Ferdman et al., examined applications running on today's clouds and presented top six most commonly found applications as follows:

- *Data serving*: serving as the backing store for large-scale web applications, e.g., Facebook inbox, Google Earth, etc.
- *MapReduce*: large-scale data analysis by first performing filtering and transformation of the data (namely, *map* procedure) and then aggregate the results (namely, *reduce* procedure)
- *Media streaming*: streaming services by packetizing and transmitting media files ranging from megabytes to gigabytes
- *SAT solver*: large-scale computations for solving complex algorithms, e.g., symbolic execution
- *Web frontend*: web services which schedule independent client requests across a large number of stateless web servers
- *Web search*: web search engines such as those powering Google and Microsoft Bing, which indexes terabytes of data obtained from online sources.

Up to now, most of the control solutions have been developed by targeting HPC workload characteristics. However, the workload characteristics of such large-scale applications are quite different from traditional HPC applications in both macroscopic and microscopic scales [4], which mandates us to develop the control solutions for the large-scale applications.

In a macroscopic scale, the application, first, is user-interactive, thereby, the amount of required computing capacity is highly variable and fast-changing [6] due to the dependence with external factors, i.e., number of clients/queries, etc. The

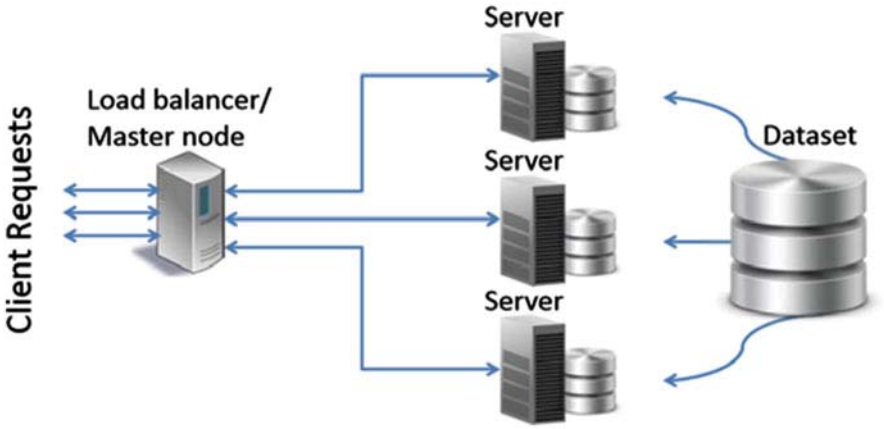

Fig. 3 An example of scale-out applications [5]

characteristics of the workload traffic are well analyzed in [7]. In the coarse-grained time interval (few tens of minutes to hours), the characteristics of users' requests are distinctly different over time while the global pattern has a strong correlation with adjacent time periods as well as the same period in different days. On the other hand, in the fine-grained time interval (less than few seconds), the characteristics of user requests depend on burstiness of traffic and arrival patterns and we can model the characteristics of users' request at the microscopic scale with (1) ON/OFF periods and (2) inter-arrival time between two consecutive requests during ON period. ON period is defined as the longest continual period during which all the request inter-arrival times are smaller than predefined value. Accordingly, OFF period is defined as a period between two on periods. As presented in [7], ON/OFF period and inter-arrival time are time-varying and uncertain while each of them forms lognormal distribution.

Second, the responsiveness (or latency) should come at the first criteria to be satisfied as the level of user satisfaction leads to the success of the business [10]. Third, the amount of required resources is usually far beyond the level that single server can sustain; thereby, massively parallel nodes are cooperatively working by forming a cluster architecture [8]. For instance, in a web search application, a big chunk of search index is divided into multiple smaller datasets, and then, allocated into multiple VMs (or servers) each of which is called a *index searching node (ISN)*. Once a query is arrived, each ISN independently searches matched data with the allocated dataset and a master node gathers the search results from multiple ISNs, then sends the results to clients. Due to the deployment of multiple nodes for a single application, such workload is called *scale-out* applications [4].

Microscopic-scale characteristics of the application are well studied in [4]. The following summarizes the four distinctive micro-architectural workload characteristics in the applications:

- High instruction cache miss rates
- Low instruction- and memory-level parallelism
- Large memory footprint far exceeding the capacity of on-chip caches
- Low on-chip and off-chip bandwidth requirements.

Due to the lack of the control solutions accounting for the distinctive workload characteristics of large-scale cloud application, in this chapter, we will present a power management solution optimized for the workload characteristics of the large-scale cloud applications.

1.3 Energy Efficiency of Datacenters

Due to the conservative cooling provision and lack of the consideration on workload characteristics, vast amount of energy is wasted in todays' datacenter. *Power usage efficiency* (PUE) is the most widely used metric to quantify the power efficiency of datacenters, which is defined as follows:

$$PUE = \frac{\text{Total power consumed by a datacenter}}{\text{Power consumed by servers}} \tag{1}$$

Thus, the lower, the better and it can ideally be reached to 1.0 According to *US Environmental Protection Agency* (EPA) report [2], the PUE of average datacenters around world amounts to 1.9, which means that for every watt of power consumed in the computing equipment, an additional 0.9 W of power is needed for cooling and power delivery. Figure 4a shows the breakdown of energy usage of typical datacenters (The PUE value amounts to $1/0.45 = 2.22$) when assuming $10 \sim 30\,\%$ IT load scenario [3]. Cooling system, comprised with chiller and CRAC/CRAH, consumes around 30 % of energy consumption while the power system spends additional 23 % of energy caused by uninterruptible power supply (UPS), power distribution unit (PDU) and AC-DC conversion losses. Other facility elements, e.g., humidifier, lighting, transformers, contribute around 2 % of total energy consumption. Such inefficiency corresponds to waste of money in the business sense. Figure 4b shows the monthly costs breakdown in a state-of-the-art datacenter assuming a 3-year server amortization and a 15-year infrastructure amortization [9]. This figure illustrates that, in less than three years, the accumulated cooling costs are higher than the actual server deployment costs, thus datacenters energy and thermal management is directly related to effective cooling and power delivery.

Among the various reasons contributing to the poor energy efficiency (e.g., voltage conversion loss in UPS, excessive cooling provision, etc.), the loss in the datacenter cooling facility caused by the over-provisioned cooling capability takes the most significant portion in the entire loss as it is adjusted to guarantee safe operating conditions of servers targeting the worst-case workload scenario which happens rarely. In order to improve the energy ineffectiveness, datacenter designers and a large set of recent search works in the literature have identified three key guidelines as follows:

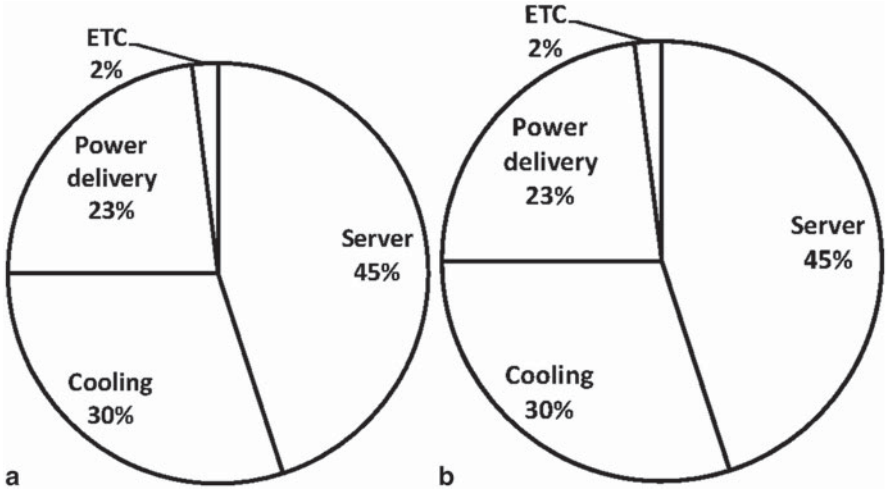

Fig. 4 **a** Breakdown of datacenter energy overheads [3]. **b** Datacenter costs breakdown assuming a 3-year servers and 15-year infrastructure amortization model [9]

- Fine-grained monitoring of PUE
- Server rack layout minimizing hot and cold air mixing by cold-aisle/hot-aisle layout, containment, duct, and analysis of computational fluid dynamics (CFD)
- Adjustment of thermostat of server room to the highest level where servers can be safely operated

However, there still exist huge gap until it reaches to its ideal value, i.e., 1.0, which necessitates the energy- and thermal-aware design in unprecedented ways. The main reason is that all these practices are still focused only on worst-case cooling scenarios designs without any holistic view that considers the dynamic cooling needs of the computing infrastructure at run-time. These results pose very drastic consequences in the design and modes of operation for next-generation datacenters.

1.4 Chapter Organization

In this chapter, we focus on presenting solutions to reduce the energy consumptions of servers and cooling systems through effective power and thermal control solutions based on accurate yet efficient power and temperature modeling and monitoring solutions. The rest of the chapter is organized as follows. Section 2 reviews state-of-the-art datacenters, especially focused on computing and cooling parts of datacenters to understand state-of-the-art technologies and figure out control knobs which are manipulated in control solutions. Section 3 shows approaches of modeling and monitoring power and temperature in servers as well as datacenters. Section 4 explains dynamic power and thermal management solutions for single servers, ranging from

Table 1 Server power breakdown [3]

Component	**Proportion** (%)
CPU	33
DRAM	30
Disk	10
Networking	5
Etc.	22

conventional air-convection cooled servers to liquid cooled ones. Section 5 explains power and thermal management solutions for large-scale computing server clusters in a datacenter. Section 6 explains the joint power and thermal management solutions for large-scale datacenters including both of computing and cooling power consumptions, especially targeting a hybrid cooling architecture which selectively uses free cooling according to required cooling capability. Section 7 summarizes the chapters.

2 State-of-the-Art in Datacenter Design

In this section, we explain state-of-the-art techniques to improve the energy efficiency of datacenters while meeting the temperature constraint, especially focusing on the two biggest energy consumers in datacenters, i.e., computing servers and datacenter cooling facility.

2.1 Computing Servers

1) Energy-Proportional Server Designs Server architectures have traditionally target performance optimization to support the ever-increasingly IT services demands and energy-efficiency has only become an important concern in the last five years. Due to the continuous technology scaling-driven performance improvement and the fact that single microprocessor architectures recently reached its performance limits [11], server designs have evolved since 2005 towards multi-cores architectures. A good example of this trend in state-of-the-art server designs is the HP DL980 blade server, which includes eight CPU sockets and each of them can support up to 10 cores [12]. Currently, the power consumed by servers takes more than 50 % of total power consumed by datacenters [3]. Table 1 shows the power breakdown of existing servers, which outlines that the largest portion of total power consumption in servers is taken by the CPU, but also DRAM memories must be considered as important blocks to develop power and thermal management strategies at server level.

In addition, future server designs trends by major server vendors, e.g., Sun Labs-Oracle, IBM, etc., show an evolution towards 3D-stacked technology integration

programs [11], which enables the integration of a larger number of processing cores in very limited chip volumes and can significantly reduce the memory access latency by stacking memory layers on top of processing cores. Furthermore, 3D integration enables easier development of heterogeneous computing architectures because it is possible to integrate multiple memory types (e.g., 3D-stacked DRAM, phase change memory), and storage (e.g., solid-state disk) devices from different manufacturing processes, as in the EuroCloud server project [13]. However, as a side effect, power density is expected to significantly increase in 3D multi-core computing systems (i.e., up to 300 W/cm^3 [14]), which will make extremely difficult to properly dissipate the generated heat with current air-based cooling systems [15]. In particular, if free cooling is used, it will be a must to consider jointly the conception of the cooling and computing architecture.

One of the recent topics in server research is achieving energy-proportional components, which implies that computing systems should consume different amounts of active power according to their actual utilization. Nowadays, although servers are currently optimized to handle high-performance computation demands, most of the servers in a datacenter run at or below 40 % utilization during a significant part of the time, yet still draw almost full power during the process [16]. Therefore, latest server designs include many sensors (e.g., power, temperature, etc.) to accurately detect the current server utilization state [17]. Also, server components (i.e., processor, memory, and disk) now provide various operating states (e.g., active/idle/sleep/dormant) as well as various voltage and frequency (v/f) levels in processor and memory [18]. Therefore, recent works [19, 20] have shown the potential of developing energy proportionality in servers by exploiting the different power states and v/f levels according to the performance demand of local server utilization. Nonetheless, all these approaches focus on power consumption optimization of computing systems, thus they do not formally guarantee an optimal v/f point under thermal-induced power variations or can provide thermal damage prediction.

In order to reduce idle-time (leakage) power consumption, server processors provide nowadays hardware support for virtualization (e.g., AMD-V, Intel VT-x), which is a technique to enable increased physical server utilization by running applications from multiple OS instances in the so-called virtual machines (VMs) [21]. Moreover, on top of the hardware support, several virtualization software frameworks (e.g., Citrixs XenServer, Microsofts Hyper-V, VMWare ESXi, etc) have been recently developed to host multiple VMs with negligible performance degradation. Figure 5a illustrates the server virtualization. Recent improvements in the server virtualization techniques enable to run applications in a virtualized server within acceptable performance loss, i.e., ~ 20 % for running CPU intensive workload [22] compared to running on a native system, while it is known to be degraded further when running memory- and disk-intensive workloads [23].

These various control options described above, i.e., power state, v/f level, VM placement, etc., give us great opportunities to achieve further power savings by fully utilizing the various control options while posing the challenges to develop an efficient control solution at the same time due to the large solution space, which necessitates us to develop an effective yet low-complexity control scheme.

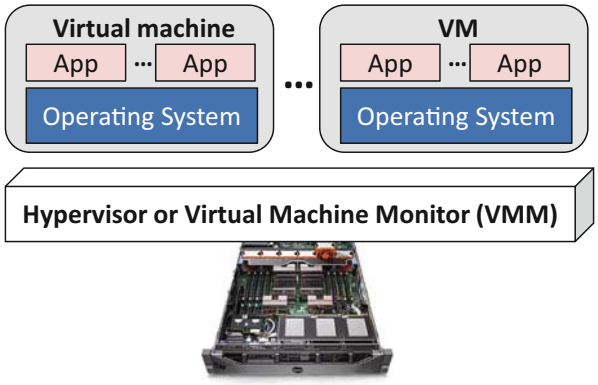

Fig. 5 Concept of server virtualization: hosting multiple VMs with the aid of hypervisor

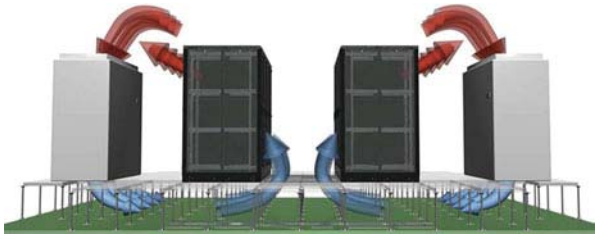

Fig. 6 Hot- and cold-aisle isolation [9]

2.2 *Cooling Infrastructure*

In order to achieve energy-efficient datacenter cooling, various solutions have been presented. In this section, we address the three most widely used and effective solutions: (1) hot- and cold-aisle isolation, (2) closed-coupled cooling, and (3) free cooling. Then, we present how to utilize the cooling solutions more effectively to achieve further energy savings.

1) Hot- and Cold-Aisle Isolation Figure 6 shows a typical way of server room cooling. The cold air is provided by computer room air conditioning (CRAC) units through a raised floor, a steel grid resting on stanchions installed 2–4 ft. above the concrete floor. The cold air flows into racks through perforated tiles, and then, hot air is exhausted through a rear side of rack after absorbing heat generated by servers in the rack. One way of improving cooling efficiency is to prevent mixing the cold air provided from CRAC and hot air exhausted by servers. It is realized by a solution, so called *hot- and cold-aisle isolation*, which arranges server racks such that the intakes of cold air in server racks are faced each other, i.e., cold aisle, while preventing the mixture of hot air in different aisle side, i.e., hot aisle. The hot air is eventually drawn by the CRAC, and then, cold air is again provided to cold aisles by exchanging the heat with cold air (or water) provided from chillers.

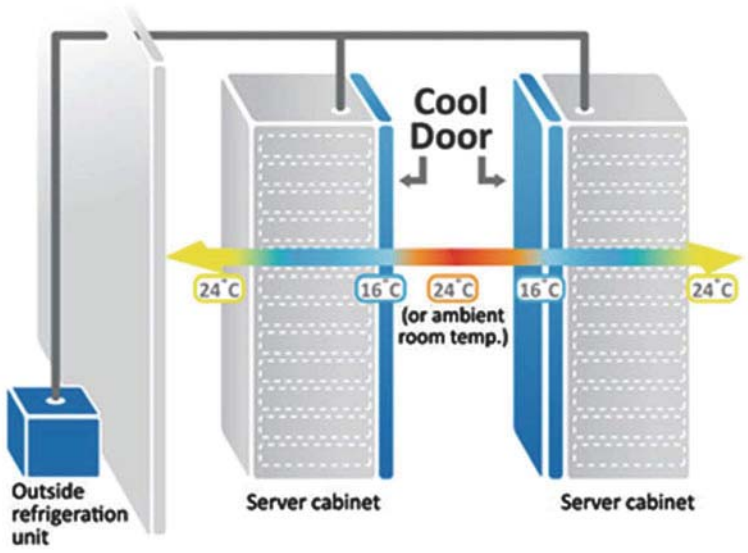

Fig. 7 In-rack cooling [24]

2) Closed-Coupled Cooling Closed-couple cooling solutions place cooling units more closely to computing units so as to remove any losses incurred throughout the delivery of cooling medium and quickly react to spatial temperature distribution. In this cooling solution, there are largely two classifications according to the granularity of computing cluster covered by single cooling unit, i.e., in-row and in-rack coolings. An in-row cooling adjusts cooling condition at every row according to the corresponding conditions while an in-row cooling adapts its cooling configuration according to operating condition at each rack. Figure 7 shows an example of an in-rack cooling solution where the cold air is directly fed into the front door of racks, namely, *CoolDoor* while the hot air is drawn by the CRAC with the same way in Sect. 2.2. The effectiveness of the solution is quite obvious in terms of the energy efficiency in that it can adjust only necessary parts instead of adjusting whole cooling configuration based on the worst-case scenario. It is reported that PUE of this cooling solution can reach down to $1.1 \sim 1.2$ [3]. However, the capital expenditure for the installation is quite high.

3) Free Cooling A recent approach to improve energy efficiency in datacenters is the concept of free cooling, which relies on the use of outside cold air and/or water for cooling instead of electricity. This is a promising architectural innovation for datacenter cooling infrastructure that can enable PUE to approach values near 1.0. Google has recently constructed two datacenters in Ireland and Belgium based on this concept and reports drastically improved PUE figures up to 1.09 [3].

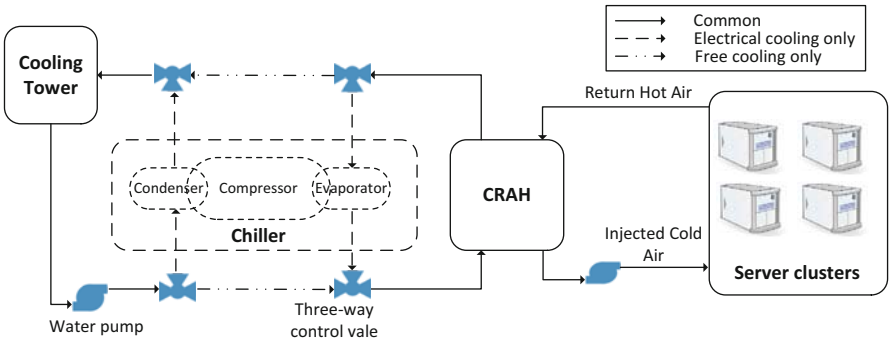

Fig. 8 Datacenter cooling architecture [43]

Despite the promising advantages on cooling-energy efficiency, the fundamental issue of free cooling is its limited applicability, as it can only be used in a very limited set of geographical locations because the cooling capability is tightly coupled with climate condition (e.g., temperature and humidity). Thus, it suffers from wide variations of cooling efficiency during the year, which translates in significantly high computing systems failure rates [25]. Hybrid cooling, which provisions back-up cooling infrastructure along with free cooling, is an intuitive solution to extend the usability of free cooling. Two main types of hybrid cooling architectures exist [26, 27]. The first architecture switches between free- and electricity-based cooling according to the outside temperature: if the outside temperature is lower than a certain threshold, free cooling is used; otherwise, chiller-based electricity cooling is employed as shown in Fig. 8. However, in real-life conditions, datacenters can use free cooling in very limited periods of the year and the average reported PUE is approximately 1.5. The second proposed architecture uses a cooperative hybrid cooling solution to increase the time free cooling is used. In this case, free cooling complements the chiller by pre-cooling hot return water with cold outside water before entering the chiller. This second architecture enables using free cooling, at least partially, for the entire year, and provides up to 50 % energy savings in cooling infrastructure (PUE$\simeq$1.25). However, it still suffers from significant higher failure rate than chiller-based solution due to lack of efficiency in the combined cooling scheme, which makes the current computing systems to operate at higher and variable temperatures. Moreover, due to the continuous increase in server power density, driven by the ever-increasing IT demand, the applicability of current free cooling will be even more limited in the future.

Figure 9 shows the variation of the power consumed by computing and cooling facilities as well as PUE measured for a datacenter equipped with hybrid cooling architecture deployed in Finland. As indicated the PUE line, PUE value varies 1.09 $\sim$ 1.60 and can be largely classified into three periods according to the PUE value. In this datacenter, free cooling is used only when the outside temperature is lower than 8 °C, which is set to very conservative value so as to cope with the worst-case

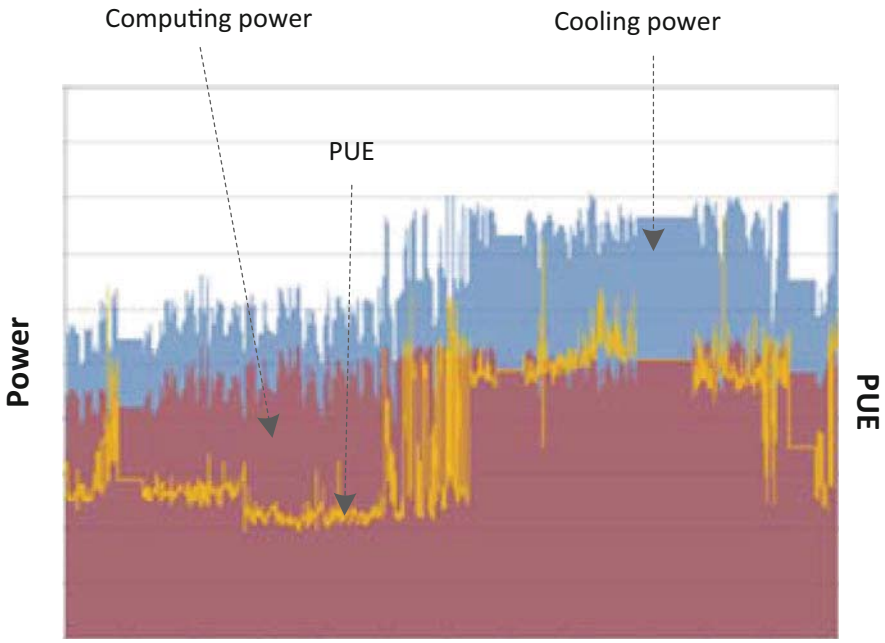

Fig. 9 Variations of power and PUE throughout a year [28] measured by a datacenter equipped with hybrid cooling architecture deployed in Finland

scenario. First, during the winter, PUE value is low as the free cooling is used for the most of the time period while it becomes increased during the summer as the electrical cooling is more frequently used as the temperature goes up.

Thus, free cooling as such cannot provide the ultimate solution to improve datacenter energy efficiency due to the limitation of the cooling capability and the dependency on outside temperature. In order to be generally applicable it must be combined in synergistic ways with innovative energy-proportional server design and cooling solutions, as well as holistic datacenter thermal control.

3 Power and Temperature Modeling and Monitoring

Accurate-yet-efficient modeling and monitoring on power and temperature of datacenters are necessary to develop control solutions for target systems. In this section, we first explain how we can model the power consumption and the temperature of existing servers and cooling facility in datacenters. Then, we address scalable and cost-effective power and temperature monitoring systems for large-scale datacenters.

3.1 Server Modeling

1) Power Modeling A server consists of various components, i.e., CPU, DRAM, disk, network interface (NIC), etc. As presented in Table 1, vast amount of the power is consumed by CPUs, memory, and disk, i.e., more than 70 %. Extensive works have been presented to accurately model power consumption of each component. *McPAT* is micro-architectural power model for chip multiprocessor (CMP), including in-order and out-of-order processor cores, networks-on-chips, shared caches, integrated memory controllers, and multiple-domain clocking, while tacking into account various process characteristics, e.g., bulk CMOS, SOI, and double-gate transistors, based on the forecast in the ITRS roadmap. The accuracy is validated using various processor implementations, i.e., Niagara, Niagara2, Alpha 21364, and Xeon Tulsa, whose errors range 10.84 $\sim$ 22.61 %, compared to the measured values. *DRAMSim* [29] and Micron's *System Power Calculator* [30] provide accurate and detailed timing and power models of various types of DRAM, e.g., DDR, DDR2. DDR3, Mobile LPDRAM, etc., accounting for the operations.

Although such accurate power models exist to model individual component of servers, it is difficult to use all such accurate models together due to the speed of the simulation. It becomes more exacerbated when we target to simulate the large number of servers in datacenters. Thus, high-level power models are widely used to track and estimate the power consumption of servers based on the observation that the power consumption for a given server is highly correlated with distinctive workload characteristics, e.g., CPU-, memory-, or disk-intensive, stressed on servers. To capture the relationship, various works have presented high-level power model which estimates the power consumption based on the utilizations [31–33]. Among them, Economous et al. [31] present a linear regression power model which estimates the server power consumption with respect to utilizations of CPU (u_{cpu}), memory (u_{mem}), and disk (u_{disk}), and network interface (u_{net}) as follows.

$$P_{server} = C_0 + C_1 u_{cpu} + C_2 u_{mem} + C_3 u_{disk} + C_4 u_{net} \tag{2}$$

where $\{C_0, C_1, C_2, C_3\}$ is a set of fitting parameters, which varies according to the target server system. This model is validated through two types of servers: (1) blade servers containing 2.2 GHz AMD Turion processor, 512 MB SDRAM, 40 GB HDD, 10/100 MBit Ethernet and (2) Itanium servers containing four Itanium2 chips, 1 GB DDR, 36 GB HDD, 10/100 MBit Ethernet. According to their evaluations, the errors are within 10 % in most of test cases using various benchmark suites, i.e., SPECcpu200, SPECjbb2000, SPECweb2005. Further evaluations for developing the high-level server power modeling have been conducted in [32] by comparing five different forms of power models as follows:

$$Type1 : P_{server} = C_0 \tag{3}$$

$$Type2 : P_{server} = C_0 + C_1 u_{cpu} \tag{4}$$

$$Type3 : P_{server} = C_0 + C_1 u_{cpu}^r \tag{5}$$

$$Type4 : P_{server} = C_0 + C_1 u_{cpu} + C_2 u_{disk} \tag{6}$$

$$Type5 : P_{server} = C_0 + C_1 u_{cpu} + C_2 u_{mem} + C_3 u_{disk} + C_4 u_{net} \tag{7}$$

Type 1 modes the power consumption in a static value. Type 2 and 3 model the power consumption with respect to CPU utilization, i.e., u_{cpu}, in linear and nonlinear manners, respectively. Type 3 and 4 add additional term to take into account the variations caused by disk (u_{disk}), memory (u_{mem}), and network (u_{net}). It concludes that Type 2 power model is enough for modeling CPU-intensive workload while Type 5 power model, using both of OS-reported component utilizations and CPU performance counters, is needed to cover broad workload characteristics, i.e., memory- and disk-intensive workloads, and aggressively power-managed servers.

In [33], Pedram et al. further enhance the accuracy of the power model by adjusting the fitting parameters according to various operating voltage and frequency and the number of active cores. It used Intel Xeon E5410 processor for the validation with various test cases, i.e., combination of the number of active cores and operating voltage and frequency level. Recently, *Joulemeter* is provided to automatically tune the parameters in power models by measuring battery usage in laptop or measuring power consumption in servers.

Fans also consume significant amount of power in servers. Indeed, it is well known that the fan power consumption has a cubic relationship with fan speed [34], as follows:

$$P_{fan} = C_0 + C_1 s_{fan}^3 \tag{8}$$

where $\{C_0, C_1\}$ is a set of fitting parameters and s_{fan} represents fan speed. Thus, lowering the fan speed enables us to reduce drastic amount of power consumption.

2) Temperature Modeling Accurate temperature models for servers are required to capture the temporal and spatial temperature variations. Especially, due to the high area and cost of placing thermal sensors in a silicon die as well as frequent failures of thermal sensors, the needs for the accurate temperature modeling becomes more important. Computational fluid dynamics (CFD) simulation is known to be a solution to develop accurate and complete 3D thermal map of servers by using numerical methods and algorithms to solve and analyze problems that involve fluid flows. In [35], Choi et al. present a CFD-based thermal modeling solution of servers by solving the governing transport equations shown in the following conservation law form:

$$\frac{\partial \rho \phi}{\partial t} + \frac{\partial \rho U_j \phi}{\phi \partial x_j} = \frac{\partial}{\partial x_j}\left(\Gamma_{phi,eff} \frac{\partial \phi}{\partial x_j}\right) + S_\phi \tag{9}$$

where ϕ is a general variable used for different context, e.g., mass, velocity, temperature, or turbulence properties; ρ is a fluid (air) density; t is a time for transient simulations; x_j is a coordinate x, y, or z direction when j is 1, 2, or 3, U_j is the velocity in each direction; Γ is the diffusion coefficient; S is the source for a particular variable such as the heat flux from a target system when the air temperature is ϕ. The four terms in Eq. (9) corresponds to transient, convection, diffusion, and source

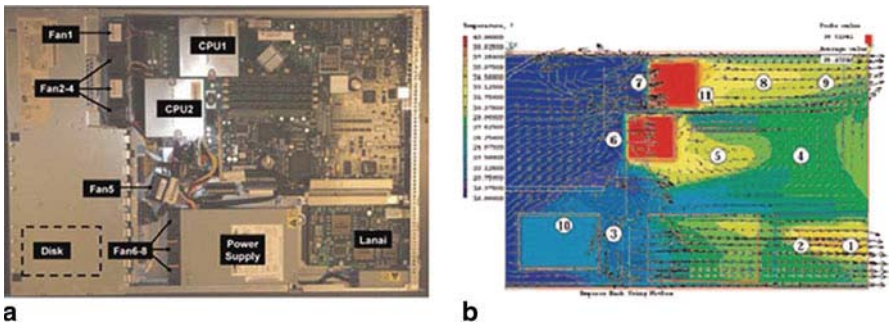

Fig. 10 Layout of IBM X335 server (**a**) and temperature map (**b**) [35]

parts of transport phenomenon at the spatial domain/extent. Figure 10a and b show pictures of IBM X335 server comprising of multiple components and its temperature map, respectively. As shown in Fig. 10, the spatial temperature variation can be accurately modeled. Despite the high accuracy of the CFD simulation, the simulation complexity is quite high because it does not have any closed-form solution for solving the differential equation in Eq. (9), which leads to adopt computer-based numerical procedures.

In [36], T. Heath et al. present a solution of constructing temperature map of servers while relieving the complexity of CFD simulation with negligible accuracy degradation, i.e., within 0.32 °C compared to CFD simulation. The simplification is achieved by abstracting heat- and air-flow with simplified graphs. Recently, a further simplified temperature model for servers has been presented in [38], especially targeting the CPU and memory sub-system of servers considering varied heat removal capability as a fan speed changes. It is developed by constructing thermal RC network of the system based on well-known duality between thermal and electrical phenomena [37], as shown in Fig. 11. In the RC network of CPU socket, P_j^c represents the power consumption of each core in a socket; R_l^c and R_v^c represent the lateral and vertical thermal resistance, respectively, where R_l^c is normally ignored as $R_v^c << R_l^c$; R_s^c and R_{ca}^c are thermal resistance of heat spreader and case-to-ambient (i.e., heat sink), respectively. C_j^c, C_s^c, and C_{ca}^c are thermal capacitances of die, heat spreader, and heat sink, respectively; T_{ja}^c represents the junction temperature which is used as an input to dynamic thermal management (DTM) units such that T_{ja}^c is lower than T_{max}. R_{ca}^c is the sum of the thermal resistances of heat sink and convective resistance, i.e., $R_{ca}^c = R_{hs}^c + R_{conv}^c$, where R_{conv} is changed according to the fan speed as follows:

$$R_{conv}^c \propto \frac{1}{A \cdot s_{fan}^{\alpha}} \tag{10}$$

where A is the effective area and α is a factor with a range of 0.8∼1.0.

In the RC network of memory part, P_{chip}^D is the power consumed in each DRAM chip; R_{chip}^D and C_{chip}^D are thermal resistance and capacitance of each chip; T_j^D is

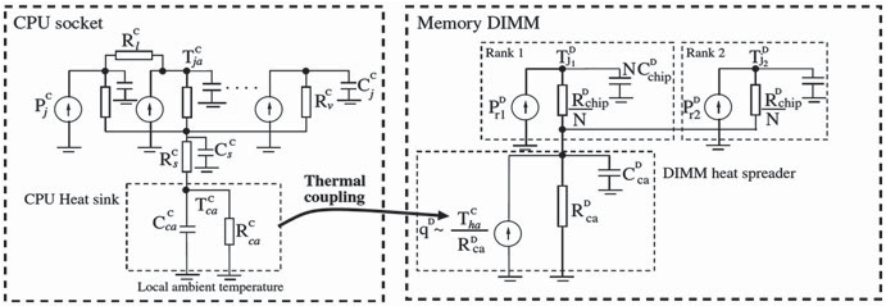

Fig. 11 RC network based temperature model [38]

the junction temperature of a DRAM chip; N is the number of ranks in a single DRAM chip. In addition, they observe that the temperature of DRAM is correlated with the temperature of CPU as the air inside a server flows from CPU to DRAM, thereby, air absorbing heat in CPU socket affects to the temperature of DRAM as it is equivalent to raising ambient temperature at DRAM. This phenomenon is called thermal coupling and modeled as follows:

$$q^D \propto \frac{T_{ha}^C}{R_{ca}^D} \tag{11}$$

where q^D is the dependent coupling heat source of the memory; T_{ha}^C is the heat sink sink temperature of the CPU; R_{ca}^D is the thermal resistance of the case to ambient of the memory DIMMs. This model is validated using Intel dual socket Xeon server, which shows a strong match between the actual measurement and the model within a 0.27 °C average error.

3.2 Datacenter Modeling

1) Computing Facility Basically, the temperature of servers in datacenter can be calculated using models in Sect. 3.1. However, for accurate temperature estimation for servers in a datacenter, we need to take into account interactions of generated heats among multiple servers in a server rack because servers are placed in a server rack in vertical direction and cold air flows from bottom to top of the server rack such that the heat generated at bottom is recirculated and affects to servers placed at upper side of the server rack. We call it *heat recirculation* in a datacenter. The amount of heat recirculation in a datacenter can be described by a cross-interference matrix, which is represented by $\Phi_{N\times N} = \{\phi_{i,j}\}$ where N is the number of servers in a server rack. $\phi_{i,j}$ indicates the contribution of the outlet heat rate of the i-th server in the inlet heat rate of the j-th one. Assuming Q_i^{out} and Q_j^{in} are, respectively, the

outlet and inlet heat rates for the i-th and j-th server, the inlet heat rate for j-th server can be calculated as follows [39]:

$$Q_j^{in} = \sum_{i=1}^{N} \phi_{i,j} Q_{out}^i + Q_{amb} + P_j \tag{12}$$

where Q_{amb} represents the heat rate delivered from cold aisle of a server room and P_j denotes the power consumed by j-th server.

In the vector form, we can write this relationship as follows:

$$\boldsymbol{Q}_{in} = \boldsymbol{\Phi}^T \boldsymbol{Q}_{out} + \boldsymbol{Q}_s + \boldsymbol{P} \tag{13}$$

Based on the heat rate, we can calculate the temperature at each server within a server rack using temperature models in Sect. 3.1.

2) Cooling Facility The typical cooling facility consists of a cooling tower, a chiller, and CRAH (or CRAC) as explained in Sect. 1.1. The heat generated by servers in a server room is absorbed by cold air provided from CRAH, and then, drawn by CRAH. CRAH exchanges the heat drawn from the server room with cold water (or air) provided from a chiller based on refrigeration cycle. In [42], A. Qouneh et al. provide a comparative and quantitative analysis of cooling power as varying processor utilization and adjusting the server room temperature accordingly. For further analysis of the power consumption of the cooling facility, some models have been presented in [40, 41] which model the power consumption based on thermo-fluid principles.

However, based on our analysis of real datacenter setups of our industrial partners in this work, we have observed that an alternative procedure can be used, where PUE mainly depends on the temperature set-point of server room (T_{room}), outside temperature (T_{out}), and total power consumed by servers (P_{cl}). Moreover, T_{room} is the dominant factor compared to the others. Thus, we can simply characterize PUE with respect to T_{room}. Figure 12 shows PUE with respect to T_{room}. As shown in this figure, the PUE of electrical and free cooling ranges $1.53 \sim 1.83$ and $1.08 \sim 1.14$, respectively. Assuming that T_{room} is set to the highest temperature of which servers in active mode can satisfy the maximum temperature limit, i.e., T_{pm}^{max}, we can model PUE as a function of the power consumption of servers, i.e., P_{pm}. By matching the results shown in Fig. 12, we can approximate the PUE with a relatively simple form, namely:

$$PUE = a_1 P_{pm}^2 + a_2 P_{pm} + a_3 \tag{14}$$

where a_1, a_2, and a_3 are curve fitting parameters. In the case of electrical and free cooling, the sets we have obtained for $\{a_1, a_2, a_3\}$ are $\{3.32 \times 10^{-5}, -9.45e \times 10^{-4}, 1.30\}$ and $\{0, 0, 1.08\}$, respectively. Then, the maximum (average) root mean square (RMS) error amounts to 4.38 % (0.76 %) and 0.56 % (0.56 %), respectively.

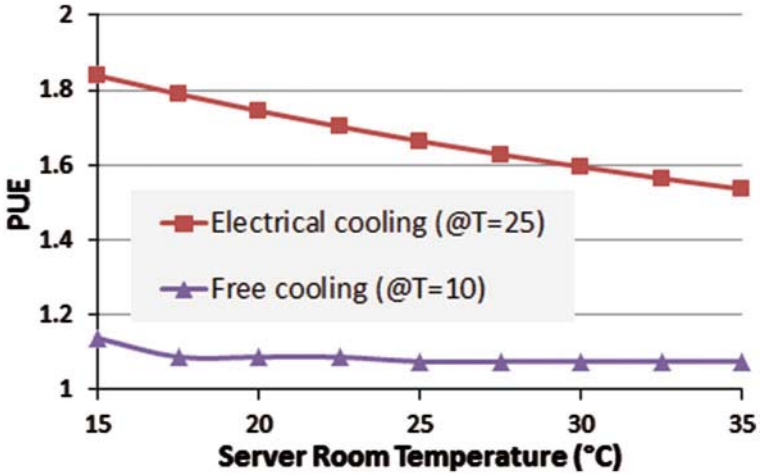

Fig. 12 Power usage effectiveness (PUE) in electrical and free cooling as power consumption of server varies [43]

Finally, the temperature of the server room, T_{room}, depends on CRAH efficiency, ϵ_{CRAH}, which is defined as follows [43]:

$$\epsilon_{CRAH} = \frac{T^{air}_{CRAH} - T_{room}}{T^{air}_{CRAH} - T^{water}_{CRAH}} \tag{15}$$

where T^{air}_{CRAH} represents temperatures of air exhausted from server room; T^{water}_{CRAH} is the temperature of chilled water flowing into the CRAH, which corresponds to the set-point of chiller and outside temperature when electrical and free cooling is used, respectively. Note that these values can be calculated using the procedure in [40], which depends on server power consumption, outside temperature, etc. Since ϵ_{CRAH} is always less than 1, T_{room} is always higher than T^{water}_{CARH}.

3.3 Monitoring System for Datacenters

Power management in datacenters is an area of increasing interest from several viewpoints as it is backed up by real concerns on energy usage and cost by modern computing systems. Data center computing applications and platforms have been typically designed without regard to power consumption. With increased awareness of energy cost, power consumption tracking and management is now an issue even for compute-intensive server clusters.

Datacenters ecosystem is facing an increasing need for decision support systems for datacenter management. Building and administration of datacenters are indeed evolving towards increasingly complex scenarios. IT infrastructure managers have

to optimize the datacenter utilization and costs, under several constraints generated by heterogeneous and diverging technical challenges: customer requirements, infrastructure costs, energy costs, physical space available, etc.

Datacenters that have some energy measuring capabilities carry out those monitoring tasks through Data-Center Infrastructure Management (DCIM). This concept includes the integration of IT and Facility Management, with the aim of centralising monitoring, management and intelligent capacity planning of data centre systems. Capacity planning focuses primarily on energy but also on power, space, network, IT equipment, cabling, cooling and environmental factors (temperature and relative humidity).

Understanding total capacity of all factors ultimately gives the optimal position where equipment should be moved, added or changed for optimised use of the available capacity. It also directly indicates where potential capacity is still present but unused (stranded capacity). Currently, in many datacenters this task is carried out manually or through site audits. This is a tedious, time-consuming and labour-intensive process, with a high risk of human error. An advanced DCIM system automates and simplifies this process, benefiting to IT and facility staff, but also to the energy efficiency of the datacenter.

A DCIM system can in particular map and manage the complete power chain and hence the energy capacity of the datacenter. Starting at the power sources (grid power or alternative power sources) up to the outlets on a rack Power Distribution Unit (PDU) or even the components within the servers, including all devices in between, DCIM systems are essential to plan energy flows and perform trending and analysis. They bring full access to all available devices, from facility to IT, as well as life cycle management, support contracts, and logical and physical cable connections.

Whereas DCIM systems are usually a good fit for large datacenters, the needs of small to medium-size urban datacenters are not adequately met today. Existing systems are generally too complex, pricy, difficult to use and not modular enough for urban facilities. In addition, solutions offered on the market today are generally proprietary and tend to lock their users in to single vendors. Innovative DCIM support systems for datacenter management are thus needed. PMSM (i.e., Power Monitor System and Management) [44–56], developed at EPFL in cooperation with Credit Suisse [45], is an example of such an innovation.

4 Power and Thermal Managements of Servers

As the servers operating workloads are time-varying, the accompanying power consumption and thermal profile vary as well. In order to maintain controlled power consumption and thermal dissipations, run-time dynamic power and thermal management (DPM and DTM) mechanisms are required. These management schemes exploit the utilization of power and temperature-affecting control knobs that exist in different layers of abstraction of the system, to aid in power and thermal reduction. In

addition, a fundamental challenge of any developed power or thermal management scheme is to have minimal, or preferably zero percent, performance degradation. If any management mechanism has a significant impact on the processing performance, it interferes with the architectural characteristics, hence considered a degrading rather than a managing element.

In this section, we explore the various power and thermal management mechanisms for server architectures. We first start by showing the state-of-the-art in power and thermal management solutions in Sect. 4.1. In Sect. 4.2, we explore our recent development in hierarchical power and thermal management schemes. Finally, we show our advances in power and thermal management in liquid-cooled server architectures.

4.1 Overview of CPU Power and Thermal Management Techniques

Power and thermal management solutions have been extensively existing in literature, which has been reflected in the various power and thermal management schemes [71, 72]. Nevertheless, we explore the recent works on power and thermal management in the state-of-the-art.

1) Temperature-Affecting Control Knobs As mentioned earlier, run-time management schemes utilize various control knobs that either reduce the causes of high heat generation, or increase the ability of the utilized cooling methodology. In the case of 3D MPSoCs, these control knobs are classified as follows.

a) Workload Activity Knobs At the software-level (application, system software, and OS), workloads can be altered and customized such that they can be thermally-aware. For example, task scheduling and task migration [73] have been extensively used to balance the workload on planar 2D MPSoCs [74]. Another example involves the intra-task instruction scheduling to prevent the processing element temperature from elevating to alarming values.

b) Circuit Switching Activity Knobs This class of control knobs affects the operating conditions of the processing element. These knobs may stall the processing element temporarily to reduce the heat generation, such as clock gating [75]. Alternatively, these knobs may reduce the operating speed of the processing element, which implies lower power consumption, hence lower heat generation, such as dynamic frequency scaling (DFS) or dynamic voltage and frequency scaling (DVFS) [75, 76].

c) Thermal Package Control Knobs The knobs at the thermal package level are responsible of changing the cooling capabilities, which is related to the injected fluid in the case of 3D MPSoCs with liquid cooling. For instance, the volumetric flow rate of the injected fluid can be varied by changing either the liquid pumping power [77], or varying the value of a flow-control valve [78].

2) Power and Thermal Management of Air-Cooled 2D and 3D MPSoCs Ogras et al. [79] proposes the control of power usage in processing elements (PEs) and routers by using model predictive control at design time, and Bogdan et al. [80] elaborate further this approach by considering both PEs and routers in the control scheme for voltage and frequency. However, they only consider power management and do not explore thermal control aspects. In fact, consolidating the power consumption in processing elements could undermine temperature issues while the power consumption is reduced. Thus, explicit thermal management schemes that include temperature as a key role in optimization or imposing temperature as a constraint are required for thermal balancing.

Initial research efforts have been focusing on combined power and thermal management by presenting a set of scheduling mechanisms for MPSoCs that perform temperature management at the system-level [81], using thread migration techniques to achieve temperature reduction in localized hot spots [75], or using a temperature-aware dynamic scheduling algorithm with negligible performance overhead [74]. These methods do not exploit history information and take reactive control actions based on the current thermal profile and frequency setting of the MPSoC.

However, recent works exploit history information to improve thermal management policies. Previous work [82] exploits a temperature forecast technique based on an auto-regressive moving average model. Another work proposes a novel technique that adapts the thermal management policy to the current workload characteristics [76], where the adaptation is done online exploiting information related to the workload history. Two recent approaches [83, 84] describe two methodologies to achieve thermal prediction by combining the information of thermal model, thermal sensors and power consumption statistical properties. These approaches rely on open-loop search or optimization where it is assumed that power can be estimated accurately.

More advanced solutions apply the concepts of *model-predictive control* (MPC) to turn the control from open-loop to closed-loop [87]. A chip-level power control algorithm based on optimal control theory is proposed [85], where the power consumption of the MPSoC is controlled to maintain the temperature of each core below a specified threshold. A recent work [86] proposes MPC utilization to solve the thermally-aware frequency assignment problem of a planar MPSoC.

However, most previous policies do not completely avoid hot-spots, but they simply reduce their frequency, because the interaction among the prediction method, the thermal behavior of the MPSoC and the frequency assignment of the MPSoC have not been addressed as a joint optimization problem.

In a similar vein, recent work considers dynamic thermal management for 3D MPSoCs. Previous work evaluates several policies for task migration and DVFS [88]. This previous work explores thermal profiles of adjacent processing elements being on the same vertical column (interlayer adjacent) or within the same layer (intralayer). Based on this analysis, a combined DVFS and a task migration policy, named *THERMOS*, is implemented. However, this work do not consider controlling the thermal packaging knobs, whether it is air or liquid cooling. Another work [89] integrates a thermally-aware task scheduler with DVFS on a two-tier 3D MPSoC

with eight cores. A recent paper proposes a temperature-aware scheduling method specifically designed for air-cooled 3D MPSoCs [91]. This method takes into account the thermal heterogeneity among the different layers of the system, but there is no study on the effect of the thermal packaging control knobs as active thermal management parameters. The resulting temperatures obtained in these papers are significantly high (85–120 °C). These results imply that 3D MPSoCs are prone to high temperatures, and with increasing power densities conventional thermal management techniques and air-based cooling are incapable of controlling the temperature while preserving system performance.

3) Thermal Management of Liquid-Cooled 3D MPSoCs Prior liquid cooling work [90] evaluates existing thermal management policies on a 3D MPSoC with a fixed-flow rate setting, and also investigates the benefits of variable flow using a policy to increment or decrement the flow rate based on temperature measurements, but without considering pump energy consumption.

Thermal management methods for 3D MPSoCs using a variable-flow liquid cooling have been recently proposed [77]. These policies use experimentally-driven sets of rules to control the temperature profile of the 3D MPSoC while ensuring performance requirements to be satisfied. These approaches use a centralized control concept, which is inappropriate if the controlled parameters increase [92], as in the case of targeted 3D MPSoC designs with liquid cooling in this work.

Recently, Qian et al. explore the use of a cyber-physical approach 3D MPSoCs thermal management with inter-tier liquid cooling [93]. They construct their control mechanism with software-based thermal estimation and prediction. They use a non-uniform liquid flow in different microchannels to meet the cooling demands of different modules. They take their control decisions on software-based thermal estimation and prediction. They use a non-uniform liquid flow in different microchannels, to meet the cooling demands of different modules. However, they have not shown the overhead of their software-based thermal estimation. Moreover, they do not show the feasibility of having a non-uniform flow in different channels, as a physical implementation.

4.2 Run-Time Hierarchical Power and Thermal Management for Server Architectures

We have proposed another proactive management scheme that relies on model predictive controller (MPC) [94]. In this work, we have developed a thermal management scheme that controls task scheduling, DVFS, and the cooling infrastructure. In particular, we target the cooling infrastructure case of interlayer liquid cooled 3D MPSoC, where we can alter dynamically the injected liquid flow rate. At each time interval, a new set of workloads arrive, and the management scheme allocates these tasks to various cores and sets the corresponding flow rate such that the predicted peak temperature is reduced while minimizing the 3D MPSoC power consumption (cooling

and computation power). Then for each processing element it applies MPC to the assigned workload such that the local predicted temperature is reduced while using the minimum computing energy possible via DVFS. The formulation of this problem is stated as follows:

$$J = \sum_{\tau=1}^{h} \left(\|\mathbf{R}\mathbf{p}_\tau\| + \|\mathbf{T}\mathbf{u}_\tau\| \right) \tag{16}$$

$$min \;\; J \tag{17}$$

$$subject\ to: \; f_{\min} \preceq \mathbf{f}_\tau \preceq \mathbf{f}_{\max} \;\; \forall\, \tau \tag{18}$$

$$\mathbf{x}_{\tau+1} = \mathbf{A}\mathbf{x}_\tau + \mathbf{B}\mathbf{p}_\tau \;\; \forall\, \tau \tag{19}$$

$$\tilde{\mathbf{C}}\mathbf{x}_{\tau+1} \preceq \mathbf{t}_{\max} \;\; \forall\, \tau \tag{20}$$

$$\mathbf{u}_\tau \succeq \mathbf{0} \;\; \forall\, \tau \tag{21}$$

$$\mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau \;\; \forall\, \tau \tag{22}$$

$$\mathbf{l}_\tau \succeq \mu \mathbf{f}_\tau^2 \;\; \forall\, \tau \tag{23}$$

$$-\mathbf{w} \preceq \mathbf{m}_{\tau+1} - \mathbf{m}_\tau \preceq \mathbf{w} \;\; \forall\, \tau \tag{24}$$

$$0 \preceq \mathbf{m}_\tau \preceq \mathbf{1} \;\; \forall\, \tau \tag{25}$$

$$\mathbf{p}_\tau = [\mathbf{l}_\tau; \mathbf{m}_\tau] \;\; \forall\, \tau \tag{26}$$

where matrices $\mathbf{A}$, $\mathbf{B}$ are related to the overall 3D MPSoC system description. These matrices represent the 3D MPSoC system using a coarse granularity of the thermal cells and where the sampling time of the resulting discrete-time system is T_{GC}. The horizon of this predictive policy is defined as h [87]. Then, the objective function J is expressed by a sum over the horizon.

In the cost function (Eq. (16)), the first term $\|\mathbf{R}\mathbf{p}_\tau\|$ is the norm of the power input vector p weighted by matrix $\mathbf{R}$. Power consumption is generated here by two main sources. Vector p is a vector containing normalized power consumption data the p tiers and the pumping power. Matrix $\mathbf{R}$ contains the maximum value of the power consumption of the tiers and the cooling system. The second term $\|\mathbf{T}\mathbf{u}_\tau\|$ is the norm of the required workload, but not yet executed. To this end, the weight matrix $\mathbf{T}$ quantifies the importance that executing the required workload from the scheduler has in the optimization process. Then, Inequality (18) defines a range of working frequencies to be used, but this does not prevent from adding in the optimization problem a limitation on the number of allowed frequency values.

Equation (19) defines the evolution of the 3D MPSoC according to the present state and inputs. Equation (20) states that temperature constraints should be respected at all times and in all specified locations. Since the system cannot execute jobs that have not arrived, every entry of $\mathbf{u}_\tau$ has to be greater than or equal to 0 as stated by Eq. 21. The undone work at time τ, u_τ is defined by Eq. 22. Equation 23 defines the relation between the power vector $\mathbf{l}$ and the working frequencies. μ is a technology-dependent constant.

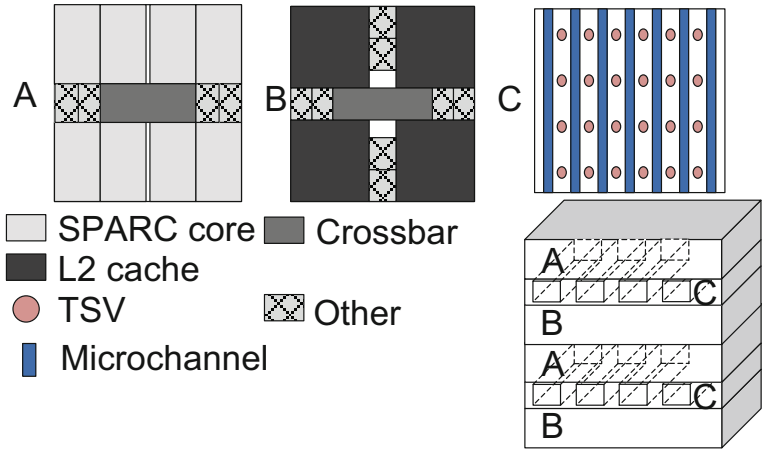

Fig. 13 Schematic diagram of the four-tier liquid-cooled 3D MPSoC used in the thermal evaluation of the proposed thermal management scheme

Then, Eqs. 24–25 define constraints on the liquid cooling management. The normalized pumping power value (**m**) scales, and any time instance τ, from 0 (no liquid injection) to 1 (power at the maximum pressure difference allowable), as shown in Eq. 25. Moreover, the maximum increment/decrement change in the pumping power value from time (τ) to ($\tau + 1$) is limited by a another normalized value **w**, as shown in Eq. 24, which models the mechanical dynamics of the pump.

Equation 26 defines formally the structure of vector **p**. Vector $\mathbf{l} \epsilon \Re^{\mathbf{p}}$ is the power input vector, where p is the number of tiers of 3D MPSoC.

Finally, the control problem is formulated over an interval of h time steps, which starts at current time τ. Indeed the result of the optimization is an optimal sequence of future control moves (i.e., amount of workload to be executed in average for each tier of the 3D MPSoC which is stored in vector **f**). Then, only the first samples of such a sequence are applied to the target 3D MPSoC, while the remaining moves are discarded. Thus, at each next time step, a new optimal control problem based on new temperature measurements and required frequencies is solved over a shifted prediction horizon (e.g., the "receding-horizon" [87] mechanism), which represents a way of transforming an open-loop design methodology into a feedback one, as at every time step the input applied to the process depends on the most recent measurements.

To evaluate the effectiveness of this thermal control, we apply this management scheme on a four-tier 3D MPSoC based on the UltrsSPARC T1 MPSoC [112], which is shown in Fig. 13. In addition, we compare it against different state-of-the-art thermal management techniques, which are as follows:

- **Liquid cooling with LB (LC_LB)** [95]: It applies the maximum cooling flow rate, while the jobs are scheduled with load balancing policy (LB). LB balances the workload by moving threads from a core's queue to another if the difference in queue lengths is over a threshold.

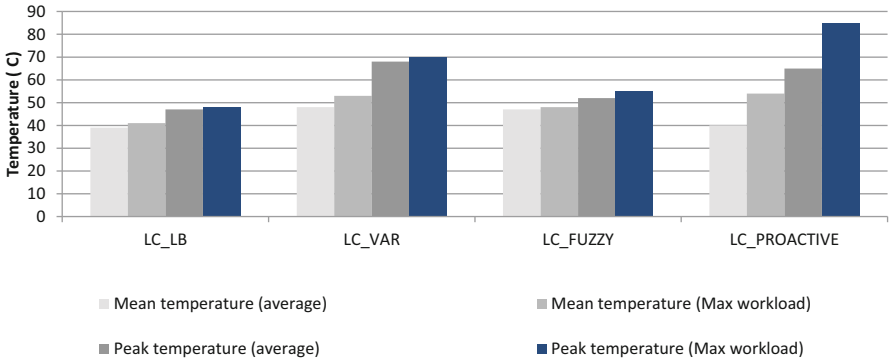

Fig. 14 Peak and average temperatures observed using all the policies, both for the average case across all workloads and maximum workload on four-tier 3D MPSoC [94]

- **LUT-based flow rate control with LB (LC_VAR)** [77]: It dynamically changes the flow rate based on the predicted maximum temperature, while the jobs are scheduled with LB.
- **Fuzzy-logic control (LC_FUZZY)** [96]: This mechanism utilized fuzzy logic in deriving thermal management mechanism that controls the variable liquid flow rate and DVFS.

In addition we refer to this management scheme as *LC_PROACTIVE* in the following paragraphs. In this evaluation of different thermal management policies, *LC_PROACTIVE* is compared with respect to the other management techniques mentioned above based on the:

- Maximum and average temperatures.
- Computational and cooling power consumption.

Thermal impact of all the policies on a four-tier 3D MPSoC (cf. Fig. 13) is shown in Fig. 14. This figure shows that LC_LB reduces the peak temperature to 47 °C, whereas LC_FUZZY and LC_VAR push the system into a higher peak of 52 and 67 °C, respectively, but still avoids any hot-spots. This is the similar case in LC_PROACTIVE, where the peak temperature reaches 84 °C. The alteration between the peak temperature comes from the fact that main target is to reduce the peak temperature to any value below 85 °C. However, since each technique has a different management policy, with different control elements, the peak and average temperatures are affected.

Figure 15 shows the total consumed power when running the various policies on the four-tier MPSoC with the average workload [94]. Energy consumption values are normalized with respect to the load balancing policy on the 3D-MPSoC with LC_LB. In this figure, LC_PROACTIVE manages to reduce the cooling power and the overall system power by 60 and 23 %, respectively, with respect to LC_LB. Moreover, LC_PROACTIVE even reduces the cooling energy more than LC_VAR and LC_FUZZY by 40 and 22 %, respectively.

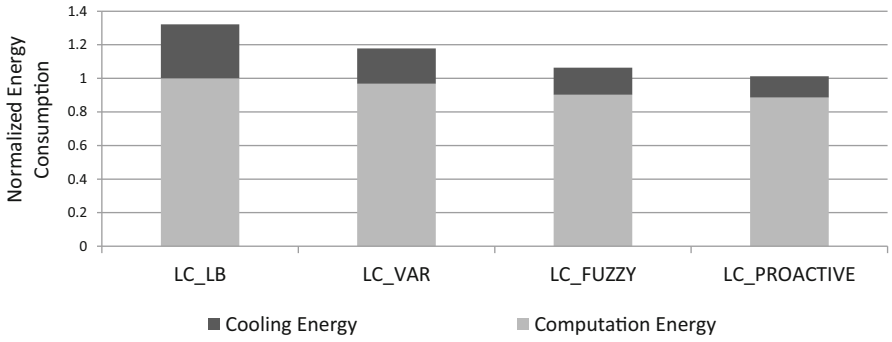

Fig. 15 The normalized energy consumption in the whole system (chip and cooling network) [94]

4.3 Design-Time Power and Thermal Optimizations

In addition to run-time management schemes, several works conduct power and thermal optimizations at design-time. In the case of MPSoCs, several approaches have been taken to optimize the power utilization and heat generation or dissipation. At the platform level, different modules can be designed to reduce the overall power density, hence heat generation, while preserving the system functionality. This approach has been taken recently in low-power (hence low temperature) processor designs such as ARM big.LITTLE processing architecture [97]. Another approach at the platform level is to reduce the operating power supply of the platform to near-threshold values [98]. Near-threshold computing allows the processing units to operate close to the voltage threshold value of the used transistor, hence reducing the overall power and thermal density.

In the case of 3D MPSoCs, recent work proposes multiple supply voltages utilization to optimize the voltage islands distribution in 3D MPSoCs [99]. In this work, a temperature-aware voltage island generation methodology is proposed that formulates this problem as a mixed-integer linear programming (MILP) problem. The main aim in this work is to minimize the thermal hotspots in 3D MPSoCs while keeping the performance and timing requirements satisfied. The interdependency between power and heat densities made it feasible to formulate this problem and achieve significant results.

Another work utilizes various microarchitectural techniques to control the thermal hotspots in 3D MPSoCs via thermal herding [100]. This technique explores different architectural disciplines by spitting several microarchitectural blocks between the different layers of 3D MPSoC to enhance the throughput while controlling the thermal hotspots such as, register file splitting. This splitting is based on general application trends and the significance of particular instructions or data locations to the execution flow.

Previous works have investigated the rearrangement of various hardware modules within the MPSoC to minimize the global thermal impact, which is also known in

literature as *temperature-aware floorplanning*. Initial work on temperature-aware floorplanning [101] has shown its significant impact on reducing the peak temperature. This work has defined a metric called *thermal diffusion* that resembles the lateral heat dissipation. This metric has been used in an optimization problem to maximize the gains of *thermal diffusion*. Other similar works have proposed simulated annealing utilization [102] or genetic algorithms [103] to achieve temperature-aware floorplanning.

In the context of 3D MPSoCs, temperature-aware floorplanning has also been extended by including the interlayer thermal dissipation and interconnect characteristics [102, 104–106]. For example, initial work has been proposed [107] for temperature-aware microarchitectural floorplanning. The main objective in this work is to place the processing submodules of a single processor in several layers such that the wire lengths and the temperatures are minimized. To achieve this, a mixed integer linear programming (MILP) problem is formulated to minimize the weighted sum of performance, area and thermal-related aspects. Another work uses simulated annealing to minimize the temperature of 3D MPSoC via floorplanning [105] by considering the additional power consumption of the interconnects.

As for liquid-cooled 3D MPSoCs, Mizunuma et al. use their thermal model to explore floorplanning solutions to homogenize temperature distributions in this architecture [108]. The results in this work, which is further assisted by the observations in other work [96], show that in the case of liquid cooled 3D MPSoC, temperature-aware floorplanning follows the trend of placing more heat dissipating modules at the fluid inlet port, while lower heat dissipating modules at the outlet port. In other words, the optimal heat dissipation pattern for temperature-aware floorplanning would be monotonically decreasing from the distance of the fluid inlet port.

Our recent proposed framework, namely *GREENCOOL*, optimizes the active cooling path of microchannel-based iterlayer liquid cooled 3D MPSoCs to balance the thermal profile of the target 3D MPSoC while significantly reducing the active cooling energy demands [109]. This design-optimization methodology uses the concept of channel modulation, where we change the microchannel aspect ratio (channel width/channel height) to enhance the heat transfer capability from the target 3D MPSoC via changing the convective thermal resistance [110]. Using the conventional CMOS fabrication process for etching the channels, such as deep reactive iron etching [111], it is possible to modulate the width of the channel from inlet to outlet (and hence its aspect ratio) and create any kind of channel width profile, while keeping the height of the channels constant. Thus, channel width modulation requires only a change in the patterns on the masks used for etching channels amounting to minimal additional fabrication costs. To summarize, using careful design it is possible to modify the local channel aspect ratios so as to contain the pumping power while constraining the thermal gradients.

To understand how the channel width affects the change in temperature due to convection (ΔT_{conv}) in detail, an analysis is performed on a single microchannel shown in Fig. 16. We start by the following set of equations governing the Nusselt number (a dimensionless form of heat transfer coefficient), and the product of friction factor

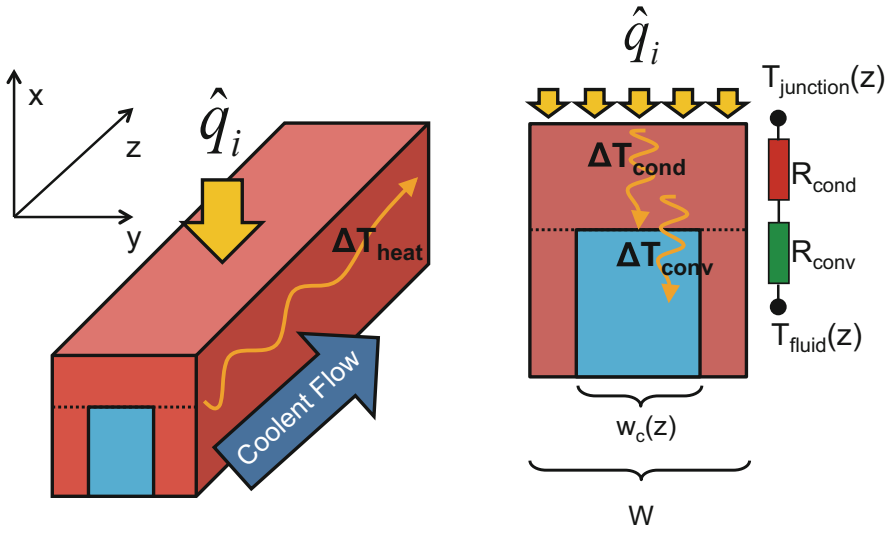

Fig. 16 Test structure: a single microchannel cooling a strip of an IC with uniform heat flux distribution. The figure shows both the 3D and the cross-sectional views [109]

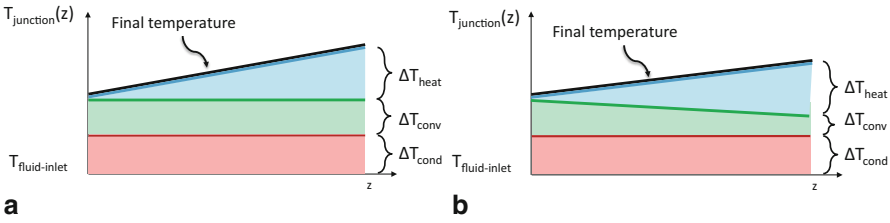

Fig. 17 R_{conv} as a function of the channel width for the structure in Fig. 16

and Reynold's number for microchannels, under fully developed conditions [110]:

$$\begin{aligned} Nu &= 8.235 \cdot (1 - 2.0421AR + 3.0853AR^2 - 2.4765AR^3 + 1.0578AR^4 \\ &\quad - 0.1861AR^5) \\ fr \cdot\ Re &= 24 \cdot (1 - 1.3553AR + 1.9467AR^2 - 1.7012AR^3 + 0.9564AR^4 \\ &\quad - 0.2537AR^5), \end{aligned} \tag{27}$$

where AR is the aspect ratio reciprocal (height/width) of the channel. Using the Nusselt number, the heat transfer coefficient (a measure of the amount of heat transferred per unit area for one Kelvin difference in temperature between the fluid and the microchannel wall surface, expressed in W/m^2K) can be written as:

$$h = \frac{k_{coolant} \cdot Nu}{d_h} \tag{28}$$

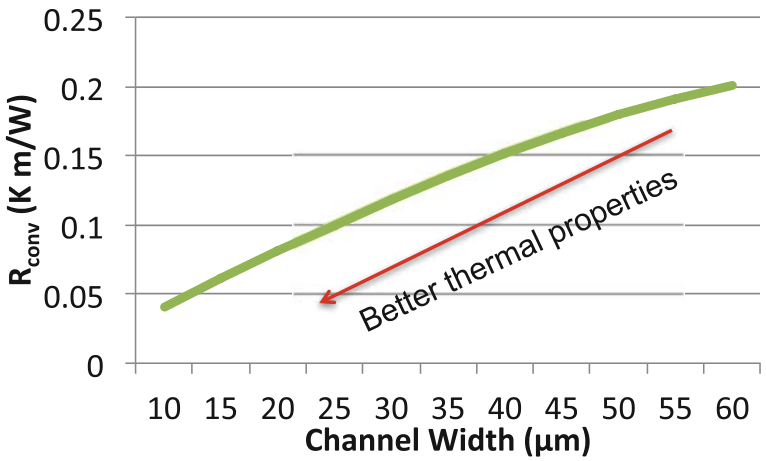

Fig. 18 Junction temperature distribution for the structure in Fig. 16. **a** With uniform non modulated channel width. **b** With modulated channel width to compensate for sensible heat absorption [109]

where $k_{coolant}$ is the thermal conductivity of the coolant and d_h is the hydraulic diameter of channel. The effective heat transfer coefficient as seen by the junction looking down the channel from the top can be written by projecting the heat transfer coefficient above from the side wall surfaces onto the top as follows:

$$h_{eff} = h\frac{2 * H_C + w_C}{W} \tag{29}$$

where H_C is the height and w_C is the width of the channel, and W is the total width of the structure as shown in Fig. 16. The convective resistance R_{conv} for this structure can be obtained as a reciprocal of this quantity. The R_{conv} for this structure is plotted as a function of w_C in Fig. 17, assuming water as the coolant, $H_C = 100$ µm, $W = 100$ µm and varying w_C from 10 to 50 µm.

Figure 17 shows that the convective resistance (and also ΔT_{conv}) drops quickly as the channel width is reduced. Since the goal is to modify the convective resistance to compensate for ΔT_{heat}, it can be postulated that the channel width must no longer be a constant but instead should be a function of the distance along the channel $w_C(z)$. The width must be larger near the inlet where the fluid temperature is low and smaller near the outlet where the fluid temperature is high. Hence, theoretically, for the case of uniform heat flux, it is possible to lower the final thermal gradient by steadily modulating the channel width from inlet to outlet, as shown in Fig. 18b.

GREENCOOL uses this principle in formulating an optimal control problem to find the optimal channel width profile for each microchannel, from the fluid inlet to outlet ports. The target of this optimization is to minimize the peak temperature and thermal gradients of the 3D MPSoC, as well as reducing the energy needed by cooling. When applied various 3D MPSoC architectures, significant thermal gradient reductions as well as cooling power savings, with respect to worst-case designs.

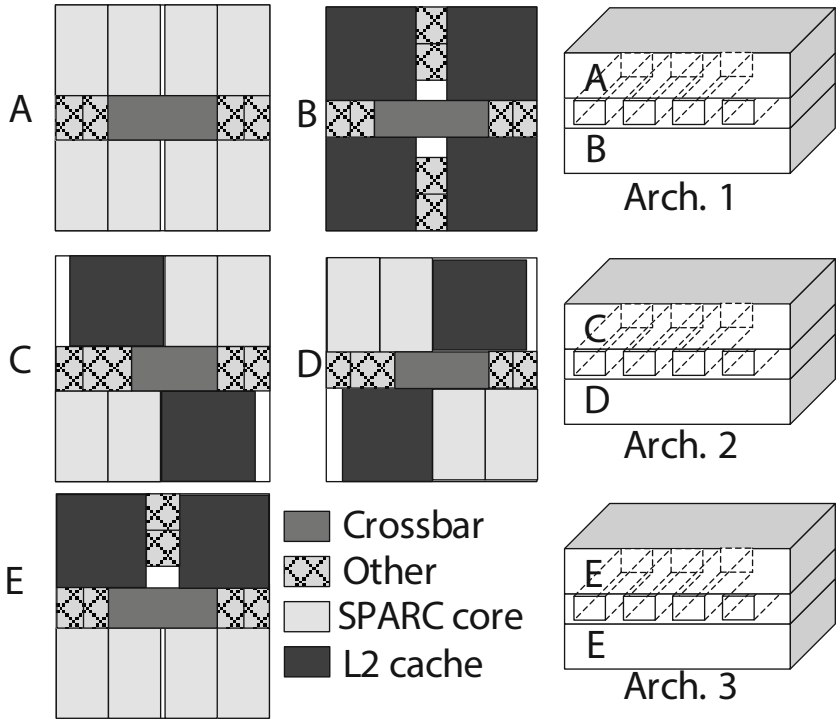

Fig. 19 Layout of the 3D-MPSoCs used in our experiments [113]

For instance, when *GREENCOOL* is applied to different architectural layouts of the UltraSPARC T1 Niagara MPSoC [112], a 31 % thermal gradient reduction is observed. Figure 19 shows the layout of the different two-dies 3D-MPSoCs used in this experiment. The dies are of size 1 cm × 1.1 cm and the heat flux densities range from 8 to 64 W/cm^2 in the two dies. Further details about the floorplan and power dissipations can be found in pervious works [77, 96, 112].

In this experiment, the worst-case (peak) power dissipation of the 3D-MPSoC functional elements [77, 96, 112] (obtained using measurements) are used in the optimization process. *GREENCOOL* achieves a thermal gradient reduction of 31 % (23 °C to 16 °C). When the peak heat flux levels were replaced by average values, this same optimal channel modulation configuration manages to reduce the thermal gradient by 21 % compared to the uniform channel width case. The thermal gradients obtained for the different cases and for various channel types are plotted in Fig. 20.

In another set of experiments to demonstrate the energy-efficiency of *GREENCOOL*, significant cooling energy savings that reach up to 80 % has been achieved [109]. Furthermore, *GREENCOOL* aids in developing efficient cooling layout in the cases where uniform cavity utilization is infeasible.

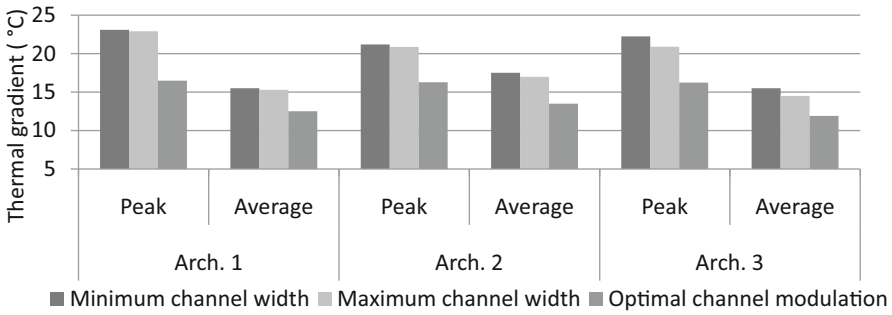

Fig. 20 Thermal gradients observed in the different 3D-MPSoC architectures dissipating peak and average level heat fluxes, using maximum, minimum and optimally modulated channel widths [113]

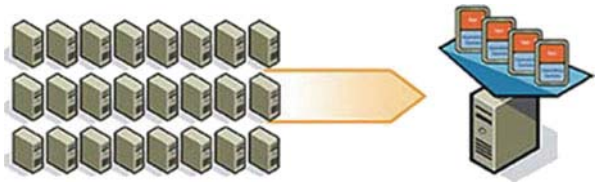

Fig. 21 Concept of server consolidation

5 Power and Thermal Managements for Server Clusters

5.1 Conventional Solution to Minimize Power Consumption for Server Clusters

In datacenters, servers are normally severely under-utilized, less than 30 % in more than 90 % of the total time [3]. In addition, as explained in Sect. 3, the power consumption of servers is not proportional to the utilization, i.e., the idle power consumes around 50 % of the peak power consumption. Due to the poor energy-proportionality, the power consumed by servers in datacenters can be reduced as we minimizes the number of active servers by packing workloads into the minimal number of active servers [46]. The technique is called *Server consolidation*. The key enabler to realize the solution is server virtualization, explained in Sect. 2.1, as it enables to migrate workloads easily by encapsulating workloads with a form of virtual machines (VMs) and run multiple VMs in a single physical server with the aid of hypervisor. Figure 21 shows the concept of the server consolidation in a virtualized server environment.

In the server consolidation, we need to take care such that the performance after the consolidation should not be degraded, or within an acceptable range. To achieve this goal, many works have developed the consolidation solutions such that the sum of the peak required utilization among co-located VMs does not exceed the server's capability [46]. However, as analyzed in many works [6], the peak utilization happens rarely and much higher than off-peak (e.g., 90th/95th/99th percentile) values. Thus,

the server consolidation based on the peak value makes us to lose the opportunity for further power savings. To overcome the conservative solution, some works [6, 47] presents server consolidation solution which packs VMs into servers based on off-peak (e.g., 90th/95th/99th percentile) of server utilization.

The advantage obtained from the server consolidation is obvious in terms of power savings. However, it may cause unexpected performance degradation due to the conflict of using shared resources among co-located VMs, especially last-level cache (LLC) [48, 49]. Tickoo et al. [49] analyzed how the performance is degraded as VMs are allocated to share LLC with others using *SpecJBB* and *Sysbench* benchmark suites in order to evaluate the amount of the performance degradation caused by different cache usage characteristics of co-located VMs. The results show that sharing LLC between two copies of VMs both hosting SpecJBB leads to $\sim 30\,\%$ performance degradation while a case of sharing LLC between VMs hosting SpecJBB and Sysbench leads to $\sim 20\,\%$ degradation. In [50], Govindan et al. characterize the amount of interference with a set of parameters, i.e., effective number of used sets and ways. Then, it presents a solution to allocate VMs by accounting for the amount of the interference such that the performance interference becomes minimized while meeting the required performance requirement.

5.2 Correlation-Aware Power and Temperature Management

We can achieve further compact server consolidation by considering correlation among workload variation. In [51], Verma et al. found out that workloads running on datacenters are strongly correlated one another. In order to achieve further power savings while maintaining quality of service (QoS) level, correlations among VMs' workload have been exploited in recent works [51–54]. In [51], Verma et al. presented a clustering-based correlation-aware VM placement solution. To efficiently characterize the workload correlation, it first transforms utilization traces with a form of an envelope which is defined as a binary sequence which is '1' when CPU utilization is higher than a threshold value, e.g., 90th percentile, otherwise '0'. Second, it clusters VMs such that the envelops of VMs' CPU utilization included in different clusters do not overlap. Finally, it allocates VMs to physical servers such that VMs in different clusters are co-located in a single server so as to minimize the possibility when peaks are coincided. To meet the performance requirement after the consolidation, it allocates VMs based on their off-peak utilization demands (e.g., 90th percentile) while reserving a shared peak buffer to handle resource demand higher than the off-peak value for all co-located VMs. However, this approach is applicable only when the envelops of VMs are stationary and distinctively different one from another, thereby, producing multiple clusters. Hence, it does not work well with applications with non-stationary and fast-changing VM behaviors.

In [52], Meng et al. presented a joint-VM sizing technique that pairs two uncorrelated VMs into a *super-VM* and provision *super-VMs* by predicting the the aggregated workloads. However, once *super-VM*s are formed, this solution does not

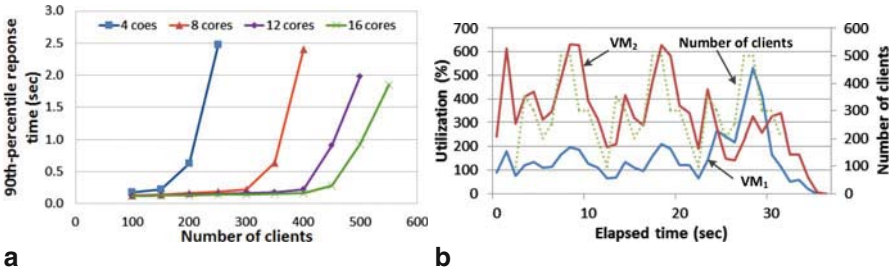

Fig. 22 **a** 90th-percentile response time (in seconds) with respect to the number of clients and allocated cores. **b** Variations of CPU utilization of two index searching nodes (ISNs) with respect the number of clients

consider the correlations of VMs within a same *super-VM* anymore. Thus, it may lose the chance of further power savings by leveraging time-varying correlations in scale-out applications. In [54], Halder et al. extends the scheme such that aggregated workload of multiple VMs can be utilized for VM placement. However, this solution can be applicable only when future servers' utilization is perfectly known.

However, all the correlation-aware VM placement solutions target conventional HPC application, thereby they cannot work well with scale-out applications whose workload characteristics are quite different, as we explained in Sect. 1.2. To overcome the drawbacks of existing solutions, we [56] developed a power management solution for datacenters hosting scale-out application, especially targeting following distinctive workload characteristics of scale-out applications. We used a websearch application in CloudSuite [4] as a proxy to characterize the workload characteristics of scale-out applications.

- **User-interactive and fast changing:** Owing to the user-interactive nature of scale-out applications, responsiveness, quantified in terms of latency, is the first criterion we need to satisfy when running the applications. Therefore, we should provision VMs in a conservative manner, based on the peak (or Nth percentile according to QoS requirement) resource demand of each VM. As the scale-out applications are commonly highly parallel, we can meet the required performance level for running VMs by assigning the right number of cores. Figure 22a demonstrates the 90th percentile response time of a websearch cluster with respect to the number of queries as we vary the number of allocated cores to host the websearch cluster from 4 to 16. Furthermore, the resource demand is time-varying and mostly lower than the provisioned amount of resources. However, as described in [6], due to the significant performance degradation caused by the long transition latency between power modes and fast changes of resource demands, dynamic power gating (turning on/off cores) cannot be applicable to such applications. Motivated by these observation, *it is required the solution allocating the right number of cores for each VM according to its peak (or off-peak depending on QoS level) resource demand to guarantee QoS levels to all VMs while scaling v/f level to achieve power savings.*

Table 2 Performance metrics of a web search application co-located with a VM running PARSEC benchmark: numbers in parenthesis show the case when a web search application is running alone

	IPC	L2 MPKI	L2 miss rate (%)
w/ Backshcoles	0.76 (0.75)	2.38 (2.40)	11.28 (11.57)
w/ Swaptions	0.75 (0.77)	2.32 (2.43)	11.02 (9.63)
w/ Facesim	0.70 (0.70)	2.41 (2.36)	11.41 (11.31)
w/ Canneal	0.76 (0.78)	2.46 (2.43)	11.76 (11.67)

The amount of required CPU utilization of websearch clusters is dynamically varied as the amount of user requests changes over time. Figure 22b shows the CPU utilization traces for two VMs with respects to the number of queries, each of which VM is an index serving node (ISN), in a single web search cluster to process queries requested from the varying number of clients. As shown in the figure, the CPU utilizations of both VMs are highly synchronized with the variation of the number of clients. Moreover, the loads between VMs in a cluster are not perfectly balanced because the CPU utilization depends on the amount of matched results corresponding to a user request. Thus, we can improve the resource utilization by sharing cores among multiple VMs, such that they can more flexibly use cores depending on their time-varying resource demands.

- **Negligible performance degradation caused by LLC conflict:** As analyzed in [4], the performance degradation caused by sharing caches is negligible because the required memory footprint is too large to be sustained by on-chip caches. Table 2 shows the measured performance metrics of a websearch application when it is allocated to share core and cache with various applications in PARSEC benchmark suite. We compared instruction per clock cycles (IPC), L2 miss-per-kilo-instruction (MPKI), and L2 miss ratio (percentage), which are obtained using Xenoprof patched for the AMD15h Bulldozer architecture [55]. The numbers in parenthesis show the cases before co-location. As presented, there are only negligible variations over all the metrics before and after the co-location, which correspond to a negligible performance degradation due to cores sharing. Motivated by these observations, *we can efficiently utilize multiple cores in a server by allocating co-located VMs to share the cores assuming that the performance degradation is negligible.*
- **High correlation among VMs:** As jobs are distributed to multiple VMs in a cluster, workloads of VMs within a same cluster are highly correlated compared to different clusters (or services). In Fig. 22b, we can observe the intra-cluster correlation between two VMs in a cluster, both of which are strongly synchronized with the variation of the number of clients. Thus, *the proposed solution takes into account the pervasive correlation in scale-out applications, i.e., within a cluster as well as among clusters, such that correlated VMs are not co-located.*

  Figure 23 illustrates an example of demonstrating the effectiveness of the correlation-aware VM provisioning solution. Let's assume that we have two servers, $Server_1$, and $Server_2$, each of which consists of eight cores, and four

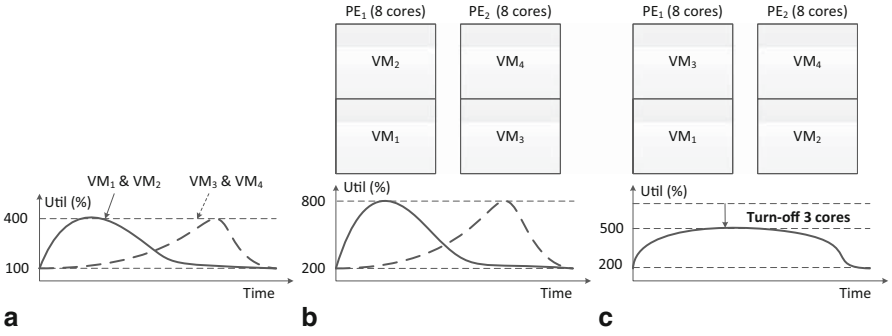

Fig. 23 A motivational example to show the effectiveness of considering correlation information: utilization traces (**a**); VM allocations (**b**) without considering correlation, and with considering correlation (**c**)

VMs, i.e., VM_1, VM_2, VM_3, and VM_4, where $\{VM_1, VM_2\}$ and $\{VM_3, VM_4\}$ are in $Service_1$ and $Service_2$, respectively. We assume that all the VMs have the same amount of the tail distribution on CPU utilization, and VMs in a same service are highly correlated (as load is quite well balanced among VMs) while VMs in different services are less correlated. Figure 23a shows an example of CPU utilization traces. If we do not take into account the correlation, we allocate sets of $\{VM_1, VM_2\}$ and $\{VM_3, VM_4\}$ into $Server_1$ and $Server_2$, respectively, as shown in Fig. 23b. In this case, the maximum CPU utilization amounts to 800 % per each server, thereby, all cores should be in active state. However, if we pair $\{VM_1, VM_3$ and $\{VM_2, VM_4\}$, as shown in Fig. 23c, the actual maximum utilization can be lowered down to 500 for both servers, thereby, we can turn-off (or idle low-power state) three cores per each server and/or lower v/f level without any quality degradation.

Based on the observations and motivations above, we presented a server consolidation solution in [56]. First, to efficiently capture correlation information, they present a low-complexity measure for evaluating workload correlation among co-located VMs, and then, developed VM allocation algorithm.

1) Efficient Correlation Measure for VM Allocation: *Pearson product-moment correlation coefficient*, or *Pearson's correlation*, is most widely used correlation measure to quantify the correlation of used CPU utilization among VMs [53]. It is calculated as the ratio of covariance of the two random variables to the product of their standard deviations. However, the overhead to calculate the metric for a certain time interval is high for a short time period due to the concentrated computation at the end of the time period, because it utilizes the average values of CPU utilization samples collected during each time period. In addition, Pearson's correlation is also partly inefficient because the value reflects correlation throughout the corresponding time interval because we only require the correlation at (off-)peak utilizations in VM placement. Equation (30) is presented in [56] as a new measure to quantify the correlation between two VMs to overcome the inefficiency of the conventional

correlation metric.

$$Cost_{i,j}^{vm} = \frac{\hat{u}_{cpu}(VM_i) + \hat{u}_{cpu}(VM_j)}{\hat{u}_{cpu}(VM_i + VM_j)} \tag{30}$$

where $Cost_{i,j}^{vm}$ represents the newly defined correlation measure between VM_i and VM_j. $\hat{u}_{cpu}(VM_i)$ is a reference utilization of VM_i, which is either the peak or the Nth percentile value depending on QoS requirement. The numerator represents the worst-case peak CPU utilization when the peaks of two VMs coincide, while the denominator is an aggregated actual peak utilization when VM_i and VM_j are co-located into a same server. Thus, the higher $Cost_{i,j}^{vm}$, the lower correlation between VM_i and VM_j. Moreover, we can update the values at each sampling period of utilization. Thus, we can save memory space to store all samples as well as evenly distributing computational effort to measure the correlation across a certain time horizon. Using our new $Cost_{i,j}^{vm}$ function, we can model correlations among all VMs by constructing a 2-D matrix, namely, $\mathcal{M}_{cost}^{vm}$ where the (i,j)-th element corresponds to $Cost_{i,j}^{vm}$.

2) Correlation-Aware VM Allocation for Scale-Out Applications Based on the correlation metric in Eq. (30), we can minimize the correlation among co-located VMs in $Server_i$, i.e., $\mathbb{V}_i^{alloc} = \{VM_{i,1}, \cdots, VM_{i,n_i^{vm}}\}$ where n_i^{vm} is the number of VMs allocated to $Server_i$, by allocating VMs such that a weight sum of $Cost_{i,j}^{vm}$ is minimized while the sum of $\hat{u}_{cpu}(VM_{i,j})$ in the server does not exceed the total CPU capability of the server, i.e., Cap_i. The correlation of $Server_i$ is defined as follows:

$$\overline{Cost}_i^{server} = \sum_{j=1}^{n_i^{vm}} w_{i,j}^{vm} \cdot \left(\sum_{k=1 \& k \neq j}^{N_{vm}} \frac{Cost_{j,k}^{vm}}{n_i^{vm} - 1} \right) \tag{31}$$

where $w_{i,j}^{vm}$ represents a weight of $VM_{i,j}$, defined as the ratio of $\hat{u}(VM_{i,j})$ to the sum of $\hat{u}(VM_{i,j})$'s of all co-located VMs in $Server_i$.

The problem is a well-known bin-packing problem [43]. In order to reduce the solution complexity within negligible solution quality degradation, Kim et al. present a heuristic based on a *First-Fit-Decreasing* where it first manipulates VMs having the highest utilization among unallocated VMs. Figure 24 shows a pseudo code to achieve this goal. In this algorithm, we periodically adjust VM allocation at every t_{period} based on the workload predictions. The algorithm largely consists of two phases: (1) *UPDATE* (lines 1 $\sim$ 8) and (2) *ALLOCATE* (lines 9 $\sim$ 18). In the *UPDATE* phase, we initialize parameters and update CPU utilization statistics. Then, we allocate VMs to servers in the *ALLOCATE* phase.

The *UPDATE* phase consists of five steps, namely:

- *Initialization*: a set of unallocated VMs ($\mathbb{V}^{unalloc}$), sets of allocated VMs ($\mathbb{V}_i^{alloc}$), remaining capacity (Rem_i) for all servers, and a correlation threshold (TH_{cost}) (lines 1 $\sim$ 4).
- *Prediction*: predict the workload based on history, as we previously prepared in [43] (line 5).

Algorithm 1 Correlation-aware VM placement

1: $\mathbb{V}^{unalloc} = \{VM_i | 1 \leq i \leq N_{vm}\}$
2: $\mathbb{V}_i^{alloc} = \{\phi\}$, where $1 \leq i \leq N_{server}$
3: $Rem_i = Cap_i$, where $1 \leq i \leq N_{server}$
4: Set an initial correlation threshold, TH_{cost}
5: Predict workloads for all VMs
6: Sort $\mathbb{V}^{unalloc}$ in descending order of $\hat{u}_{cpu}(VM_i)$
7: Update a correlation map, $\mathcal{M}_{cost}^{vm}$
8: Calculate the estimated number of servers, $\tilde{N}_{server}$

UPDATE

9: **while** ($\mathbb{V}^{unalloc}$ is not empty) **do**
10: **for** $i = 1 \rightarrow \tilde{N}_{server}$ **do**
11: VM_j=FINDVM($\mathbb{V}^{unalloc}$,$\mathbb{V}_i^{alloc}$,Rem_i,TH_{cost})
12: **while** ($VM_j \neq NULL$) **do**
13: $\mathbb{V}_i^{alloc} = \mathbb{V}_i^{alloc} + \{VM_j\}$
14: $Rem_i = Rem_i - \hat{u}(VM_j)$
15: $\mathbb{V}^{unalloc} = \mathbb{V}^{unalloc} - \{VM_j\}$
16: VM_j=FINDVM($\mathbb{V}^{unalloc}$,$\mathbb{V}_i^{alloc}$,Rem_i,TH_{cost})
17: $TH_{cost} = \alpha \cdot TH_{cost}$
18: Sort servers in descending order of Rem_i

ALLOCATE

Fig. 24 The correlation-aware VM placement presented in [56]

- *Sorting*: we sort VMs in $\mathbb{V}^{unalloc}$ in descending order of predicted $\hat{u}_{cpu}(VM_i)$ to reduce the fragmentation of the bin-packing problem (line 6)
- *Update cost function*: update $\mathcal{M}_{corr}^{vm}$ by updating the $Cost_{i,j}^{vm}$ for all VM pairs (line 7)
- *Estimate the number of active server sets*: determine the number of estimated active servers, i.e., $\tilde{N}_{server}$, as presented in Eq. (32) (in line 8):

$$\tilde{N}_{server} = \frac{\sum_{i=1}^{N_{vm}} \tilde{\hat{u}}_{cpu}(VM_i)}{N_{core}} \tag{32}$$

where $\tilde{\hat{u}}_{cpu}$ represents an estimate of $\hat{u}_{cpu}$. Then, $\tilde{N}_{server}$ is equal to the minimum number of servers to accommodate all VMs in $\mathbb{V}^{unalloc}$. We provision VMs to reduce the number of active servers while satisfying performance requirements.

Based on the update information and the predictions, we allocate VMs in *ALLOCATE* phase by iterating the procedure (in line 10 $\sim$ 18) until all VMs are allocated to $\tilde{N}_{server}$ servers (line 9).

- Select a server having the largest remaining CPU capability, i.e., Rem_i (line 10).
- Find a VM to be allocated into $Server_i$ which has the highest $\overline{Cost}_i^{server}$ with VMs in $\mathbb{V}_i^{alloc}$, while satisfying two conditions: (1) $\overline{Cost}_i^{server}$ should be larger than TH_{cost}; and (2) $\hat{u}_{cpu}(VM_i)$ should be less than or equal to Rem_i (line 11)

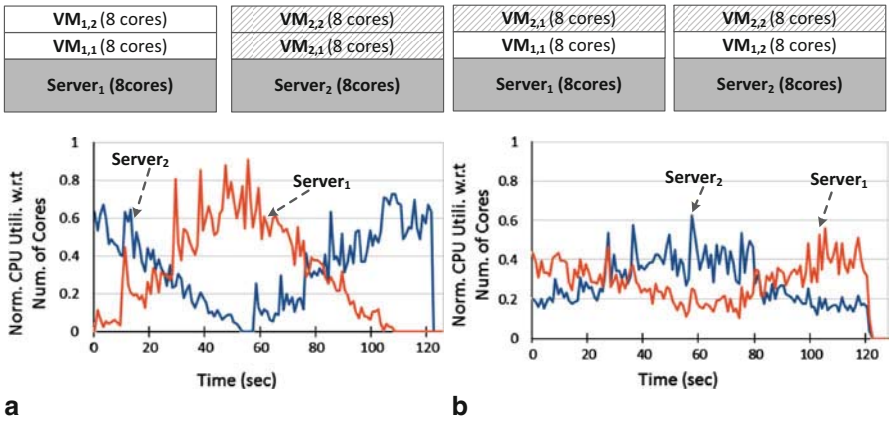

Fig. 25 CPU utilization traces: correlation-unaware (**a**) and correlation-aware (**b**) VM placements [56]

- Update $\mathbb{V}_i^{alloc}$, Rem_i, and $\mathbb{V}^{unalloc}$ accordingly in caase we find a VM (lines $12 \sim 15$)
- Iterate the procedure to find VMs to be allocated in $Server_i$ until until there is no VM left (lines $12 \sim 16$).
- If we have unallocated VMs at the end of the iteration, we need to repeat the procedure in lines $10 \sim 16$ with a degenerated TH_{cost} by a factor of α (line 17) along with a list of servers sorted in descending order of Rem_i (line 18)

3) Setting Voltage and Frequency Level Due to the correlation-aware VM allocation, the actual peak server utilization becomes much lower than the server's computing capability. Figure 25 shows the comparisons of CPU utilization traces when we allocate VMs in correlation-unaware and correlation-aware manner, respectively. Websearch benchmark is used in CloudSuite benchmark suite [4] and configured two websearch clusters each of which has two VMs, i.e., ISNs, in a single websearch cluster, and applied cosine and sin wave user request patterns to each cluster. In the figure, $VM_{i,j}$ represents j-th VM in i-th websearch cluster. As shown in Fig. 22b, workloads of VMs in a same cluster are highly correlated. Thus, a correlation-unaware VM placement solution allocates VMs in a same cluster into a same server as shown in Fig. 25a while the correlation-aware solution allocates VMs in different websearch clusters into a same server as shown in Fig. 25b. As illustrated in the figure, the correlation-aware VM placement solution leads to lowered peak CPU utilization, which enables to lower voltage and frequency (v/f) levels for further power savings.

However, we do not know exactly how much we can lower v/f level when multiple VMs are co-located into a server because $Cost_{i,j}^{vm}$ only captures the correlation between two VMs. An empirical solution to provide rough guideline to lower v/f is provided in [56] which utilizes $\overline{Cost_i}^{server}$ in Eq. (31). Figure 26 shows an empirical relationship representing possible v/f slowdown for servers with respect to

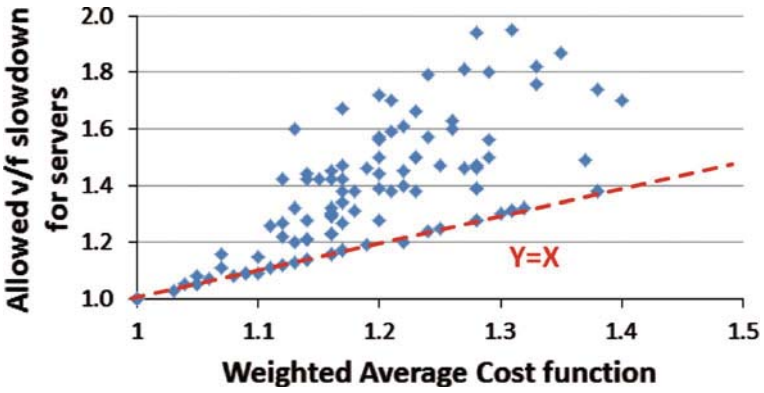

Fig. 26 Relationship between $\overline{Cost_i^{server}}$ in Eq. (31) and possible v/f scaling factor

$\overline{Cost_i^{server}}$. The dots are scattered while the red line, which is a form of $y = x$, shows the lower bound where we can safely lower v/f level without any performance degradation. Based on this relationship, we can determine the frequency level of $Server_i$, i.e., f_i as follows:

$$f_i = \left(\frac{1}{\overline{Cost_i^{server}}} \right) \cdot \left(\frac{\sum_{j=1}^{n_i^{vm}} \hat{u}_{cpu}(VM_{i,j})}{N_{core}^{server}} \right) \cdot f^{max} \tag{33}$$

where f^{max} is the maximum frequency level. f_i is set by lowering the worst-case peak required frequency level (i.e., the second parenthesis assuming the situation when peaks of VMs coincide) with a factor of $1/\overline{Cost_i^{server}}$.

Figure 27 shows 90th percentile response time of websearch benchmark in four different VM placement solutions and v/f levels.

- *Segregated*: allocate VMs into a server such that no VMs share cores
- *Shared-UnCorr*: allocate VMs to share cores without any consideration on their correlation
- *Shared-Corr (2.1G)*: correlation-aware VM allocation while running a server with 2.1 GHz
- *Shared-Corr (1.9G)*: correlation-aware VM allocation while running a server with 1.9 GHz

As shown in Fig. 27, allocating VMs to share cores provides better performance compared to the segregated allocation case. In addition, the correlation-aware VM allocation provides better response time compared to the correlation-unaware allocation scheme as it enables to reduce the actual CPU utilization, thereby the lowered utilization can be used to lower the v/f/ level without any performance degradation compared to the correlation-unaware solution.

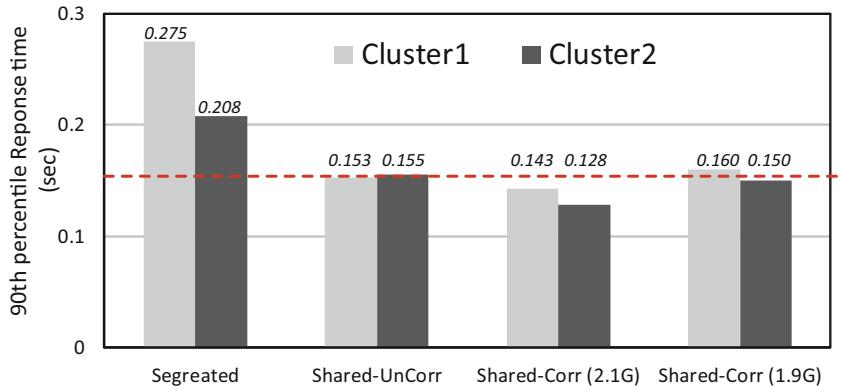

Fig. 27 90th percentile response time of $Cluster_1$ and $Cluster_2$ for three different VM allocations in [56]

4) Simulation Results: Effectiveness of the 3Correlation-Aware VM Placement For further validation of the correlation-aware VM placement, Kim et al. in [56] performed the evaluation using server utilization traces from an actual datacenter setup. It used CPU utilization traces of 40 VMs where each sample is collected at every 5 min for a day while synthesizing fine-grained utilization per 5 s with a lognormal random number generator [7], whose mean is the same as the sampled value for the corresponding 5-min sample. It targeted an Intel Xeon E5410 server configuration which consists of eight cores and two frequency levels (2.0 GHz and 2.3 GHz) and used the power model in [33] to compare the power consumption results among various solutions. It compares three different VM placement approaches as follows.

- *Best-Fit-Decreasing placement (BFD)*: a conventional best-fit-decreasing heuristic approach without taking into account correlation information
- *Peak clustering-based placement (PCP)* [51]: a correlation-aware VM allocation clustering VMs based on the envelopes of VMs' CPU utilization such that VMs coinciding their peaks of the envelopes are not allocated in a same server
- *Correlation-aware placement (CAP)* [56]: a correlation-aware VM allocation considering workload characteristics of scale-out applications manipulating a new correlation metric in Eq. (30).

Table 3a compares the power consumption and performance violations of the three approaches when we statically set the v/f level at the time of VM placement, i.e., $t_{period} = 1$ h. The power consumption results are normalized with respect to the power consumed by *BFD*, and the maximum violation shows the maximum per-period ratio of the number of over-utilized time instances (i.e., when the aggregated utilization among co-located VMs is beyond the CPU capacity of a corresponding server) to t_{period}, during the entire periods, i.e., 24 h. *CAP* provides up to 13.7 % power savings compared to *BFD* and *PCP*, while drastically reducing the number of the violations. It is noteworthy that *PCP* provides almost similar results with *BFD*

Table 3 Comparisons for static (**a**) and dynamic (**b**) v/f scaling

(a)

	Normalized power	Maximum violations (%)
BFD	1	18.2
PCP [51]	0.999	18.2
CAP	*0.863*	*2.6*

(b)

BFD	1	20.3
PCP	0.997	20.3
CAP	*0.958*	*3.1*

because, due to high and fast-changing correlations among VMs in our utilization traces, *PCP* classifies VMs into only '1' cluster during the most of the time periods (22 out of 24 time periods). When the number of clusters is '1', *PCP* behaves exactly same with *BFD*. The power savings obtained by our proposed solution are due to the aggressive-yet-safe v/f settings utilizing the lowered actual peak resource demand, i.e., Eq. (33). Moreover, the proposed solution provides a drastic reduction of the violations (i.e., 15.6 %) compared to the other approaches. Note that we allocated VMs based on their peak utilizations, which were predicted from the their history. Despite the provision based on the peak utilization, we observed quality degradation over the three approaches due to the mis-predictions of the peak utilization, especially during abrupt workload changes. However, the proposed solution 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.

Table 3b shows the comparisons for the simulated case of servers using dynamic v/f scaling for further investigation of the effectiveness of *CAP*. To prevent frequent oscillations of v/f level (which affects server reliability [70]), we performed the v/f scaling at every 12 samples (i.e., 1 min). As shown in Table 3b, the power savings become smaller compared to the static v/f scaling because the other approaches also adaptively scale v/f level according to the time-varying utilization demand. However, the amount of the violations is unacceptably high in the other approaches. Thus, more servers need to be activated to achieve the same QoS level obtained by the proposed solution, which leads to higher power consumption.

6 Power Minimization of Datacenters with Hybrid Cooling Architectures

The power consumption of datacenter can be further optimized as we jointly reduce the computing and cooling power consumption because the conventional computing power minimization solutions discussed in Sect. 5 usually require higher cooling capability due to the increased heat density of active servers by increasing actual

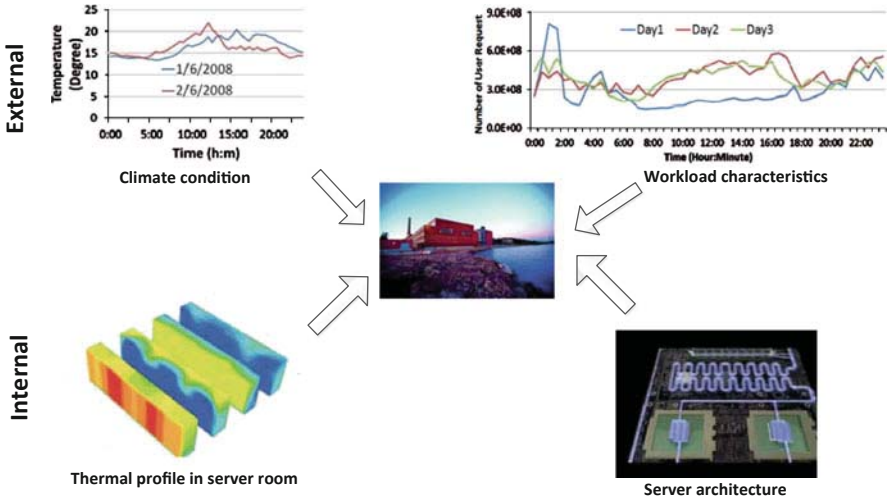

Fig. 28 Proposed solution overview [43]

CPU utilization. Especially, when it comes along with hybrid cooling solutions in a datacenter [26–28, 59–58], explained in Sect. 2.2, we need to revisit existing VM placement solutions [60–65] as it further reduces the chance of using free cooling as the solutions requires higher cooling capability due to the higher operating temperature of active servers. Motivated by this observation, Kim et al. present a joint power and thermal optimization solution for datacenters equipped with hybrid cooling architecutre to achieve further power savings while satisfying service-level agreement (SLA) requirements by extending the usability of free cooling for datacenters having a hybrid cooling architecture [43]. Figure 28 illustrates the solution overview explained in this section. The solution largely takes into account four input parameters as follows:

- Climate condition
- Workload characteristics
- Temperature profile in a server room
- Server cooling architecture

As the climate condition and workload characteristics are non-deterministic, the solution is implemented using a predictive control scheme utilizing predictions of the values. The temperature profile of a server room and the dependency between the server temperature and cooling solutions can be modeled using the solutions explained in Sect. 3.

6.1 Formal Problem Definition

To jointly minimize the computing and cooling power consumption of a datacenter equipped with hybrid cooling architecture, we need to determine the optimal pair of cooling mode, m_{co} (electrical vs. free cooling) and maximum power consumption of active servers (namely, *power capping*) based on four input parameters. In addition, switching cooling mode, i.e., turning on and off chillers, leads to overhead in terms of power and time. Thus, we jointly minimize the number of cooling mode switches along with the power consumption by judiciously considering the switching overhead into the objective function. Based on the requirements, the problem can be formulated as follows:

$$\text{Find} \quad \chi = \{m_{co}, [b_{i,j}]_{N_{pm} \times N_{vm}}\} \tag{34}$$

$$\text{Minimize} \quad J_{dc} = P_{cl} + P_{co} + O_{tr} \tag{35}$$

$$\text{Subject to} \quad T_{pm_i} \leq T_{pm}^{max}, \text{where } 1 \leq i \leq N_{pm} \tag{36}$$

$$Pr(t_{act} > t_{req}) \leq (1 - \beta) \tag{37}$$

The problem we are trying to tackle is two-fold, namely, determining both the (1) cooling mode and (2) VM placement such that the power consumption of datacenter, i.e., $P_{dc} = P_{cl} + P_{co}$ where P_{cl} and P_{co} represent the computing and cooling power consumption in a datacenter, and the overhead caused by cooling mode transition, i.e., O_{tr}, are jointly minimized while satisfying temperature and SLA requirements. m_{co} represents datacenter cooling mode: '1' when free cooling is selected, otherwise '0'; $b_{i,j}$ is a binary variable representing VM placement: '1' when vm_j is mapped into pm_i; N_{pm} and N_{vm} represent the number of servers and VMs, respectively; J_{dc} is an objective function consisting of power consumption of datacenter, i.e., $P_{dc} = P_{cl} + P_{co}$, and overhead caused by switching cooling mode, i.e., O_{tr}; T_{pm_i} and T_{pm}^{max} represent temperature of i-th server (or physical machine) and the maximum temperature constraint of servers, respectively. Then, t_{act} and t_{req} are actual and required execution time, respectively, and $Pr(t_{act} > t_{req})$ represents the probability when t_{act} is larger than t_{req}; β is SLA requirement.

As a matter of fact, this optimization problem can be translated into a bin-packing problem with variable bin size by exploiting the analogy between a bin and a server because, for a given bin size (analogy with threshold of server utilization), the power consumption is minimized when the number of bins (analogy with the number of active servers in which VMs are assigned) is minimized, i.e., server consolidation. Hence, the bin size, i.e., the threshold of server utilization, depends on m_{co} as well as T_{out}. However, due to the interdependency between m_{co} and $b_{i,j}$'s, the solution complexity is even higher than conventional bin-packing problem.

To reduce the solution complexity, we can solve this problem with a two-phase solution. First, we determine a power-optimal pair of $\{m_{co}, u_{pm}^{th}\}$ such that J_{dc} is minimized while satisfying temperature and performance requirements assuming

that ideal VM consolidation[3] is applied, i.e., utilization of every active server equals to u_{pm}^{th} while others are '0'. Second, we assign VMs to servers such that the number of servers where VMs are allocated is minimized while total utilization of every server does not exceed u_{pm}^{th}. Moreover, in order to achieve further improvement by considering time-varying characteristics of T_{out} and the user requests, we iterate the optimization procedure at every predefined time interval, t_{opt}. Note that we can reuse server consolidation techniques explained in Sect. 5. Therefore, in this section, we simply focus on explaining the first step of this problem.

6.2 Multi-objective Trade-offs Exploration Between Cooling Mode and Utilization Threshold

We explore the best approach to determine the optimal pair of $\{m_{co}, u_{pm}^{th}\}$ which minimizes the multi-objective function, J_{dc}. Since external conditions, i.e., outside temperature and user requests, are time-varying, the optimal pair of $\{m_{co}, u_{pm}^{th}\}$ varies as well. Thus, we periodically adjust $\{m_{co}, u_{pm}^{th}\}$ based on the predictions of the external conditions and the predictive sequence of cooling mode transition. Assuming the ideal VM consolidation at a certain instant, we can approximate the problem as follows:

$$\text{Find} \quad \chi(k) = \{m_{co}(k), u_{pm}^{th}(k)\} \tag{38}$$

$$\text{Min} \quad J_{dc}(k) = \sum_{l=k}^{k+N_h-1} \alpha^{l-k}\left(\widetilde{P}_{cl}(l) + \widetilde{P}_{co}(l) + \widetilde{O}_{tr}(l)\right) \tag{39}$$

$$\text{s.t} \quad u_{pm}^{th}(l) \geq \frac{\hat{U}_{tot}(l)}{N_{pm}}, \forall l \in [k, k+N_h-1] \tag{40}$$

$$u_{pm}^{th}(l) \leq min\left(u_{pm}^{max}, u_{pm}^{temp,max}(l)\right), \forall l \tag{41}$$

where N_h is the number of time periods; α is a weighting factor, $0 \leq \alpha \leq 1$; $\widetilde{P}_{cl}(l)$, $\widetilde{P}_{co}(l)$, and $\widetilde{O}_{tr}(l)$ are predictions of P_{cl}, P_{co}, and O_{tr} at the l-th period, which are expressed as follows:

$$\widetilde{P}_{cl}(l) = \sum_{mode \in \{act,idle,sleep\}} \widetilde{N}_{pm}^{mode}(l)\widetilde{P}_{pm}^{mode}(l) \tag{42}$$

$$\widetilde{P}_{co}(l) = (PUE(u_{pm}^{th}(l)) - 1) \cdot \widetilde{P}_{cl}(l) \tag{43}$$

$$\widetilde{O}_{tr}(l) = w_{tr}^{co} \cdot (m_{co}(l) - m_{co}(l-1))^2 \tag{44}$$

[3] In order to reduce the solution complexity, we find the solution assuming that the ideal VM consolidation. The approach is optimistic in that the estimated power consumption is lower than actual scenario due to the fragmentation of the server utilization caused by different utilizations among VMs and fractional ratio of the obtained server utilization to VM utilization in actual scenario.

where $\widetilde{P}_{pm}^{mode}(l)$ is the estimated average power consumption of server at the l-th period when the operating mode of the server is active (i.e., $u_{pm} = u_{pm}^{th}(k)$ based on the assumption of ideal VM consolidation), idle, and sleep modes. $\widetilde{N}_{pm}^{mode}(l)$ is the corresponding number of servers. PUE is obtained using Eq. (14). $(m_{co}(l) - m_{co}(l-1))^2$ represents whether cooling mode is switched at the l-th period, and w_{tr}^{co} is a weighting factor which models the overhead caused by cooling mode transition. $\widetilde{N}_{pm}^{act}(l)$, $\widetilde{N}_{pm}^{idle}(l)$, and $\widetilde{N}_{pm}^{sleep}(l)$ are defined as follows:

$$\widetilde{N}_{pm}^{act}(l) = \frac{\widetilde{U}_{tot}(l)}{u_{pm}^{th}(l)} \tag{45}$$

$$\widetilde{N}_{pm}^{idle}(l) = \frac{\hat{U}_{tot}(l)}{u_{pm}^{th}(l)} - \widetilde{N}_{pm}^{act}(l) \tag{46}$$

$$\widetilde{N}_{pm}^{sleep}(l) = N_{pm} - (\widetilde{N}_{pm}^{act}(l) + \widetilde{N}_{pm}^{idle}(l)) \tag{47}$$

where N_{pm} is the number of servers; $\widetilde{U}_{tot}(l)$ is the prediction of average user requests normalized with respect to the maximum number of user requests processed by single server, i.e., $0 \leq \widetilde{U}_{tot}(l) \leq N_{pm}$; $\hat{U}_{tot}(l)$ is the normalized maximum[4] user requests which is characterized a priori based on extensive characterization.

The first constraint (Eq. (40)) represents the lower bound of $u_{pm}^{th}(l)$ which is determined such that $\hat{U}_{tot}(l)$ user requests can be processed while satisfying SLA requirement. The second constraint (Eq. (41)) represents the upper bound of $u_{pm}^{th}(l)$, which is determined by the minimum value between the utilization level where multiple VMs can run in a single server without acceptable performance loss (i.e., u_{pm}^{max}) and the highest utilization satisfying maximum temperature constraint, i.e., $u_{pm}^{temp,max}(l)$ which is obtained from temperature models in Sect. 3.1.

At the start of k-th period, we solve the optimization problem with two steps: (1) prediction of the external condition, i.e., $\widetilde{U}_{tot}$ and T_{out} for $[k, k + N_h - 1]$-th periods and (2) optimization to find $\{m_{co}(k), u_{pm}^{th}(k)\}$.

1) Temperature and Workload Prediction At the start of k-th period, we predict $T_{out}(l)$ and $\widetilde{U}_{tot}(l)$ where $k \leq l \leq (k + N_h - 1)$. Prediction of T_{out}'s can accurately be predicted by daily and weekly weather forecast. However, accurate prediction of $\widetilde{U}_{tot}$'s is not trivial due to uncertain and non-stationary characteristics of user requests. For accurate prediction, we adopt non-stationary Kalman filter [66], which outperforms other predictors especially when a prediction value is uncertain and non-stationary.

$\widetilde{U}_{tot}(k)$ is predicted based on the history of measured U_{tot} in past few periods as well as the history of the same period in past few days (or weeks). The predictions

[4] In this work, we target the SLA violation to be less than 5 %. Thus, we used 95th-percentile value instead of the maximum value to characterize the worst-case behavior of the corresponding period. Considering the correlation among VMs, we can use lower percentile values, e.g., 90-, 80-th percentile, etc., to reduce more power consumption while satisfying SLA requirement, as presented in [51]. Our optimization approach is directly applicable to these cases as well.

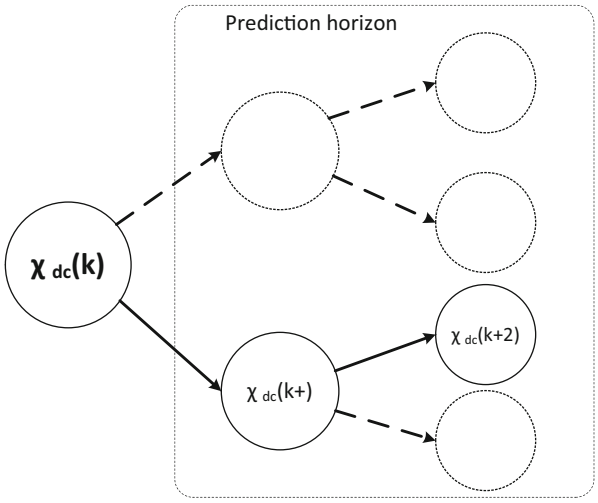

Fig. 29 An example of the predictive control scheme when $N_h = 3$

obtained from the former history is denoted as $\widetilde{U}_{tot}^{(1)}(k)$ while the other is denoted as $\widetilde{U}_{tot}^{(2)}(k)$. Then, we can obtain $\widetilde{U}_{tot}(k)$ by a weighted sum of $\widetilde{U}_{tot}^{(1)}(k)$ and $\widetilde{U}_{tot}^{(2)}(k)$ as follows:

$$\widetilde{U}_{tot}(k) = w_p^{(1)}\widetilde{U}_{tot}^{(1)}(k) + (1 - w_p^{(1)})\widetilde{U}_{tot}^{(2)}(k) \tag{48}$$

where weight, $w_p^{(i)}(k)$ is weight factor.

2) Predictive Control Scheme To solve the multi-objective problem considering the uncertainty of T_{out} and $\widetilde{U}_{tot}$, we adopt receding horizon control scheme as shown in Fig. 29. At the start of the k-th period, we first predict $\widetilde{U}_{tot}$'s and T_{out}'s for $[k, k + N_h - 1]$-th periods as explained in Sect. 6.2. Second, we find the optimal utilization threshold corresponding to each cooling mode, i.e., $m_{co} = \{0, 1\}$, for $[k, k + N_h - 1]$-th periods, as follows. For a given cooling mode, we can express $\widetilde{P}_{dc}(k) = \widetilde{P}_{dc}(k) + \widetilde{P}_{cl}(k)$ as a continuous form with respect to $u_{pm}^{th}(k)$ using Eqs. (42)–(47). In addition, $\widetilde{P}_{dc}(k)$ is convex with respect to $u_{pm}^{th}(k)$ because, as $u_{pm}^{th}(k)$ increases, $\widetilde{P}_{cl}(k)$ is monotonically decreased (due to the decreased number of active servers) while $\widetilde{P}_{co}(k)$ increases because PUE is monotonically increased. Figure 30 shows the relationship of the power consumption with respect to the u_{pm}^{th}. When an electrical cooling is used, we can find an inflection point as the computing and the cooling power consumptions varies in opposite directions. When a free cooling is used, the total power consumption is usually decreased as the decrease of the computing power as u_{th}^{pm} increases is much larger than the increase of the cooling power. However, the cooling capability of the free cooling is limited, thereby, u_{th}^{pm} cannot be set too high.

Owing to the continuity and convexity of $\widetilde{P}_{dc}(k)$ with respect to $u_{pm}^{th}(k)$ for given $m_{co}(k)$, the unconstrained optimal solution of $u_{pm}^{th}(k)$ can be obtained by finding

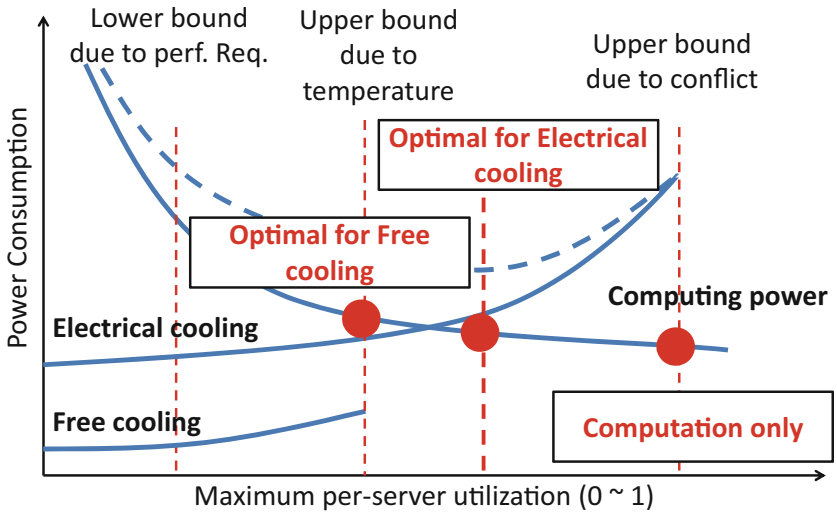

Fig. 30 Solution overview

value which satisfies following equation.

$$\text{Find } u_{pm}^{th}(k) \Longrightarrow \frac{\partial\left(P_{cl}(k) + P_{co}(k)\right)}{\partial u_{pm}^{th}(k)} = 0 \tag{49}$$

The root can be efficiently obtained by root-finding algorithms, e.g., Newton-Raphson method, binary search, etc. [67]. When $u_{pm}^{th}(k)$ satisfies the constraint, we directly set utilization threshold with $u_{pm}^{th}(k)$; otherwise, we set $u_{pm}^{th}(k)$ with lower-bound (Eq. (40)) and upper-bound (Eq. (41)) values so as to satisfy the constraint.

Third, with the pairs of $\{m_{co}, u_{pm}^{th}\}$'s and including the overhead caused by cooling mode transition, i.e., O_{tr}, we find the optimal sequence of cooling mode transition from k-th to $(k + N_h - 1)$-th periods, i.e., $\chi_{dc}(k) \rightarrow \chi_{dc}(k+1|k) \rightarrow \cdots \rightarrow \chi_{dc}(k + N_h - 1|k)$ where $\chi_{dc}(k + l|k)$ is the optimal solution at $(k + l)$-th period when $\chi_{dc}(k)$ is determined as the optimal solution at k-th period. Then, we select only $\chi_{dc}(k)$ and discard the other steps of the sequence. Finally, the entire process is repeated at the start of $(k + 1)$-th period with the updated predictions.

The complexity is $O(N_{pm}^{N_h-1})$. Despite the exponential complexity, the solution can normally be found in low overhead by confining the search range to the proximity of previous N_{pm}'s.

Table 4 Comparisons of power consumption and number of cooling mode transitions in May, June, and July

Period	FIXED-TEMP	P-ADAPTIVE	PT-ADAPTIVE
May 1 ∼ May 4	1.00 / 7	0.781 / 8	0.784 / 6
June 1 ∼ June 4	1.00 / 0	0.738 / 8	0.743 / 4
July 1 ∼ July 4	1.00 / 0	0.879 / 29	0.898 / 13

6.3 Simulation Results

To evaluate the effectiveness of the joint optimization, we used CloudSim [68], an event-driven simulator providing toolkits to model behavior of cloud system components such as datacenters, virtual machines (VMs), and scheduling policies. We configured the target system with 100 servers and 100 VMs and used temperature data measured at EPFL in Lausanne, Switzerland from May 2008 to July 2008. To account for the overhead caused by VM migration, we assumed 100 s and 10 % as the migration time and performance degrdation, respectively. Then, we compared the following cooling mode decision solutions for datacenters:

- **FIXED-TEMP**: a conventional cooling mode decision scheme which uses free cooling only when T_{out} is lower than fixed pre-defined temperature, i.e., $T_{th} = 10\,^{\circ}\mathrm{C}$ [28], and sets u_{pm}^{th} to u_{pm}^{max}.
- **P-ADAPTIVE**: this is our first proposed scheme which adaptively adjusts the cooling mode and the utilization threshold such that only power consumption of datacenter is minimized.
- **PT-ADAPTIVE**: this is our second proposed scheme which jointly optimizes the power consumption and transition overhead caused by cooling mode transition with receding horizon control scheme.

To simply evaluate the effectiveness of the joint cooling mode decision scheme, we applied the same VM allocation solution based on the peak [46] for all the three comparisons above. Remind that these solutions are complementary with existing VM allocation and power management solutions.

Table 4 shows the comparisons in terms of power consumption and number of cooling mode transitions during the first four days in May, Jun, and July. The first column represents the simulated time period. The second to fourth columns show the normalized power consumption with respect to FIXED-TEMP and the number of cooling mode transitions in each month.

First, in May, PT-ADAPTIVE yields 21.6 % power savings compared to FIXED-TEMP. The reason for the improvement can be analyzed by observing the traces of cooling mode and utilization schedules presented in Fig. 31 where a and b depict the traces for FIXED-TEMP and PT-ADAPTIVE, respectively. X-axis represent data (month/date) and the left and right Y-axis are cooling mode/utilization and outside temperature, respectively. The temperature ranges $7 \sim 22\,^{\circ}\mathrm{C}$ in May, thereby FIXED-TEMP uses the free cooling for short time period only when the outside

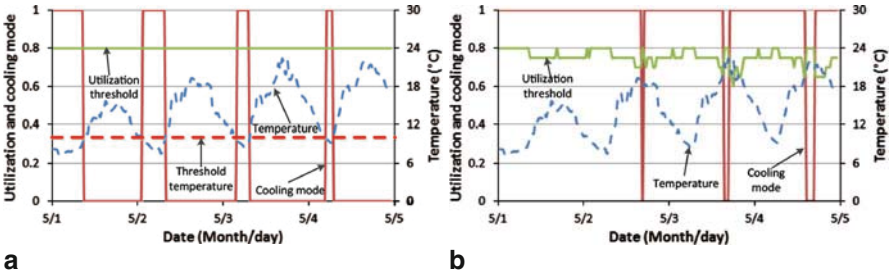

Fig. 31 Schedule of free mode and utilization threshold in May: MAX-UTIL (**a**) and PT-ADAPTIVE(**b**)

Table 5 Comparisons of power consumption and the number of cooling mode transitions as P_{static}/P_{tot} changes in June

P_{static}/P_{tot}	FIXED-TEMP	P-ADAPTIVE	PT-ADAPTIVE
0.3	1.00 / 0	0.722 / 2	0.722 / 2
0.5	1.00 / 0	0.738 / 8	0.743 / 4
0.7	1.00 / 0	0.852 / 24	0.878 /12

temperature is lower than the threshold value, i.e., 10 °C in this evaluation. On the contrary, PT-ADAPTIVE enables to use the free cooling for the longer time period as it dynamically adjusts the maximum power consumption of servers by capping the maximum server utilization according to the amount of demanding workload and the outside temperature.

The most highest power savings are observed in June, i.e., 25.7 %. The reason is that the outside temperature is always higher than 10 °C, which makes impossible to use the free cooling in FIXED-TEMP while PT-ADAPTIVE still decides to use the free cooling by lowering the maximum server power consumption. However, in July, the temperature is too high to use the free cooling while meeting the performance requirements despite capping the maximum server power consumption, which leads to rather smaller power savings, i.e., 10.2 %, compared to other months.

Compared to P-ADAPTIVE, PT-ADAPTIVE provides almost similar (or slightly less) power savings. However, PT-ADAPTIVE schedules the cooling mode such that the number of cooling mode transitions is drastically reduced by accounting for the overhead caused by the cooling mode transitions. Especially, in July, P-ADAPTIVE switches the cooling mode too often, i.e., around 7 times per day while PT-ADAPTIVE can reduce the number of transitions down to 3.25 times per day. Figure 32a and b show the traces of P-ADAPTIVE and PT-ADAPTIVE in July, respectively.

One important observation is that the effectiveness of PT-ADAPTIVE gets enhanced as the energy proportionality of server becomes improvement, which is the direction where server designers are now focusing on. Table 5 shows the normalized power consumption in June as the power-proportionality of servers, defined as the

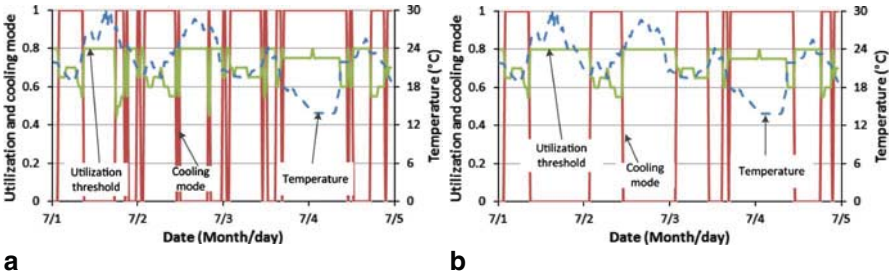

Fig. 32 Schedule of free mode and utilization threshold in July: P-ADAPTIVE (**a**) and PT-ADAPTIVE(**b**)

ratio of the static power to the total power consumption, i.e., P_{static}/P_{tot}, is at 0.3, 0.5, and 0.7. As shown in Table 5, PT-ADAPTIVE provides more power savings as P_{static}/P_{tot} is lowered. As a matter of fact, when P_{static}/P_{tot} is low, we can use free cooling for longer periods of time by lowering the server utilization threshold, thereby we have a smaller number of active servers. Furthermore, as state-of-the-art servers are designed to achieve higher energy-proportionality [69], these experiments demonstrate that PT-ADAPTIVE is able to provide even more power savings in possible future datacenter setups.

7 Conclusions

Recently, the energy-efficiency constraints have become the dominant limiting factor for datacenters due to their unprecedented increase of growing size and electrical power demands. In this chapter, we have explained the power and thermal modeling and control solutions which can play a key role to reduce the power consumption of datacenters considering time-varying workload characteristics while maintaining the performance requirements and the maximum temperature constraints. We have first explained simple-yet-accurate power and temperature models for computing servers, and then, extended the model to cover computing servers and cooling infrastructure of datacenters. Second, we have presented the power and thermal management solutions for servers manipulating various control knobs such as voltage and frequency of servers, workload allocation, and even cooling capability, especially, flow rate of liquid cooled servers). Finally, we have presented the solution to minimize the server clusters of datacenters by proposing a solution which judiciously allocates virtual machines to servers considering their correlation, and then, the joint optimization solution which enables to minimize the total energy consumption of datacenters with hybrid cooling architecture (including the computing servers and the cooling infrastructure of datacenters).

Acknowledgment This work has been partially supported by the Nano-Tera.ch TRANSCEND Strategic Action, the PMSM: CT Monitoring research grant for ESL-EPFL funded by Credit Suisse AG, an ERO Research Grant Donation from Oracle for ESL-EPFL, and the EC FP7 GreenDataNet STREP project (agreement No. 609000).

References

1. K. G. Brill, "The invisible crisis in the data center: The economic meltdown of Moore's law," *white paper, Uptime Institute*, 2007.
2. Energy Star Program, "EDA Report to Congress on Server and Data Center Energy Efficiency," 2007.
3. L. A. Barroso and U. Holzle. "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures on Computer Architecture* 4, no. 1 (2009): 1–108.
4. 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," in *ACM SIGARCH Computer Architecture News*, vol. 40, no. 1, pp. 37–48. ACM, 2012.
5. A. Adileh, P. Lotfi-Kamran, S. Volos, S. Volos, and C. Kaynak, "CloudSuite on Flexus tutorial," in *international symposium on computer architecture (ISCA)* 2012.
6. D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pp. 319–330. IEEE, 2011.
7. T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Computer Communication Review 40*, no. 1 (2010): 92–99.
8. H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pp. 750–757. IEEE, 2012.
9. 42U Datacenter Efficiency Consulting Corporation, "Data Center Energy Efficiency Calculator," http://www.42u.com/efficiency/energy-efficiency-calculator.htm, 2011.
10. E. Schurman and J. Brutlag, "The user and business impact of server delays, additional bytes, and HTTP chunking in web search," in *Presentation at the OReilly Velocity Web Performance and Operations Conference*, 2009.
11. G. H. Loh and Y. Xie, "3D stacked microprocessor: Are we there yet?," *Micro, IEEE 30*, no. 3 (2010): 60–64.
12. HP DL980, [online available] http://h18000.www1.hp.com/products/servers/platforms/ .
13. Eurocloud, [online avalable] http://www.eurocloudserver.com/ .
14. D. Meisner and T. F. Wenisch, "Does low-power design imply energy efficiency for data centers?," in *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*, pp. 109–114. IEEE Press, 2011.
15. A. Coskun, J. Meng, D. Atienza, and M. M. Sabry, "Attaining single-chip, high-performance computing through 3D systems with active cooling," *Micro, IEEE 31*, no. 4 (2011): 63–75.
16. U. S. Deparment of Energy, "FEMP Best Practices Guide for Energy-Efficient Data Center Design," in 2011.
17. A. N. Nowroz, R. Cochran, and S. Reda, "Thermal monitoring of real processors: Techniques for sensor allocation and full characterization," in *Proceedings of the 47th Design Automation Conference*, pp. 56–61. ACM, 2010.
18. H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu, "Memory power management via dynamic voltage/frequency scaling," in *Proceedings of the 8th ACM international conference on Autonomic computing*, pp. 31–40. ACM, 2011.

19. R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," in *ACM SIGARCH Computer Architecture News*, vol. 36, no. 1, pp. 48–59. ACM, 2008.
20. X. Wang and Y. Wang, "Coordinating power control and performance management for virtualized server clusters," *Parallel and Distributed Systems, IEEE Transactions on 22*, no. 2 (2011): 245–259.
21. R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. Martins, A. V. Anderson, S. M. Bennett, A. Kagi, F. H. Leung, and L. Smith, "Intel virtualization technology," *Computer 38*, no. 5 (2005): 48–56.
22. P. Muditha Perera and C. Keppitiyagama, "A performance comparison of hypervisors," in *Advances in ICT for Emerging Regions (ICTer), 2011 International Conference on*, pp. 120–120. IEEE, 2011.
23. N. Huber, M. Quast, M. Hauck, and S. Kounev, "Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments," in *CLOSER*, pp. 563–573. 2011.
24. CoolDoor, [online available] http://www.cooldoor.com.au/html/specifications.html .
25. M. Pawlish and A. S. Varde, "Free cooling: A paradigm shift in data centers," in *Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on*, pp. 347–352. IEEE, 2010.
26. D. Garday, "Reducing data center energy consumption with wet side economizers," *White paper, Intel* (2007).
27. D. Atwood and J. G. Miner, "Reducing data center cost with an air economizer," *White Paper: Intel Corporation* (2008).
28. T. Lu, X. Lu, M. Remes, and M. Viljanen, "Investigation of air management and energy performance in a data center in Finland: Case study," *Energy and Buildings 43*, no. 12 (2011): 3360–3372.
29. D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob,"DRAMsim: a memory system simulator," *ACM SIGARCH Computer Architecture News 33*, no. 4 (2005): 100–107.
30. Micron's system power calculators, [online available] http://www.micron.com/products/support/power-calc.
31. D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *Proceedings of Workshop on Modeling, Benchmarking, and Simulation*, pp. 70–77. 2006.
32. S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A Comparison of High-Level Full-System Power Models," *HotPower* 8 (2008): 3–3.
33. M. Pedram and I, Hwang,"Power and performance modeling in a virtualized server system," in *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pp. 520–526. IEEE, 2010.
34. M. K. Patterson, "The effect of data center temperature on energy efficiency," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, pp. 1167–1174. IEEE, 2008.
35. J. Choi, Y. Kim, A. Sivasubramanjam, J. Srebric, Q. Wang, and J. Lee, "A CFD-based tool for studying temperature in rack-mounted servers," *Computers, IEEE Transactions on 57*, no. 8 (2008): 1129–1142.
36. T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini, "Mercury and freon: temperature emulation and management for server systems," in *ACM SIGARCH Computer Architecture News*, vol. 34, no. 5, pp. 106–116. ACM, 2006.
37. W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 14*, no. 5 (2006): 501–513.
38. R. Ayoub, R. Nath, and T. Rosing, "JETC: Joint energy thermal and cooling management for memory and CPU subsystems in servers," in *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, pp. 1–12. IEEE, 2012.

39. E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, pp. 145–150. ACM, 2009.
40. D. C. Hwang., V. P. Manno, M. Hodes, and G. J. Chan, "Energy savings achievable through liquid cooling: A rack level case study," in *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2010 12th IEEE Intersociety Conference on*, pp. 1–9. IEEE, 2010.
41. T. J. Breen, E. J. Walsh, J. Punch, A. J. Shah, and C. E. Bash, "From chip to cooling tower data center modeling: Part I influence of server inlet temperature and temperature rise across cabinet," in *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2010 12th IEEE Intersociety Conference on*, pp. 1–10. IEEE, 2010.
42. A. Qouneh, C Li, and T. Li. "A quantitative analysis of cooling power in container-based data centers," in *Workload Characterization (IISWC), 2011 IEEE International Symposium on*, pp. 61–71. IEEE, 2011.
43. J. Kim, M. Ruggiero, and D. Atienza, "Free cooling-aware dynamic power management for green datacenters," in *High Performance Computing and Simulation (HPCS), 2012 International Conference on*, pp. 140–146. IEEE, 2012.
44. Smart data center energy monitoring: a thermal-aware design approach to 'Green IT', http://esl.epfl.ch/cms/op/edit/lang/en/pid/57400
45. Credit Suisse, https://www.credit-suisse.com/
46. E. Pakbaznia, *et al.*, "Minimizing data center cooling and server power costs," in *Proc. ISLPED*, 2009.
47. N. Bobroff, *et al.*, "Dynamic placement of virtual machines for managing sla violations," in *Proc. IM* 2007.
48. P. Padala, X. Zhu, Z.i Wang, S. Singhal, and K. G. Shin. "Performance evaluation of virtualization technologies for server consolidation," in *HP Labs Tec. Report*, 2007.
49. O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, "Modeling virtual machine performance: challenges and approaches," in *ACM SIGMETRICS Performance Evaluation Review 37*, 2010.
50. S. Govindan, J. Liu, A. Kansal, and A. Sivasubramaniam. "Cuanta: quantifying effects of shared on-chip resource interference for consolidated virtual machines," in Proceedings of the 2nd ACM Symposium on Cloud Computing, p. 22. ACM, 2011.
51. A. Verma, *et al.*, "Server workload analysis for powr minimization using consolidation," in *Proc. USENIX*, 2009.
52. X. Meng, *et al.*, "Efficient resource provisioning in compute clouds via VM multiplexign," in *Proc. ICAC*, 2010.
53. M. Chen, *et al.*, "Effective VM sizing in virtualized data centers," in *Proc. IM*, 2011.
54. K. Halder, *et al.*, "Risk aware provisioning and resource aggregation based consolidation of virtual machines," in *Proc. Cloud*, 2012.
55. A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel, "Diagnosing performance overheads in the xen virtual machine environment," in *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, pp. 13–23. ACM, 2005.
56. J. Kim, M. Ruggiero, D. Atienza, and M. Lederberger, "Correlation-aware virtual machine allocation for energy-efficient datacenters," in Proc. *Conference on Design, Automation and Test in Europe (DATE)*, pp. 1345–1350, 2013.
57. M. K. Patterson, D. Atwood, and J. G. Miner, "Evaluation of air-side economizer use in a compute-intensive data center," *ASME*, 2009.
58. M. Pervila and J. Kangasharju,"Running servers around zero degrees," *ACM SIGCOMM Computer Communication Review 41*, no. 1 (2011): 96–101.
59. "Google data center," http://www.google.cim/about/datacenters/#.
60. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review 37*, no. 5 (2003): 164–177.

61. C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation-Volume 2*, pp. 273–286. USENIX Association, 2005.
62. D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster computing 12*, no. 1 (2009): 1–15.
63. G. Dhiman, G. Marchetti, and T. Rosing, "vGreen: a system for energy efficient computing in virtualized environments," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, pp. 243–248. ACM, 2009.
64. J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on and Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pp. 179–188. IEEE, 2010.
65. J.-W. Jang, M. Jeon, H.-S. Kim, H. Jo, J.-S. Kim, and S.l Maeng, "Energy reduction in consolidated servers through memory-aware virtual machine scheduling," *Computers, IEEE Transactions on 60*, no. 4 (2011): 552–564.
66. S.-Y. Bang, K. Bang, S. Yoon, and E.-Y. Chung, "Run-time adaptive workload estimation for dynamic voltage scaling," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 28*, no. 9 (2009): 1334–1347.
67. K. Madsen, "A root-finding algorithm based on Newton's method," *BIT Numerical Mathematics 13*, no. 1 (1973): 71–75.
68. R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *High Performance Computing and Simulation, 2009. HPCS'09. International Conference on*, pp. 1–11. IEEE, 2009.
69. D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: eliminating server idle power," in *ACM Sigplan Notices*, vol. 44, no. 3, pp. 205–216. ACM, 2009.
70. Y. Guo, D. Zhu, and H. Aydin, "Reliability-aware power management for parallel real-time applications with precedence constraints," in *Green Computing Conference and Workshops (IGCC), 2011 International*, pp. 1–8. IEEE, 2011.
71. J. Kong et al. Recent thermal management techniques for microprocessors. In *ACM Computing Surveys*, 44(3):13:1–13:42, 2012.
72. I. Koren and C. M. Krishna. Temperature-aware computing. In *Sustainable Computing: Informatics and Systems*, 1(1):46–56, 2011.
73. J. Choi et al. Thermal-aware task scheduling at the system software level. In *ISLPED*, 2007.
74. A. K. Coskun, T. Simunic Rosing, and K. Whisnant. Temperature aware task scheduling in MPSoCs. In *DATE*, pages 1659–1664, 2007.
75. J. Donald and M. Martonosi. Techniques for multicore thermal management: Classification and new exploration. In *ISCA*, pages 78–88, 2006.
76. A. K. Coskun et al. Temperature management in multiprocessor socs using online learning. In *DAC*, pages 890–893, 2008.
77. A. K. Coskun et al. Energy-efficient variable-flow liquid cooling in 3D stacked architectures. In *DATE*, pages 111–116, 2010.
78. Festo electric automation technology. http://www.festo-didactic.com/ov3/media/customers/1100/00966360001075223683.pdf.
79. Y. U. Ogras, R. Marculescu, D. Marculescu, and E. G. Jung. Design and management of voltage-frequency island partitioned networks-on-chip. *IEEE Transactions on VLSI*, 17(3):330–341, 2009.
80. P. Bogdan, S. Jian, R. Tornero, and R. Marculescu. An optimal control approach to power management for multi-voltage and frequency islands multiprocessor platforms under highly variable workloads. In *ISNoC*, pages 35–42, 2012.
81. W-L. Hung, Y. Xie, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin. Thermal-aware task allocation and scheduling for embedded systems. In *DATE*, pages 898–899, 2005.

82. A. K. Coskun, T. Simunic Rosing, and K. Gross. Proactive Temperature Balancing for Low-Cost Thermal Management in MPSoCs. In *ICCAD*, pages 250–257, 2008.
83. R. J. Cochran et al. Consistent Runtime Thermal Prediction and Control Through Workload Phase Detection. In *DAC*, pages 62–67, 2010.
84. Y. Zhang et al. Adaptive and Autonomous Thermal Tracking for High Performance Computing Systems. In *DAC*, pages 68–73, 2010.
85. Y. Wang et al. Temperature-constrained power control for chip multiprocessors with online model estimation. In *ISCA*, pages 314–324, 2009.
86. F. Zanini et al. Online Convex Optimization-Based Algorithm For Thermal Management of MPSoCs. In *GLSVLSI*, pages 203–208, 2010.
87. A. Bemporad et al. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 –20, 2002.
88. C. Zhu et al. Three-dimensional chip-multiprocessor run-time thermal management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(8):1479–1492, August 2008.
89. X. Zhou et al. Thermal management for 3D processors via task scheduling. In *ICPP*, pages 115–122, 2008.
90. A. K. Coskun, J. Ayala, D. Atienza, T. Simunic Rosing. Modeling and Dynamic Management of 3D Multicore Systems with Liquid Cooling. In *VLSI-SoC*, pages 60–65, 2009.
91. A. K. Coskun et al. Dynamic thermal management in 3D multicore architectures. In *DATE*, pages 1410–1415, 2009.
92. T. Emi et al. Tape: Thermal-aware agent-based power economy for multi/many-core architectures. In *ICCAD*, pages 302 –309, 2009.
93. H. Qian et al. Cyber-physical thermal management of 3D multi-core cache-processor system with microfluidic cooling. *ASP Journal of Low Power Electronics*, 7(1):1–12, 2011.
94. F. Zanini, M. M. Sabry, D. Atienza, and G. De Micheli. Hierarchical thermal management policy for high-performance 3d systems with liquid cooling. *IEEE JETCAS*, 1(2):88–101, 2011.
95. F. Mulas et al. Thermal balancing policy for multiprocessor stream computing platforms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(12):1870–1882, 2009.
96. M. M. Sabry et al. Energy-Efficient Multi-Objective Thermal Control for Liquid-Cooled 3D Stacked Architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(12):1883–1896, 2011.
97. P. Greenalgh. Big.LITTLE Processing with ARM Cortex-A15 and Cortex-A7. www.arm.com/files/downloads/big.LITTLE_Final.pdf.
98. R. G. Dreslinski et al. Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits. In *Proc. of the IEEE*, 98(2), 2010.
99. N. Xu et al. Thermal-Aware Post Layout Voltage-Island Generation for 3D ICs. In *Journal of Computer Science and Technology*, 28(4):671–681, 2013.
100. K. Puttaswamy and G. H. Loh. Thermal Herding: Microarchitecture Techniques for Controlling Hotspots in High-Performance 3D-Integrated Processors. In *HPCA*, pages 193–204, 2007.
101. Y. Han et al. Temperature Aware Floorplanning. In*Workshop on Temperature Aware Computing Systems*, 2005.
102. K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron. A Case for Thermal-Aware Floorplanning at the Microarchitectural Level. In*Journal of Instruction-Level Parallelism*, 8:1–16, 2005.
103. W-L. Hung et al. Thermal-Aware Floorplanning Using Genetic Algorithms. In*ISQED*, 2005.
104. J. Cong, J. Wei, and Y. Zhang. A Thermal-Driven Floorplanning Algorithm for 3D-ICs. In*ICCAD*, pages 306–313, 2004.
105. W.-L. Hung et al. Interconnect and Thermal-Aware Floorplanning for 3D Microprocessors. In*ISQED*, pages 98–104, 2006.

106. M. Healy et al. Multiobjective Microarchitectural Floorplanning for 2-D and 3-D ICs. In*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(1), 2007.
107. M. Ekpanyapong et al. Thermal-aware 3D Microarchitectural Floorplanning. Georgia Institute of Technology, 2004.
108. H. Mizunuma et al. Thermal Modeling and Analysis for 3D-ICs with Integrated Microchannel Cooling. In*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(9):1293–1306, 2011.
109. M. M. Sabry et al. Greencool: An energy-efficient liquid cooling design technique for 3d mpsocs via channel width modulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(4):524–537, 2013.
110. R. Shah and A. London. *Laminar flow forced convection in ducts*. New York: Academic Press, 1978.
111. Y. Tan et al. Modeling and simulation of the lag effect in a deep reactive ion etching process. *Journal of Micromechanics and Microengineering*, 16, 2006.
112. A. Leon et al. A power-efficient high-throughput 32-thread SPARC processor. *ISSCC*, 42(1):7 – 16, 2007.
113. M. M. Sabry, A. Sridhar, and D. Atienza. Thermal balancing of liquid-cooled 3d-mpsocs using channel modulation. In *DATE*, 2012.