DOI:10.1145/3524284

DANIEL ABADI
ANASTASIA AILAMAKI
DAVID ANDERSEN
PETER BAILIS
MAGDALENA BALAZINSKA
PHILIP A. BERNSTEIN
PETER BONCZ
SURAJIT CHAUDHURI
ALVIN CHEUNG
ANHAI DOAN
LUNA DONG
MICHAEL J. FRANKLIN
JULIANA FREIRE
ALON HALEVY
JOSEPH M. HELLERSTEIN
STRATOS IDREOS
DONALD KOSSMANN
TIM KRASKA
SAILESH KRISHNAMURTHY
VOLKER MARKL
SERGEY MELNIK
TOVA MILO
C. MOHAN
THOMAS NEUMANN
BENG CHIN OOI
FATMA OZCAN
JIGNESH PATEL
ANDREW PAVLO
RALUCA POPA
RAGHU RAMAKRISHNAN
CHRISTOPHER RE
MICHAEL STONEBRAKER
DAN SUCIU

Every five years, a group of the leading database researchers meet to reflect on their community's impact on the computing industry as well as examine current research challenges.

The Seattle Report on Database Research

FROM THE INCEPTION of the field, academic database research has strongly influenced the database industry and vice versa. The database community, both research and industry, has grown substantially over the years. The relational database market alone has revenue upwards of $50B. On the academic front, database researchers continue to be recognized with significant awards. With Michael Stonebraker's Turing Award in 2014, the community can now boast of four Turing Awards and three ACM Systems Software Awards.

Over the last decade, our research community pioneered the use of columnar storage, which is used in all commercial data analytic platforms. Database systems offered as cloud services have witnessed explosive growth. Hybrid transactional/analytical processing (HTAP) systems are now an important

IMAGE BY EMIL TIMPLARU

segment of the industry. Furthermore, memory-optimized data structures, modern compilation, and code-generation have significantly enhanced performance of traditional database engines. All data platforms have embraced SQL-style APIs as the predominant way to query and retrieve data. Database researchers have played an important part in influencing the evolution of streaming data platforms as well as distributed key-value stores. A new generation of data cleaning and data wrangling technology is being actively explored.

These achievements demonstrate that our community is strong. Yet, in technology, the only constant is change. Today's society is a data-driven one, where decisions are increasingly based on insights from data analysis. This societal transformation places us squarely in the center of technology disruptions. It has caused the field to become broader and exposed many new challenges and opportunities for data management research.

In the fall of 2018, the authors of this report met in Seattle to identify especially promising research directions for our field. There is a long tradition of such meetings, which have been held every five years since 1988.[1,3,4,7,8,11–13] This report summarizes findings from the Seattle meeting[2,9] and subsequent discussions, including panels at ACM SIGMOD 2020[6] and VLDB 2020.[5] We begin by reviewing key technology trends that impact our field the most. The central part of the report covers research themes and specific examples of research challenges that meeting participants believe are important for database researchers to pursue, where their unique technical expertise is especially relevant such as cleaning and transforming data to support data science pipelines and disaggregated engine architectures to support multitenant cloud data services. We close by discussing steps the community can take for impact beyond solving technical research challenges.

Unlike database conference proceedings such as ACM SIGMOD and VLDB, this report does not attempt to provide a comprehensive summary of the wide breadth of technical challenges being pursued by database researchers or the many innovations introduced by the industry, for example, confidential computing, cloud security, blockchain technology, or graph databases.

What has Changed for the Database Community in the Last Five Years?

The last report identified big data as our field's central challenge.[1] However, in the last five years, the transformation has accelerated well beyond our projections, in part due to technological breakthroughs in *machine learning* (ML) and artificial intelligence (AI). The barrier to writing ML-based applications has been sharply lowered by widely available programming frameworks, such as TensorFlow and PyTorch, architectural innovations in neural networks leading to BERT and GPT-3, as well as specialized hardware for use in private and public clouds. The database community has a lot to offer ML users given our expertise in data discovery, versioning, cleaning, and integration. These technologies are critical for machine learning to derive *meaningful* insights from data. Given that most of the valuable data assets of enterprises are governed by database systems, it has become imperative to explore how SQL querying functionality is seamlessly integrated with ML. The community is also actively pursuing how ML can be leveraged to improve the database platform itself.

A related development has been the rise of *data science* as a discipline that combines elements of data cleaning and transformation, statistical analysis, data visualization, and ML techniques. Today's world of data science is quite different from the previous generation of statistical and data integration tools. Notebooks have become by far the most popular interactive environment. Our expertise in declarative query languages can enrich the world of data science by making it more accessible to domain experts, especially those without traditional computer science background.

As personal data is increasingly valuable to customize the behavior of applications, society has become more concerned about the state of *data governance* as well as *ethical and fair use of data*. This concern impacts all fields of computer science but is especially important for data platforms, which must enforce such policies as custodians of data. Data governance has also led to the rise of confidential cloud computing whose goal is to enable customers to leverage the cloud to perform computation even though customers keep their data encrypted in the cloud.

Usage of *managed cloud data systems*, in contrast to simply using virtual machines in the cloud, has grown tremendously since our last report observed that "cloud computing has become mainstream."[2] The industry now offers on-demand resources that provide extremely flexible elasticity, popularly referred to as *serverless*. For cloud analytics, the industry has converged on a *data lake architecture*, which uses on-demand elastic compute services to analyze data stored in cloud storage. The elastic compute could be extract, transformation, and load (ETL) jobs on a big data system such as Apache Spark, a traditional SQL data warehousing query engine, or an ML workflow. It operates on cloud storage with the network in-between. This architecture *disaggregates* compute and storage, enabling each to scale independently. These changes have profound implications on how we design future data systems.

Industrial Internet-of-Things (IoT), focusing on domains such as manufacturing, retail, and healthcare, greatly accelerated in the last five years, aided by cheaper sensors, versatile connectivity, cloud data services, and data analytics infrastructure. IoT has further stress-tested our ability to do efficient data processing at the edge, do fast data ingestion from edge devices to cloud data infrastructure, and support data analytics with minimal delay for real-time scenarios such as monitoring.

Finally, there are *significant changes in hardware*. With the end of Dennard scaling[10] and the rise of compute-intensive workloads such as Deep Neural Networks (DNN), a new generation of powerful accelerators leveraging FPGAs, GPUs, and ASICs are now available. The memory hierarchy continues to evolve with the advent of faster SSDs and low-latency NVRAM. Improvements in network bandwidth and latency have been remarkable. These developments point to the need to rethink the hardware-software co-design of the next generation of database engines.

Research Challenges

The changes noted here present new research opportunities and while we have made progress on key challenges in the last report,[2] many of those problems demand more research. Here, we summarize these two sets of research challenges, organized into four subsections. The first part addresses data science where our community can play a major role. The following section focuses on data governance. The last two sections cover cloud data services and the closely related topic of database engines. Advances in ML have influenced the database community's research agenda across the board. Industrial IoT and hardware innovations have influenced cloud architectures and database engines. Thus, ML, IoT, and hardware are three cross-cutting themes and feature in multiple places in the rest of this section.

Data science. The NSF CISE Advisory Council[a] defines data science as "*the processes and systems that enable the extraction of knowledge or insights from data in various forms, either structured or unstructured*." Over the past decade, it has emerged as a major interdisciplinary field and its use drives important decisions in enterprises and discoveries in science.

From a technical standpoint, data science is about the pipeline from raw input data to insights that requires use of data cleaning and transformation, data analytic techniques, and data visualization. In enterprise database systems, there are well-developed tools to move data from OLTP databases to data warehouses and to extract insights from their curated data warehouses by using complex SQL queries, online analytical processing (OLAP), data mining techniques, and statistical software suites. Although many of the challenges in data science are closely related to problems that arise in enterprise data systems, modern data scientists work in a different environment. They heavily use Data Science Notebooks, such as Jupyter, Spark, and Zeppelin, despite their weaknesses in versioning, IDE integration, and support for asynchronous tasks. Data scientists rely on a rich ecosystem of open source libraries such as Pandas for sophisticated analysis, including the latest ML frameworks. They also work with data lakes that hold datasets with varying levels of data quality—a significant departure from carefully curated data warehouses. These characteristics have created new requirements for the

[a] https://www.nsf.gov/cise/ac-data-science-report/CISEACDataScienceReport1.19.17.pdf

database community to address, in collaboration with the researchers and engineers in machine learning, statistics, and data visualization.

Data to insights pipeline. Data science pipelines are often complex with several stages, each with many participants. One team prepares the data, sourced from heterogeneous data sources in data lakes. Another team builds models on the data. Finally, end users access the data and models through interactive dashboards. The database community needs to develop simple and efficient tools that support building and maintaining data pipelines. Data scientists repeatedly say that data cleaning, integration, and transformation together consume 80%-90% of their time. These are problems the database community has experienced in the context of enterprise data for decades. However, much of our past efforts focused on solving algorithmic challenges for important "point problems," such as schema mapping and entity resolution. Moving forward, we must adapt our community's expertise in data cleaning, integration, and transformation to aid the iterative end-to-end development of the data-to-insights pipeline.

Data context and provenance. Unlike applications built atop curated data warehouses, today's data scientists tap into data sources of varying quality for which correctness, completeness, freshness, and trustworthiness of data cannot be taken for granted. Data scientists need to understand and assess these properties of their data and to reason about their impact on the results of their data analysis. This requires understanding the context of the incoming data and the processes working on it. This is a *data provenance* problem, which is an active area of research for the database community. It involves tracking data, as it moves across repositories, integrating and analyzing the metadata as well as the data content. Beyond explaining results, data provenance enables reproducibility, which is key to data science, but is difficult, especially when data has a limited retention policy. Our community has made progress, but much more needs to be done to develop *scalable* techniques for data provenance.

key insights

- **Data science and database research communities must work together closely to enable data to insights pipeline.**
- **Data governance is an increasingly important societal challenge in today's data-rich world.**
- **Architectures for cloud data services need rethinking to take into consideration hardware trends, disaggregation, and new consumption models.**

Data exploration at scale. As the volume and variety of data continues to increase, our community must develop more effective techniques for discovery, search, understanding, and summarization of data distributed across multiple repositories. For example, for a given dataset, a user might want to search for public and enterprise-specific structured data that are joinable, after suitable transformations, with this dataset. The joined data may then provide additional context and enrichment for the original dataset. Furthermore, users need systems that support *interactive* exploratory analyses that can scale to large datasets, since high latency reduces the rate at which users can make observations, draw generalizations, and generate hypotheses. To support these requirements, the system stack for data exploration needs to be further optimized using both algorithmic and systems techniques. Specifically, *data profiling*, which provides a statistical characterization of data, must be efficient and scale to large data repositories. It should also be able to generate at low latency approximate profiles for large data sets to support interactive data discovery. To enable a data scientist to get from a large volume of raw data to insights through data transformation and analysis, low latency and scalable data visualization techniques are needed. Scalable data exploration is also key to addressing challenges that arise in data lakes (see "Database Engines").

Declarative programming. Even though popular data science libraries such as Pandas support tabular view of data using the DataFrame abstraction, their programming paradigms have important differences with SQL. The success of declarative query languages in boosting programmer productivity in relational databases as well as big data systems point to an opportunity to investigate language abstractions to bring the full power of declarative programming to specify all stages of data-to-insights pipelines, including data discovery, data preparation, and ML model training and inference.

Metadata management. Our community can advance the state of the art for the tracking and managing metadata related to data science experiments and ML models. This includes automated labeling and annotations of data, such as identification of data types. Metadata annotations as well as provenance need to be searchable to support experimentation with different models and model versioning. Data provenance could be helpful to determine when to retrain models. Another metadata challenge is minimizing the cost of modifying applications as a schema evolves, an old problem where better solutions continue to be needed. The existing academic solutions to schema evolution are hardly used in practice.

Data governance. Consumers and enterprises are generating data at an unprecedented rate. Our homes have smart devices, our medical records are digitized, and social media is publicly available. All data producers (consumers and enterprises) have an interest in constraining how their data is used by applications while maximizing its utility, including controlled sharing of data. For instance, a set of users might allow the use of their personal health records for medical research, but not for military applications. Data governance is a suite of technologies that supports such specifications and their enforcement. We now discuss three key facets of data governance that participants in the Seattle Database meeting thought deserves more attention. Much like data science, the database community needs to work together with other communities that share interest in these important concerns to bring transformative changes.

Data use policy. The European Union's General Data Protection Regulation (GDPR) is a prime example of such a directive. To implement GDPR and similar data use policy, metadata annotations and provenance must accompany data items as data is shared, moved, or copied according to a data use policy. Another essential element of data governance is auditing to ensure data is used by the right people for the right purpose per the data usage policy. Since data volumes continue to rise sharply, scalability of such auditing techniques is critically important. Much work is also needed to develop a framework for data collection, data retention and data disposal that supports policy constraints and will enable research on the trade-off between utility of data and limiting data gathering. Such a framework can also help answer when data may be safely discarded given a set of data usage goals.

Data privacy. A very important pillar of data governance is data privacy. In addition to cryptographic techniques to keep the data private, data privacy includes the challenges of ensuring that aggregation and other data analytic techniques may be applied effectively on a data set without revealing any individual member of the dataset. Although models such as differential privacy and local differential privacy address these challenges, more work is needed to understand how best to take advantage of these models in database platforms without significantly restricting the class of query expressions. Likewise, enabling efficient multiparty computation to enable data sharing across organizations without sacrificing privacy is an important challenge.

Ethical data science. Challenges in countering bias and discrimination in leveraging data science techniques, especially for ML, have gained traction in research and practice. The bias often comes from the input data itself such as when insufficiently representative data is used to train models. We need to work with other research communities to help mitigate this challenge. *Responsible data management* has emerged recently as a new research direction for the community and contributes to the interdisciplinary research in the broader area of *Fairness, Accountability, Transparency, and Ethics* (FATE).

Cloud services. The movement of workloads to the cloud has led to explosive growth for cloud database services, which in turn has led to substantial innovation as well as new research challenges, some of which are discussed below.

Serverless data services. In contrast to Infrastructure-as-a-Service (IaaS), which is akin to renting servers, serverless cloud database services support a consumption model that has usage-based pricing along with on-demand auto-scaling of compute and storage resources. Although the first generation of serverless cloud database services is already available and increasingly popular, there is need for research innovations to solve some of the fundamental challenges of this consumption model. Specifically, in serverless data services, users pay not only for the resources they consume but also for how quickly those resources can be allocated to their workloads. However, today's cloud database systems do not tell users how quickly they will be able to auto-scale (up and down). In other words, there is lack of transparency on the service-level agreement (SLA) that captures the trade-off between the cost of and the delay in autoscaling resources. Conversely, the architectural changes in the cloud data services that will best address the requirements for autoscaling and pay-as-you-go need to be understood from the ground up. The first example of a serverless pay-as-you-go approach that is already available today is the Function-as-a-Service (FaaS) model. The database community has made significant contributions toward developing the next generation of serverless data services, and this remains an active research area.

Disaggregation. Commodity hardware used by cloud services is subject to hardware and software failures. It treats directly attached storage as ephemeral storage and instead relies on cloud storage services that support durability, scalability, and high availability. The disaggregation of storage and compute also provides the ability to *scale compute and storage independently*. However, to ensure low latency of data services, such disaggregated architectures must use caching across multiple levels of memory hierarchy inexpensively and can benefit from limited compute within the storage service to reduce data movement (see "Database Engines"). Database researchers need to develop *principled* solutions for OLTP and analytics workloads that are suitable for a disaggregated architecture. Finally, leveraging disaggregation of memory from compute is a problem still wide open. Such disaggregation will allow compute and memory to scale independently and make more efficient use of memory among compute nodes.

Multitenancy. The cloud offers an opportunity to rethink databases in a world with an abundance of resources that can be pooled together. However, it is critical to efficiently support multitenancy do careful capacity management to control costs and optimize utilization. The research community can lead by rethinking the resource management aspect of database systems considering multitenancy. The range of required innovation here spans reimagining database systems as composite microservices, developing mechanisms for agile response to alleviate resource pressure as demand causes local spikes, and reorganizing resources among active tenants dynamically, all while ensuring tenants are isolated from noisy neighbor tenants.

Edge and cloud. IoT has resulted in a skyrocketing number of computing devices connected to the cloud, in some cases only intermittently. The limited capabilities of these devices, and diverse characteristics of their connectivity (for example, often disconnected, limited bandwidth for offshore devices, or ample bandwidth for 5G-connected devices), and their data profiles will lead to new optimization challenges for distributed data processing and analytics.

Hybrid cloud and multi-cloud. There is a pressing need to identify architectural approaches that enable on-premises data infrastructure and cloud systems to take advantage of each other instead of relying on "cloud only" or "on-premises only". In an ideal world, on-premises data platforms would seamlessly draw upon compute and storage resources available in the cloud "on-demand." We are far from that vision today even though a single control plane for data split across on-premises and cloud data is beginning to emerge. The need to take advantage of specific services available only on one cloud, avoid being locked in the "walled garden" of a single infrastructure cloud, and increase resilience to failures, has led enterprise customers to spread their data estate across multiple public clouds. Recently we have seen emergence of *data clouds* by providers of multi-cloud data services that not only support movement of data across the infrastructure clouds, but also allow their data services to operate over data split across multiple infrastructure clouds. Understanding novel optimization challenges as well as selectively leveraging past research on heterogeneous and federated databases deserves our attention.

Auto-tuning. While auto-tuning has always been desirable, it has become critically important for cloud data services. Studies of cloud workloads indicate that many cloud database applications do not use appropriate configuration settings, schema designs, or access structures. Furthermore, as discussed earlier, cloud databases need to support a diverse set of time-varying multitenant workloads. No single configuration or resource allocation works well universally. A predictive model that helps guide configuration settings and resource reallocation is desirable. Fortunately, telemetry logs are plentiful for cloud services and present a great opportunity to improve the auto-tuning functionality through use of advanced analytics. However, since the cloud provider is not allowed to have access to the tenant's data objects, such telemetry log analysis must be done in an "eyes off" mode, that is, inside of the tenant's compliance boundary. Last but not the least, cloud services provide a unique opportunity to experiment with changes to data services and measure the effectiveness of their changes, much like how Internet search engines leveraged query logs and experimented with changes in ranking algorithms.

SaaS cloud database applications. All tenants of Software-as-Service (SaaS) database applications share the same application code and have approximately (or exactly) the same database schema but no shared data. For cost effectiveness, such SaaS database applications must be multitenant. One way to support such multitenant SaaS applications is to have all tenants share one database instance with the logic to support multitenancy pushed into the application stack. While this is simple to support from a database platform perspective, it makes customization (for example, schema evolution), query optimization, and resource sharing among tenants harder. The other extreme is to spawn a separate database instance for each tenant. While this approach is flexible and offers isolation from other tenants, it fails to take advantage of the commonality among tenants and thus may incur higher cost. Yet another approach is to pack tenants into shards with large tenants placed in shards of their own. Although these architectural alternatives are known, principled tradeoffs among them as well as identifying additional support at the database services layer that may be beneficial for SaaS database applications deserves in-depth study.

> **As the volume and variety of data continues to increase, our community must develop more effective techniques for discovery, search, understanding, and summarization of data distributed across multiple repositories.**

Database engines. Cloud platforms and hardware innovations are leading to the exploration of new architectures for database systems. We now discuss some of the key themes that have emerged for research on database engines:

Heterogeneous computation. We see an inevitable trend toward heterogeneous computation with the death of Dennard scaling and the advent of new accelerators to offload compute. GPUs and FPGAs are available today, with the software stack for GPUs much better developed than for FPGAs. The progress in networking technology, including adoption of RDMA, is also receiving the attention of the database community. These developments offer the opportunity for database engines to take advantage of stack bypass. The memory and storage hierarchy are more heterogeneous than ever before. The advent of high-speed SSDs has altered the traditional tradeoffs between in-memory systems and disk-based database engines. Engines with the new generation of SSDs are destined to erode some of the key benefits of in-memory systems. Furthermore, availability of NVRAM may have significant impact on database engines due to their support for persistence and low latency. Re-architecting database engines with the right abstractions to explore hardware-software co-designs in this changed landscape, including disaggregation in the cloud context, has great potential.

Distributed transactions. Cloud data management systems are increasingly geo-distributed both within a region (across multiple availability zones) and across multiple geographic regions. This has renewed interest in industry and academia on the challenges of processing distributed transactions. The increased complexity and variability of failure scenarios, combined with increased communication latency and performance variability in distributed architectures has resulted in a wide array of trade-offs between consistency, isolation level, availability, latency, throughput under contention, elasticity, and scalability. There is an ongoing debate between two schools of thought: (a) Distributed transactions are hard to process at scale with high throughput and availability and low latency without giving up some traditional transactional guarantees. Therefore, consistency and isolation guarantees are reduced at the expense of increased developer complexity. (b) The complexity of implementing a bug-free application is extremely high unless the system guarantees strong consistency and isolation. Therefore, the system should offer the best throughput, availability, and low-latency service it can, without sacrificing correctness guarantees. This debate will likely not be fully resolved anytime soon, and industry will offer systems consistent with each school of thought. However, it is critical that application bugs and limitations in practice that result from weaker system guarantees be better identified and quantified, and tools be built to help application developers using both types of system achieve their correctness and performance goals.

Data lakes. There is an increasing need to consume data from a variety of data sources, structured, semi-structured, and unstructured, to transform and perform complex analyses *flexibly*. This has led to a transition from a classical data warehouse to a data *lake* architecture for analytics. Instead of a traditional setting where data is ingested into an OLTP store and then swept into a curated data warehouse through an ETL process, perhaps powered by a Big Data framework such as Spark, the data lake is a flexible storage repository. Subsequently, a variety of compute engines can operate on the data that are of varying data quality, to curate it or execute complex SQL queries, and store the results back in the data lake or ingest them into an operational system. Thus, data lakes exemplify a disaggregated architecture with the separation of compute and storage. An important challenge for data lakes is finding relevant data for a given task efficiently. Therefore, solutions to open problems in scalable data exploration and metadata management, discussed in the Data Science section, are of importance. While the flexibility of data lakes is attractive, it is vital that the guard rails of data governance are firmly adhered to, and we refer the reader to that section of the report for more details. To ensure consistency of data and high data quality so that the result of analytics is as accurate as possible, support for transactions, enforcement of schema constraints, and data validation are central concerns. Enabling scalable querying on the heterogeneous collection of data demands caching solutions that trade off performance, scale, and cost.

The cloud offers an opportunity to rethink databases in a world with an abundance of resources that can be pooled together. However, it is critical to efficiently support multitenancy to control costs and optimize utilization.

Approximation in query answering. As the volume of data continues to explode, we must seek techniques that reduce latency or increase throughput of query processing. For example, leveraging approximation for *fast progressive visualization* of answers to queries over data lakes can help exploratory data analysis to unlock insights in data. Data sketches are already mainstream and are classic examples of effective approximations. Sampling is another tool used to reduce the cost of query processing. However, support for sampling in today's big data systems is quite limited and does not cater to the richness of query languages such as SQL. Our community has done much foundational work in approximate query processing, but we need a better way to expose it in a programmer-friendly manner with clear semantics.

Machine learning workloads. Modern data management workloads include ML, which adds an important, new requirement for database engines. While ML workloads include training as well as inferencing, supporting the latter efficiently is an immediate need. Today the challenge of efficiently supporting "in-database" inferencing is achieved by leveraging database extensibility mechanisms. As we look forward, the ML models that are invoked as part of inferencing, must be treated as first-class citizens inside databases. ML models may be browsed and queried as database objects and database systems need to support popular ML programming frameworks. While today's database systems can support inferencing over relatively simple models, the increasing popularity and effectiveness of extremely large models such as BERT and GPT-3 requires database engine developers to leverage heterogeneous hardware and work with architects responsible for building ML infrastructure using FPGAs, GPUs, and specialized ASICs.

Machine learning for reimagining data platform components. Recent advances in ML have inspired our community to reflect on how data engine components could potentially use ML to significantly advance the state of the art. The most obvious such opportunity is *auto tuning*. Database systems can systematically replace "magic numbers" and thresholds with ML models to auto-tune system configurations. Availability of ample training data also provides opportunities to explore new approaches that take advantage of ML for query optimization or multidimensional index structures, especially as state-of-the-art solutions to these problems have seen only modest improvements in the last two decades. ML-model driven engine components must demonstrate significant benefits as well as robustness when test data or test queries deviate from the training data and training queries. To handle such deviations, the ML models need to be augmented with guardrails so that the system degrades gracefully. Furthermore, a well-thought-out software engineering pipeline to support the life cycle of a ML-model driven component will be important.

Benchmarking and reproducibility. Benchmarks tremendously helped move forward the database industry and the database research community. It is necessary to focus on benchmarking for new application scenarios and database engine architectures. Existing benchmarks (for example, TPC-E, TPC-DS, TPCH) are very useful but do not capture the full breadth of our field, for example, streaming scenarios and analytics on new types of data such as videos. Moreover, without the development of appropriate benchmarking and data sets, a fair comparison between traditional database architectures and ML-inspired architectural modifications to the engine components will not be feasible. Benchmarking in the cloud environment also presents unique challenges since differences in infrastructure across cloud providers makes apples to apples comparison more difficult. A closely related issue is reproducibility of performance results in database publications. Fortunately, since 2008, database conferences have been encouraging reproducibility of results in the papers accepted in ACM SIGMOD and VLDB. Focus on reproducibility also increases rigor in selection of workloads, databases, parameters picked for experimentation, and how results are aggregated and reported.

Community

In addition to technical challenges, the meeting participants discussed steps the community of database researchers can take to enhance our ability to contribute to and learn from the emerging data challenges.

We will continue the rich tradition of learning from users of our systems and using database conferences as meeting places for both users and system innovators. Industry tracks of our conferences foster such interaction, by discussing industry challenges and innovations in practice. This is more important due to today's rapidly changing data management challenges. We must redouble our efforts to learn from application developers or SaaS solution providers in industry verticals.

As our community develops new systems, releasing them as part of the existing popular ecosystems of open source tools or easy-to-use cloud services will greatly enhance the ability to receive feedback and do iterative improvements. Recent examples of such systems that benefited from significant input from the database community include Apache Spark, Apache Flink, and Apache Kafka. In addition, as a community, we should take advantage of every opportunity to get closer to application developers and other users of database technology to learn their unique data challenges.

The database community must do a better job integrating database research with the data science ecosystem. Database techniques for data integration, data cleaning, data processing, and data visualization should be easy to call from Python scripts.

Conclusion

We see many exciting research directions in today's data-driven world around data science, machine-learning, data governance, new architectures for cloud systems, and next-generation data platforms. This report summarized results from the Seattle Database meeting and subsequent community discussions,[5,6] which identified a few of the important challenges and opportunities for the database community to continue its tradition of strong impact on research and industry. Supplementary materials from the meeting is available on the event website.[9]

Acknowledgments. The Seattle Database meeting was supported financially by donations from Google, Megagon Labs, and Microsoft Corp. Thanks to Yannis Ioannidis, Christian Konig, Vivek Narasayya, and the anonymous reviewers for their feedback on earlier drafts.

References

1. Abadi D. et.al. The Beckman report on database research. *Commun. ACM 59*, 2 (Feb. 2016), 92–99.
2. Abadi, D. et al. The Seattle report on database research. *SIGMOD Rec. 48*, 4 (2019) 44–53 (2019)
3. Abiteboul, S. et al. The Lowell database research self-assessment. *Commun. ACM 48*, 5 (May 2005), 111–118.
4. Agrawal, R. et al. The Claremont report on database research. *Commun. ACM 52*, 6 (June 2009), 56–65.
5. Bailis, P., Balazinska, M., Luna Dong, X., Freire, J., Ramakrishnan, R., Stonebraker, M., Hellerstein, J. Winds from Seattle: Database research directions. In *Proceedings of the VLDB Endow. 13*, 12 (2020), 3516.
6. Balazinska, M., Chaudhuri, S., Ailamaki, A., Freire, J., Krishnamurthy, S., Stonebraker, M. The next 5 years: What opportunities should the database community seize to maximize its impact? In *Proceedings of SIGMOD Conf.* (2020), 411–414.
7. Bernstein, P. et. al. Future directions in DBMS research—The Laguna Beach participants. *ACM SIGMOD Record 18*, 1 (1989), 17–26.
8. Bernstein, P. et al. The Asilomar report on database research. *ACM SIGMOD Record 27*, 4 (1998), 74–80.
9. The Database Research Self-Assessment Meeting, 2018; https://db.cs.washington.edu/events/other/2018/database_self_assessment_2018.html
10. Dennard, R.H. et.al. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE J. of Solid-State Circuits SC-9*, 5 (Oct. 1974), 256–268.
11. Silberschatz, A., Stonebraker, M. and Ullman, J.D. Database systems: Achievements and opportunities. *Commun. ACM 34*, 10 (Oct. 1991), 110–120.
12. Silberschatz, A. et al. Strategic directions in database systems—breaking out of the box. *ACM Computing Surveys 28*, 4 (1996), 764–778.
13. Silberschatz, A., Stonebraker, M. and Ullman, J.D. Database research: Achievements and opportunities into the 21st century. *ACM SIGMOD Record 25*, 1 (1996), 52–63.

Surajit Chaudhuri (surajitc@microsoft.com) served as the corresponding author for this article.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/the-seattle-report