Future of Database System Architectures

Future of Database System Architectures

Gustavo Alonso
alonso@inf.ethz.ch
ETH
Zurich, Switzerland

Natassa Ailamaki
anastasia.ailamaki@epfl.ch
EPFL
Lausanne, Switzerland

Sailesh Krishnamurthy
sailesh@gmail.com
Google
Palo Alto, United States

Sam Madden
smadden@csail.mit.edu
MIT
Boston, United States

Swami Sivasubramanian
swami@amazon.com
AWS
Seattle, United States

Raghu Ramakrishnan*
raghu@microsoft.com
Microsoft
Redmond, United States

ABSTRACT

Over the past two decades, we have experienced major technology disruptions on multiple fronts, none bigger than the emergence of cloud computing, which has led to fundamental changes in how database software is architected. We are seeing several new trends that are similarly shaping the future of data management.

With the demise of Moore's Law, we are now seeing a lot of interest (and start-ups with significant investments) in hardware database accelerators, exploring FPGAs, GPUs, and more. Economies of scale in the cloud make it possible to move to hardware many things that were done in software, the trend will continue and increase.

Modern data estates are spread across data located on premises, on the edge and in one or more public clouds, spread across various sources like multiple relational databases, file and storage systems, and no- SQL systems, both operational and analytic. This phenomenon is referred to as *data sprawl.*

We are also seeing the emergence of many novel data workloads. For example, rich data pipelines are an increasingly common workload. And finally, Machine Learning is having a rapidly increasing role in every aspect of the database software lifecycle.

This SIGMOD panel will discuss the impact of the above changes and trends on database hardware and software architectures. How will these changes impact DB system design, how will DB systems look like in the near future? Where are the hardest research challenges? What learnings from the past will guide us through these disruptions?

CCS CONCEPTS

• Information systems → DBMS engine architectures; **Relational parallel and distributed DBMSs**; **Information storage systems**; **• Computing methodologies → Machine learning**; **Distributed computing methodologies**; **• Security and privacy → Software and application security**; **• Computer systems organization → Cloud computing**.

*Moderator.

ACM ISBN 978-1-4503-9507-6/23/06.
https://doi.org/10.1145/3555041.3589360

KEYWORDS

database architecture, cloud computing, machine learning, hardware acceleration, data governance, security, privacy, distributed computing, parallel databases, lakehouses, data warehouses

ACM Reference Format:
Gustavo Alonso, Natassa Ailamaki, Sailesh Krishnamurthy, Sam Madden, Swami Sivasubramanian, and Raghu Ramakrishnan. 2023. Future of Database System Architectures. In *Companion of the 2023 International Conference on Management of Data (SIGMOD-Companion '23), June 18–23, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3555041.3589360

1 INTRODUCTION

Over the past two decades, we have experienced major technology disruptions on multiple fronts, none bigger than the emergence of cloud computing, which has led to fundamental changes in how database software is architected:

- Disaggregation of compute and storage, helping to disaggregate hardware and software
- Elastic resource allocation
- Increased latencies across the memory hierarchy
- Emergence of Software-as-a-Service
- Emergence of large-scale parallel and distributed data services
- Increased use of Machine Learning for managing cloud services

With the demise of Moore's Law, we are now seeing a lot of interest (and start-ups with significant investments) in *hardware database accelerators*, exploring FPGAs, GPUs, and more. Economies of scale in the cloud make it possible to move to hardware many things that were done in software, the trend will continue and increase. For instance, tasks such as network function virtualization or expensive operations much needed in the cloud like data compression and encryption are moving to specialized accelerators and away from the bottleneck that is the CPU. These accelerators, once in place, offer many opportunities for further specialization: smart NICs, smart storage, near data processing, tailored arithmetic/logic units for ML specific data types, etc.

What is interesting about many of these use cases is that it is not often a matter of the accelerator being more efficient but of the accelerator expanding the capabilities of a compute node in ways that free up CPU capacity for user workloads. When these ideas are seen from the perspective of databases, the question is not only what of the typical functions of an engine can be made more efficient but, perhaps more importantly, what additional and new functionality could engines offer taking advantage of those accelerators.

From a research point of view, an interesting new field emerges: *How can data processing in general and databases in particular influence and leverage emerging hardware trends?* Hardware evolution and the predominance of the cloud thus create a unique environment driving computing in completely new directions. Most of the established assumptions about data management no longer hold and many known optimizations and designs are irrelevant or even detrimental in modern systems.

In recent years, another data trend has accelerated: modern data estates are spread across data located on premises, on the edge and in one or more public clouds, spread across various sources like multiple relational databases, file and storage systems, and no-SQL systems, both operational and analytic. This phenomenon is referred to as *data sprawl*. Data administrators who wish to ensure appropriate use of data (including access control, compliance with company policies, e.g., on retention, and external regulations such as the EU's General Data Protection Regulation (GDPR)) across the entire organization have to inventory their data, identify what parts of it are sensitive, and govern the sensitive data appropriately across the entirety of their sprawling data estate.

Today, governance of data is completely siloed; each of the data subsystems has its own (and varied) governance features. Policies applied to sensitive data are applied piece-meal by iterating over all the data sources in a custom language specific to each source. This makes data governance cumbersome, error-prone (because a given policy must be manually enforced across different subsystems, inconsistencies can easily arise), and expensive. Emerging standards and systems such as Open Policy Agent (OPA) and its policy language REGO seek to address these challenges.

Given data sprawl, what is the future shape of data governance, and how must database architectures evolve to support it?

We are also seeing the emergence of many novel data workloads. For example, data pipelines are an increasingly common workload, as seen in the growth of frameworks like Apache Airflow and companies in this space. *Will designing efficient data pipelines grow in importance, Or will the emergence of cloud-centric analytics lead to streamlining of common pipeline patterns (data aggregation into lakes or warehouses) being commoditized? What other workloads are likely to redefine how we think about data management?*

And finally, Machine Learning is having a rapidly increasing role in every aspect of the database software lifecycle, and it is worth looking at the various issues we have highlighted through the lens of how ML can help in addressing it. Notably, the rapid emergence of *Foundation Models* such as CHatGPT, trained on very large corpora, is likely to have a transformative effect on how everyone who engages with a database does so in future; everyone is likely to have an FM-based "assistant" to help carry out their task. *But what impact will this technology have on database architectures, on how we build and operate our data services?*

This panel will discuss the above changes and trends. How will these changes impact DB system design, how will DB systems look like in the near future? Where are the hardest research challenges? What learnings from the past will guide us through these disruptions? Will future DB systems architectures be inclusive and aware of their resources, be it storage, network, and compute hardware? Will we see a trend towards specialization (no general purpose computers but architectures tailored to cloud deployments)? Come listen to the panelists and ask them whatever is on *your* mind!