Cloud Data Services: Workloads, Architectures and Multi-Tenancy

Other titles in Foundations and Trends® in Databases

Data Provenance
Boris Glavic
ISBN: 978-1-68083-828-2

FPGA-Accelerated Analytics: From Single Nodes to Clusters
Zsolt István, Kaan Kara and David Sidler
ISBN: 978-1-68083-734-6

Distributed Learning Systems with First-Order Methods
Ji Liu and Ce Zhango
ISBN: 978-1-68083-700-1

Cloud Data Services: Workloads, Architectures and Multi-Tenancy

Vivek Narasayya
Microsoft Research
viveknar@microsoft.com

Surajit Chaudhuri
Microsoft Research
surajitc@microsoft.com

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Databases

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

V. Narasayya and S. Chaudhuri. *Cloud Data Services: Workloads, Architectures and Multi-Tenancy*. Foundations and Trends® in Databases, vol. 10, no. 1, pp. 1–107, 2021.

ISBN: 978-1-68083-775-9

Foundations and Trends® in Databases

Volume 10, Issue 1, 2021

Editorial Scope

Topics

Foundations and Trends® in Databases publishes survey and tutorial articles in the following topics:

- Data Models and Query Languages
- Query Processing and Optimization
- Storage, Access Methods, and Indexing
- Transaction Management, Concurrency Control and Recovery
- Deductive Databases
- Parallel and Distributed Database Systems
- Database Design and Tuning
- Metadata Management
- Object Management
- Trigger Processing and Active Databases
- Data Mining and OLAP
- Approximate and Interactive Query Processing
- Data Warehousing
- Adaptive Query Processing
- Data Stream Management
- Search and Query Integration
- XML and Semi-Structured Data
- Web Services and Middleware
- Data Integration and Exchange
- Private and Secure Data Management
- Peer-to-Peer, Sensornet and Mobile Data Management
- Scientific and Spatial Data Management
- Data Brokering and Publish/Subscribe
- Data Cleaning and Information Extraction
- Probabilistic Data Management

Information for Librarians

Foundations and Trends® in Databases, 2021, Volume 10, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.

Contents

1 Introduction **2**
1.1 Workloads . . . 2
1.2 Manageability . . . 4
1.3 Multi-Tenancy . . . 5
1.4 Consumption models . . . 6
1.5 Scope and outline of this article . . . 7

2 Cloud Data Services: Workloads and Architectures **9**
2.1 Data center architecture and implications . . . 10
2.2 Online Transaction Processing (OLTP) . . . 13
2.3 Data Analytics: Data Warehousing and Big Data Systems . 21
2.4 NoSQL . . . 32
2.5 Observations . . . 34

3 Multi-Tenancy: Background **37**
3.1 Virtualization Technologies in Cloud Databases . . . 38
3.2 Observations . . . 43

4 SLAs and Pricing Models **46**
4.1 SLOs and SLAs . . . 47
4.2 Factors driving SLAs . . . 48
4.3 Resource-level SLAs . . . 49
4.4 Observations . . . 54

5 Resource Management **56**
5.1 Cluster management 57
5.2 Moving Databases Across Nodes 60
5.3 Resource Scheduling 62
5.4 Performance Isolation 64
5.5 Resource estimation 68
5.6 Observations 71

6 Efficiency and Cost **72**
6.1 Improving Efficiency 73
6.2 Auto Tuning, Monitoring and Tracing 76
6.3 Observations 80

7 Serverless Databases **81**
7.1 Challenges for Serverless Databases 82
7.2 Commercial Serverless Cloud Data Services 84
7.3 Research Proposals 87
7.4 Observations 89

8 Open Problems and Conclusion **90**
8.1 Concluding Remarks 90
8.2 Open Problems 91

Acknowledgements **94**

References **95**

Cloud Data Services: Workloads, Architectures and Multi-Tenancy

Vivek Narasayya[1] and Surajit Chaudhuri[2]

[1]*Microsoft Research; viveknar@microsoft.com*
[2]*Microsoft Research; surajitc@microsoft.com*

ABSTRACT

Enterprises are moving their business critical workloads to public clouds at an accelerating pace. Cloud data services for Online Transaction Processing (OLTP), Data Analytics and NoSQL are essential building blocks for enterprise applications. Multi-tenancy is a crucial tenet for cloud data service providers that allows sharing of data center resources across tenants, thereby reducing cost. In this article we review architectures of today's cloud data services and identify trends and challenges that arise in multi-tenant cloud data services. We survey techniques that have been developed for enabling elasticity, providing SLAs, ensuring performance isolation and reducing cost. We review the emerging paradigm of serverless databases and point out opportunities and challenges. We identify open research problems in the fast-changing landscape of cloud data services.

Vivek Narasayya and Surajit Chaudhuri (2021), "Cloud Data Services: Workloads, Architectures and Multi-Tenancy", Foundations and Trends® in Databases: Vol. 10, No. 1, pp 1–107. DOI: 10.1561/1900000060.

1

Introduction

The worldwide public cloud database services market is large and growing rapidly. The wave of cloud adoption is being driven by digital transformation projects within enterprises that are migrating their applications to the cloud. According to one market study (MarketResearch, 2019), the cloud database services market is estimated to grow from around USD 12 billion in 2020 to around USD 24.8 billion in 2025. Furthermore, according to a Gartner report in 2019 titled "The Future of the DBMS Market is Cloud" (Gartner DBMS Future, 2019), a large percentage (around 68%) of the growth of the overall database market in 2018 came from cloud databases. This report also estimates that by 2022 around 75% of all databases will be deployed or migrated to a cloud platform. Major cloud vendors worldwide that offer public cloud database services include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, Alibaba Cloud, and IBM Cloud.

1.1 Workloads

Cloud database services have been developed to meet the diverse needs of enterprise applications. An overview of the classes of database services available in the cloud are shown in Figure 1.1. Relational online

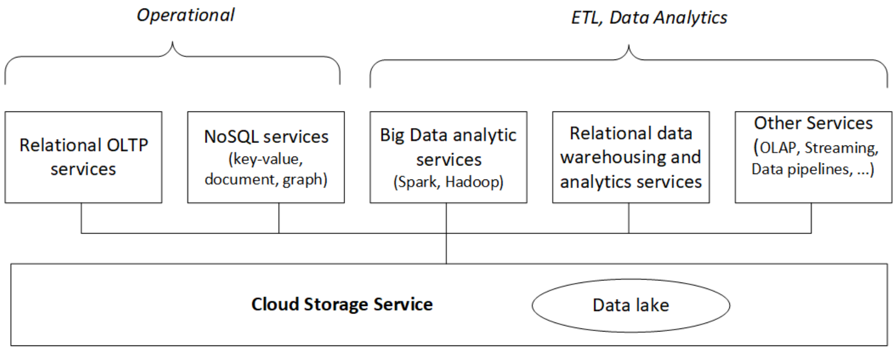

Figure 1.1: Cloud database services

transaction processing (OLTP) services enable enterprises to run their operational workloads spanning multiple industries including ATMs and electronic transfers in banking, reservation systems for airlines, shopping carts and sales in retail, and inventory control in manufacturing. OLTP services are characterized by ability to provide data consistency, high throughput, and high concurrency. Examples of cloud data services geared predominantly towards OLTP workloads are Amazon RDS and Amazon Aurora (Verbitski *et al.*, 2017), Azure SQL Database and Azure SQL Hyperscale (Antonopoulos *et al.*, 2019), and Google Cloud SQL and Google Cloud Spanner (Bacon *et al.*, 2017).

While SQL remains a popular language and is widely used, a class of NoSQL cloud data services have also gained popularity in the last decade. These services cater to applications that need to primarily store and query unstructured data using non-relational data models such as key-value, document, and graph. Examples of such NoSQL cloud data services include Google BigTable (Chang *et al.*, 2008), Amazon Dynamo DB (DeCandia *et al.*, 2007), Azure Cosmos DB (A Technical Overview of Azure Cosmos DB, 2020), MongoDB Atlas (MongoDB Atlas, 2020) and Apache Cassandra.

Data analytic workloads enable enterprises to derive actionable insights from their data (Chaudhuri *et al.*, 2011). Extract-Transform-Load (ETL) services enable transforming data from operational and external sources to prepare for use by analytic services. The industry is seeing a convergence of data warehousing and Big Data platforms towards a *data lake* architecture, where data is stored on low cost blob storage service in formats optimized for analytic query processing. This data is analyzed with elastic compute nodes using different query processing engines. All major cloud vendors now support the data lake architecture. Services such as AWS Redshift (AWS Redshift, 2020), Azure Synapse (Azure SQL Data Warehouse, 2020), Google BigQuery (Google BigQuery, 2020) and Snowflake (Dageville *et al.*, 2016) support relational data warehousing workloads in the cloud. Enterprises also use Big Data services such as Spark (Zaharia *et al.*, 2010), which uses the MapReduce paradigm popularized by the Big Data compute engine Hadoop (Apache Hadoop, 2020)). Spark is used for diverse workloads such as ETL jobs that prepare data for analysis, and for data analytics using and statistical and machine learning. Finally, there are several other cloud data services such as online analytic processing (OLAP) that supports ad-hoc interactive analysis over multi-dimensional data, and analytics over streaming data that enables near real-time decision making.

1.2 Manageability

In the on-premises setting, the customer, typically an enterprise, is responsible for provisioning and managing the entire stack of both hardware and software needed to run their database. The stack includes server machines, storage, networking, virtualization technology such as virtual machines (VMs) and containers, operating system, database management system (DBMS) and data. Although, the cost of hardware alone in on-premises is perceived to be cheaper than in the cloud, the total cost of ownership taking into account the manageability costs often make the cloud more attractive for enterprises.

Cloud database services offer improved manageability compared to on-premises databases. One option for customers is to provision a VM in the cloud, with a pre-installed image of the DBMS and run their workload in the VM. Durability of data stored in cloud storage services is guaranteed. This takes away the burden of managing hardware, VMs, and storage. Cloud providers are increasingly offering more *managed* versions of cloud database services. These manageability enhancements include patching and upgrades of the operating system and DBMS, ensuring high availability and disaster recovery of the database, automated backups, and point-in-time recovery and some facets of performance tuning.

Software-as-a-service vendors such as Salesforce and Microsoft Dynamics, that build their ERP and CRM applications on top of cloud database services, further elevate the degree of manageability provided to customers. In such SaaS applications, customers have no access to the underlying databases. While SaaS providers heavily rely on cloud database providers for manageability, they ultimately bear the responsibility of managing databases including any aspects not supported by the cloud database. For example, the SaaS provider is typically responsible for all aspects of performance monitoring and tuning.

Finally, a more recent trend aims to bring the benefits of improved manageability of the cloud platforms to on-premise databases. Platforms such as Amazon RDS for VMWare and Azure Stack allow enterprises to run database instances and storage in an on-premises environment, but manage administrative tasks such as provisioning, software installation, patching, backup and monitoring using the same control plane technology used in the public cloud platform.

1.3 Multi-Tenancy

In on-premises databases the customer owns all hardware resources in their enterprise data center, and hence the databases . In contrast, public clouds are *multi-tenant*. Multi-tenancy is crucial since dedicating hardware for each tenant is simply not cost effective. In a multi-tenant cloud data service multiple databases, potentially from different customers (a.k.a. *tenants*), share computing resources of the cloud provider including CPU, memory and disk resources of a machine and the data center network.

Cloud providers virtualize the available physical resources of a machine such as CPU, memory, disk I/O, network I/O, and local storage into logical units (e.g. VMs, containers). Each tenant's database executes within such a logical unit of virtualization. This provides two major benefits to the tenant and the cloud provider. First, virtualization typically offers a certain degree of security and performance isolation between tenants. For instance, virtualization technology may enforce resource governance mechanisms to reduce impact of a noisy neighbor on the performance of a tenant. Second, virtualization of resources is the key to enabling multi-tenancy since multiple databases can now be consolidated onto a single machine (node) while still ensuring security and performance isolation. Consolidation is *crucial* for enabling cloud providers to lower the cost of providing the service, in particular by reducing the capital expenditure (CapEx), since increased consolidation means they need to purchase fewer servers to serve the same number of databases. Thus, multi-tenancy, is a core tenet of cloud databases, and indeed cloud computing.

For a cloud database provider, a fundamental design choice is determining what virtualization technology to use so as to suitably balance security, performance and cost considerations. In practice, they employ a variety of technology ranging from VMs, containers, operating system processes, logical containers within a DBMS process, to even sharing tables within a single database. The choice, in turn, has a big impact on the quality of service, how resources are managed and the cost of running the service.

1.4 Consumption models

The most widely used consumption model in the cloud are *provisioned* databases. At the time the database is created, the tenant specifies a fixed set of resources that is promised to the database. The tenant pays for the fixed set of resources whether or not they use them. Such a consumption model is attractive for enterprises workloads since they offer the assurance that resources are always available.

More recently, cloud providers have started to offer *serverless* databases, which offer a different consumption model. Tenants no longer need to provision resources ahead of time. Rather, the cloud provider acquires and releases resources in response to demands of the database workload, and tenants only pay for resources that their workloads actually use. To enable good performance for serverless databases at low cost, cloud providers need to develop resource management techniques that are elastic and efficient.

1.5 Scope and outline of this article

Today's data center architectures, multi-tenancy and novel consumption models bring new requirements that do not exist, or are not as important, in on-premise database systems. These requirements have led to new technical challenges and significant changes in the software architecture of databases.

1. *Quality of Service and Pricing*: Customers of cloud database services may have varying needs of quality of service for performance, availability and cost. There are different kinds of service-level objectives (SLOs) that cloud service providers strive to achieve. For some services, providers guarantee an SLA for availability or performance. Similarly, there are different models of pricing of database services. These SLAs and pricing depends on several factors including multi-tenancy.

2. *Resource management*: Multi-tenancy requires that the resources in a datacenter be managed so as to ensure that each tenant achieves the performance and availability SLOs despite sharing resources with other tenants. This gives rise to technical challenges in managing clusters of machines, performance isolation, resource estimation, resource scheduling, and placement and migration of databases within a cluster of machines.

3. *Cost*: The success of cloud data services crucially depends on cost of goods sold (COGS) for the cloud provider and the total cost of ownership (TCO) for the tenant. As a consequence, several techniques have been developed such as resource harvesting and resource overcommitting, which reduces capital expenditure (CapEx), service intelligence, which reduced operating expenditure (OpEx), and auto-tuning functionality, which reduces resource consumption and TCO for tenants.

4. *Security*: Multi-tenancy in cloud database gives rise to challenges in security such as preventing malicious tenants or system and database administrators from gaining access to data.

This article focuses on state-of-the-art public cloud data services for Relational OLTP, Relational and Big Data Analytics, and NoSQL workloads. In Chapter 2, we review these workloads, and discuss the architectures of modern cloud data services, drawing contrast among services across multiple cloud providers and academic research where appropriate. In Chapter 3 we provide the background of different models of multi-tenancy that have been adopted by cloud databases. Chapter 4 focuses on the quality of service guarantees offered in commercial cloud data services and novel proposals from the research literature. We also discuss pricing options and the opportunities that they enable. In Chapter 5 we discuss the challenges and solutions for the key issues in resource management: cluster management, performance isolation, resource scheduling, and resource estimation. In Chapter 6 we review techniques for reducing cost for customers and providers of cloud data services through techniques such as overcommit, resource harvesting and auto-tuning. In Chapter 7, we review the emerging paradigm of *serverless databases*, which aim to raise the level of abstraction for provisioning and elasticity in the cloud along with pay-per-use billing.

We have made an attempt to discuss techniques developed both in industry as well as academic research. In each chapter of this article, we conclude with our observations and a set of open technical challenges on that topic. In Chapter 8 we conclude with a discussion of open issues that go beyond the specific topics discussed in individual chapters. We note that although security in cloud data services is an important topic in its own right, it is beyond the scope of this article.

References

Abadi, D. (2012). "Consistency tradeoffs in modern distributed database system design: CAP is only part of the story". *Computer*. 45(2): 37–42.

Abadi, D., A. Ailamaki, D. Andersen, P. Bailis, M. Balazinska, P. Bernstein, P. Boncz, S. Chaudhuri, A. Cheung, A. Doan, *et al.* (2020). "The Seattle Report on Database Research". *ACM SIGMOD Record*. 48(4): 44–53.

Aguilar-Saborit, J., R. Ramakrishnan, K. Srinivasan, K. Bocksrocker, Y. Ioalagia, M. Sankara, and M. Shafiei. (20120). "POLARIS: The Distributed SQL Engine in Azure Synapse". *Proceedings of the VLDB Endowment*.

Ahmad, M., A. Aboulnaga, S. Babu, and K. Munagala. (2008). "Modeling and exploiting query interactions in database systems". In: *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM. 183–192.

Akdere, M., U. Çetintemel, M. Riondato, E. Upfal, and S. B. Zdonik. (2012). "Learning-based query performance modeling and prediction". In: *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE. 390–401.

Alizadeh, M. and T. Edsall. (2013). "On the data path performance of leaf-spine datacenter fabrics". In: *2013 IEEE 21st annual symposium on high-performance interconnects*. IEEE. 71–74.

Amazon Athena. (2020). https://aws.amazon.com/athena/.

Amazon Firecracker. (2020). https://aws.amazon.com/about-aws/whats-new/2018/11/firecracker-lightweight-virtualization-for-serverless-computing/.

Amazon RDS Multi-AZ. (2020). https://aws.amazon.com/rds/features/multi-az/.

AWS Redshift. (2020). https://aws.amazon.com/redshift/.

Antonopoulos, P., A. Budovski, C. Diaconu, A. Hernandez Saenz, J. Hu, H. Kodavalla, D. Kossmann, S. Lingam, U. F. Minhas, N. Prakash, *et al.* (2019). "Socrates: The New SQL Server in the Cloud". In: *Proceedings of the 2019 International Conference on Management of Data*. ACM. 1743–1756.

Apache Hadoop. (2020). http://hadoop.apache.org.

Appuswamy, R., G. Graefe, R. Borovica-Gajic, and A. Ailamaki. (2019). "The five-minute rule 30 years later and its impact on the storage hierarchy". *Communications of the ACM*. 62(11): 114–120.

Automatic Plan Correction. (2020). https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning.

Armbrust, M., T. Das, L. Sun, B. Yavuz, S. Zhu, M. Murthy, J. Torres, H. van Hovell, A. Ionescu, A. Łuszczak, *et al.* (2020). "Delta lake: high-performance ACID table storage over cloud object stores". *Proceedings of the VLDB Endowment*. 13(12): 3411–3424.

Armbrust, M., R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, *et al.* (2015). "Spark sql: Relational data processing in spark". In: *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1383–1394.

AtScale. (2020). https://www.atScale.com/.

Amazon Aurora Serverless. (2020). https://aws.amazon.com/rds/aurora/serverless/.

Azure SQL DB Automatic Tuning. (2020). https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning.

A Technical Overview of Azure Cosmos DB. (2020). https://azure.microsoft.com/en-us/blog/a-technical-overview-of-azure-cosmos-db/.

Azure Cosmos DB Serverless. (2020). https://azure.microsoft.com/en-us/blog/build-apps-of-any-size-or-scale-with-azure-cosmos-db/.

Azure SQL Data Warehouse. (2020). https://azure.microsoft.com/en-us/services/sql-data-warehouse/.

Bacon, D. F., N. Bales, N. Bruno, B. F. Cooper, A. Dickinson, A. Fikes, C. Fraser, A. Gubarev, M. Joshi, E. Kogan, *et al.* (2017). "Spanner: Becoming a sql system". In: *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM. 331–343.

Banerjee, I., F. Guo, K. Tati, and R. Venkatasubramanian. (2013). "Memory overcommitment in the ESX server". *VMware Technical Journal*. 2(1): 2–12.

Barham, P., R. Isaacs, R. Mortier, and D. Narayanan. (2003). "Magpie: Online Modelling and Performance-aware Systems." In: *HotOS*. 85–90.

Google BigQuery. (2020). https://cloud.google.com/bigquery.

Burns, B., B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes. (2016). "Borg, Omega, and Kubernetes". *Queue*. 14(1): 10.

Chang, F., J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. (2008). "Bigtable: A distributed storage system for structured data". *ACM Transactions on Computer Systems (TOCS)*. 26(2): 1–26.

Chaudhuri, S. (1998). "An overview of query optimization in relational systems". In: *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. ACM. 34–43.

Chaudhuri, S., U. Dayal, and V. Narasayya. (2011). "An overview of business intelligence technology". *Communications of the ACM*. 54(8): 88–98.

Chaudhuri, S. and V. Narasayya. (2007). "Self-tuning database systems: a decade of progress". In: *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment. 3–14.

Chaudhuri, S., V. Narasayya, and R. Ramamurthy. (2004). "Estimating progress of execution for SQL queries". In: *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM. 803–814.

Chi, Y., H. J. Moon, and H. Hacigümüş. (2011a). "iCBS: incremental cost-based scheduling under piecewise linear SLAs". *Proceedings of the VLDB Endowment*. 4(9): 563–574.

Chi, Y., H. J. Moon, H. Hacigümüş, and J. Tatemura. (2011b). "SLA-tree: a framework for efficiently supporting SLA-based decisions in cloud computing". In: *Proceedings of the 14th International Conference on Extending Database Technology*. ACM. 129–140.

Chodorow, K. (2013). *MongoDB: the definitive guide: powerful and scalable data storage*. " O'Reilly Media, Inc."

Clark, C., K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. (2005). "Live migration of virtual machines". In: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association. 273–286.

MarketResearch. (2019). https://www.marketsandmarkets.com/Market-Reports/cloud-database-as-a-service-dbaas-market-1112.html.

Corbett, J. C., J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, *et al.* (2013). "Spanner: Google's globally distributed database". *ACM Transactions on Computer Systems (TOCS)*. 31(3): 8.

Curino, C., D. E. Difallah, C. Douglas, S. Krishnan, R. Ramakrishnan, and S. Rao. (2014). "Reservation-based scheduling: If you're late don't blame us!" In: *Proceedings of the ACM Symposium on Cloud Computing*. ACM. 1–14.

Curino, C., E. P. Jones, S. Madden, and H. Balakrishnan. (2011). "Workload-aware database monitoring and consolidation". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM. 313–324.

Dageville, B., T. Cruanes, M. Zukowski, V. Antonov, A. Avanes, J. Bock, J. Claybaugh, D. Engovatov, M. Hentschel, J. Huang, *et al.* (2016). "The snowflake elastic data warehouse". In: *Proceedings of the 2016 International Conference on Management of Data*. ACM. 215–226.

Dally, W. J. and B. P. Towles. (2004). *Principles and practices of interconnection networks*. Elsevier.

Das, S., M. Grbic, I. Ilic, I. Jovandic, A. Jovanovic, V. R. Narasayya, M. Radulovic, M. Stikic, G. Xu, and S. Chaudhuri. (2019). "Automatically indexing millions of databases in microsoft azure sql database". In: *Proceedings of the 2019 International Conference on Management of Data*. ACM. 666–679.

Das, S., F. Li, V. R. Narasayya, and A. C. Konig. (2016). "Automated demand-driven resource scaling in relational database-as-a-service". In: *Proceedings of the 2016 International Conference on Management of Data*. ACM. 1923–1934.

Das, S., V. R. Narasayya, F. Li, and M. Syamala. (2013). "CPU sharing techniques for performance isolation in multi-tenant relational database-as-a-service". *Proceedings of the VLDB Endowment*. 7(1): 37–48.

Das, S., S. Nishimura, D. Agrawal, and A. El Abbadi. (2010). "Live database migration for elasticity in a multitenant database for cloud platforms". *CS, UCSB, Santa Barbara, CA, USA, Tech. Rep*. 9: 2010.

Dean, J. and L. A. Barroso. (2013). "The tail at scale". *Communications of the ACM*. 56(2): 74–80.

Dean, J. and S. Ghemawat. (2008). "MapReduce: simplified data processing on large clusters". *Communications of the ACM*. 51(1): 107–113.

DeCandia, G., D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. (2007). "Dynamo: amazon's highly available key-value store". In: *ACM SIGOPS operating systems review*. Vol. 41. No. 6. ACM. 205–220.

Ding, B., S. Das, R. Marcus, W. Wu, S. Chaudhuri, and V. R. Narasayya. (2019). "Ai meets ai: Leveraging query executions to improve index recommendations". In: *Proceedings of the 2019 International Conference on Management of Data*. 1241–1258.

Duggan, J., U. Cetintemel, O. Papaemmanouil, and E. Upfal. (2011). "Performance prediction for concurrent database workloads". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM. 337–348.

Dutt, A., C. Wang, A. Nazi, S. Kandula, V. Narasayya, and S. Chaudhuri. (2019). "Selectivity estimation for range predicates using lightweight models". *Proceedings of the VLDB Endowment*. 12(9): 1044–1057.

Elastic Pools in Azure SQL Database. (2020). https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-pool.

Elmore, A. J., S. Das, D. Agrawal, and A. El Abbadi. (2011). "Zephyr: live migration in shared nothing databases for elastic cloud platforms". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM. 301–312.

Al-Fares, M., A. Loukissas, and A. Vahdat. (2008). "A scalable, commodity data center network architecture". *ACM SIGCOMM computer communication review*. 38(4): 63–74.

Fonseca, R., G. Porter, R. H. Katz, and S. Shenker. (2007). "X-trace: A pervasive network tracing framework". In: *4th {USENIX} Symposium on Networked Systems Design & Implementation ({NSDI} 07)*.

Ganapathi, A., H. Kuno, U. Dayal, J. L. Wiener, A. Fox, M. Jordan, and D. Patterson. (2009). "Predicting multiple metrics for queries: Better decisions enabled by machine learning". In: *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE. 592–603.

Gandhi, A., S. Thota, P. Dube, A. Kochut, and L. Zhang. (2016). "Autoscaling for hadoop clusters". In: *2016 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE. 109–118.

Ghosh, R. and V. K. Naik. (2012). "Biting off safely more than you can chew: Predictive analytics for resource over-commit in iaas cloud". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE. 25–32.

Gong, Z., X. Gu, and J. Wilkes. (2010). "Press: Predictive elastic resource scaling for cloud systems". In: *Network and Service Management (CNSM), 2010 International Conference on*. Ieee. 9–16.

Gordon, A., M. Hines, D. Da Silva, M. Ben-Yehuda, M. Silva, and G. Lizarraga. (2011). "Ginkgo: Automated, application-driven memory overcommitment for cloud computing". *Proc. RESoLVE*.

Grandl, R., G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella. (2015). "Multi-resource packing for cluster schedulers". *ACM SIGCOMM Computer Communication Review*. 44(4): 455–466.

Gray, J. and F. Putzolu. (1987). "The 5 minute rule for trading memory for disc accesses and the 10 byte rule for trading memory for CPU time". In: *Proceedings of the 1987 ACM SIGMOD international conference on Management of data*. 395–398.

Greenberg, A., J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. (2009). "VL2: a scalable and flexible data center network". In: *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 51–62.

Gulati, A., A. Merchant, and P. J. Varman. (2010). "mClock: handling throughput variability for hypervisor IO scheduling". In: *Proceedings of the 9th USENIX conference on Operating systems design and implementation*. USENIX Association. 437–450.

Hellerstein, J. M., J. Faleiro, J. E. Gonzalez, J. Schleier-Smith, V. Sreekanti, A. Tumanov, and C. Wu. (2018). "Serverless computing: One step forward, two steps back". *arXiv preprint arXiv:1812.03651*.

Herodotou, H., H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu. (2011). "Starfish: A Self-tuning System for Big Data Analytics." In: *Cidr*. Vol. 11. No. 2011. 261–272.

Hightower, K., B. Burns, and J. Beda. (2017). *Kubernetes: up and running: dive into the future of infrastructure*. " O'Reilly Media, Inc."

Hindman, B., A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. H. Katz, S. Shenker, and I. Stoica. (2011). "Mesos: A platform for fine-grained resource sharing in the data center." In: *NSDI*. Vol. 11. No. 2011. 22–22.

Huang, B., N. W. Jarrett, S. Babu, S. Mukherjee, and J. Yang. (2015). "Cümülön: Matrix-based data analytics in the cloud with spot instances". *Proceedings of the VLDB Endowment*. 9(3): 156–167.

Iorgulescu, C., R. Azimi, Y. Kwon, S. Elnikety, M. Syamala, V. Narasayya, H. Herodotou, P. Tomita, A. Chen, J. Zhang, *et al.* "PerfIso: Performance Isolation for Commercial Latency-Sensitive Services". In: *2018 USENIX Annual Technical Conference USENIX ATC 18), pages=519–532, year=2018, organization=USENIX Association*.

Jain, N., I. Menache, and O. Shamir. (2014). "On-demand, spot, or both: Dynamic resource allocation for executing batch jobs in the cloud".

Jonas, E., J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar, *et al.* (2019). "Cloud programming simplified: A berkeley view on serverless computing". *arXiv preprint arXiv:1902.03383*.

Jones, C. and J. Wilkes. (2016). "Service Level Objectives. Site Reliability Engineering: How Google Runs Production Systems".

Kakivaya, G., L. Xun, R. Hasha, S. B. Ahsan, T. Pfleiger, R. Sinha, A. Gupta, M. Tarta, M. Fussell, V. Modi, *et al.* (2018). "Service fabric: a distributed platform for building microservices in the cloud". In: *Proceedings of the Thirteenth EuroSys Conference*. ACM. 33.

Karger, D., E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. (1997). "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web". In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 654–663.

Khoussainova, N., M. Balazinska, and D. Suciu. (2012). "Perfxplain: debugging mapreduce job performance". *Proceedings of the VLDB Endowment*. 5(7): 598–609.

Kipf, A., T. Kipf, B. Radke, V. Leis, P. Boncz, and A. Kemper. (2018). "Learned cardinalities: Estimating correlated joins with deep learning". *arXiv preprint arXiv:1809.00677*.

Lang, W., K. Ramachandra, D. J. DeWitt, S. Xu, Q. Guo, A. Kalhan, and P. Carlin. (2016). "Not for the Timid: On the Impact of Aggressive Over-booking in the Cloud". *Proceedings of the VLDB Endowment*. 9(13): 1245–1256.

Las-Casas, P., G. Papakerashvili, V. Anand, and J. Mace. (2019). "Sifter: Scalable Sampling for Distributed Traces, without Feature Engineering". In: *Proceedings of the ACM Symposium on Cloud Computing*. 312–324.

Li, J., A. C. König, V. Narasayya, and S. Chaudhuri. (2012). "Robust estimation of resource consumption for sql queries using statistical techniques". *Proceedings of the VLDB Endowment*. 5(11): 1555–1566.

Lo, D., L. Cheng, R. Govindaraju, P. Ranganathan, and C. Kozyrakis. (2015). "Heracles: improving resource efficiency at scale". In: *ACM SIGARCH Computer Architecture News*. Vol. 43. No. 3. ACM. 450–462.

Lu, J., Y. Chen, H. Herodotou, and S. Babu. (2019). "Speedup Your Analytics: Automatic Parameter Tuning for Databases and Big Data Systems". *Proceedings of the VLDB Endowment*. 12(12): 1970–1973.

Luo, G., J. F. Naughton, C. J. Ellmann, and M. W. Watzke. (2004). "Toward a progress indicator for database queries". In: *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM. 791–802.

Marcus, R., P. Negi, H. Mao, C. Zhang, M. Alizadeh, T. Kraska, O. Papaemmanouil, and N. Tatbul. (2019). "Neo: A learned query optimizer". *arXiv preprint arXiv:1904.03711*.

Marcus, R. and O. Papaemmanouil. (2018). "Deep reinforcement learning for join order enumeration". In: *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. ACM. 3.

Melnik, S., A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis. (2010). "Dremel: interactive analysis of web-scale datasets". *Proceedings of the VLDB Endowment*. 3(1-2): 330–339.

Melnik, S., A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, T. Vassilakis, H. Ahmadi, D. Delorey, S. Min, *et al.* (2020). "Dremel: a decade of interactive SQL analysis at web scale". *Proceedings of the VLDB Endowment*. 13(12): 3461–3472.

Misra, P. A., Í. Goiri, J. Kace, and R. Bianchini. "Scaling distributed file systems in resource-harvesting datacenters". In: *2017 USENIX Annual Technical Conference USENIX ATC 17), pages=799–811, year=2017, organization=USENIX Association*.

MongoDB Atlas. (2020). https://www.mongodb.com/.

Narasayya, V., S. Das, M. Syamala, B. Chandramouli, and S. Chaudhuri. (2013). "Sqlvm: Performance isolation in multi-tenant relational database-as-a-service".

Narasayya, V., I. Menache, M. Singh, F. Li, M. Syamala, and S. Chaudhuri. (2015). "Sharing buffer pool memory in multi-tenant relational database-as-a-service". *Proceedings of the VLDB Endowment*. 8(7): 726–737.

AWS Nitro System. (2020). https://aws.amazon.com/ec2/nitro/.

Oracle MultiTenant. (2013). https://www.oracle.com/technetwork/database/multitenant-wp-12c-1949736.pdf.

Ortiz, J., V. T. De Almeida, and M. Balazinska. (2015). "Changing the Face of Database Cloud Services with Personalized Service Level Agreements." In: *CIDR*.

Pavlo, A., G. Angulo, J. Arulraj, H. Lin, J. Lin, L. Ma, P. Menon, T. C. Mowry, M. Perron, I. Quah, *et al.* (2017). "Self-Driving Database Management Systems." In: *CIDR*.

Perron, M., R. Castro Fernandez, D. DeWitt, and S. Madden. (2020). "Starling: A Scalable Query Engine on Cloud Functions". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 131–141.

Google Persistent Disk. (2020). https://cloud.google.com/persistent-disk/.

Popescu, A. D., A. Balmin, V. Ercegovac, and A. Ailamaki. (2013). "PREDIcT: towards predicting the runtime of large scale iterative analytics". *Proceedings of the VLDB Endowment*. 6(14): 1678–1689.

Microsoft Power BI. (2020). https://powerbi.microsoft.com.

Rosenblum, M. and T. Garfinkel. (2005). "Virtual machine monitors: Current technology and future trends". *Computer*. 38(5): 39–47.

Roy, S., A. C. König, I. Dvorkin, and M. Kumar. (2015). "Perfaugur: Robust diagnostics for performance anomalies in cloud services". In: *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE. 1167–1178.

Selinger, P. (2017). "Optimizer Challenges in a Multi-Tenant World". In: *High Performance Transaction Systems, HPTS 2017*.

Sethi, R., M. Traverso, D. Sundstrom, D. Phillips, W. Xie, Y. Sun, N. Yegitbasi, H. Jin, E. Hwang, N. Shingte, *et al.* (2019). "Presto: SQL on Everything". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. 1802–1813.

Shen, Z., S. Subbiah, X. Gu, and J. Wilkes. (2011). "Cloudscale: elastic resource scaling for multi-tenant cloud systems". In: *Proceedings of the 2nd ACM Symposium on Cloud Computing*. ACM. 5.

Sigelman, B. H., L. A. Barroso, M. Burrows, P. Stephenson, M. Plakal, D. Beaver, S. Jaspan, and C. Shanbhag. (2010). "Dapper, a large-scale distributed systems tracing infrastructure".

Azure SQL DB Serverless. (2020). https://docs.microsoft.com/en-us/azure/sql-database/sql-database-serverless.

Oracle SQL Trace. (2020). https://docs.oracle.com/database/121/TGSQL/tgsql_trace.htm.

Sreekanti, V., C. W. X. C. Lin, J. M. Faleiro, J. E. Gonzalez, J. M. Hellerstein, and A. Tumanov. (2020). "Cloudburst: Stateful Functions-as-a-Service". *arXiv preprint arXiv:2001.04592*.

Tableau Online. (2020). https://www.tableau.com/products/cloud-bi.

Gartner DBMS Future. (2019). https://www.gartner.com/document/3941821.

Urgaonkar, B., P. Shenoy, and T. Roscoe. (2009). "Resource overbooking and application profiling in a shared internet hosting platform". *ACM Transactions on Internet Technology (TOIT)*. 9(1): 1.

Valiant, L. G. and G. J. Brebner. (1981). "Universal schemes for parallel communication". In: *Proceedings of the thirteenth annual ACM symposium on Theory of computing*. 263–277.

Van Aken, D., A. Pavlo, G. J. Gordon, and B. Zhang. (2017). "Automatic database management system tuning through large-scale machine learning". In: *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM. 1009–1024.

Vavilapalli, V. K., A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, *et al.* (2013). "Apache hadoop yarn: Yet another resource negotiator". In: *Proceedings of the 4th annual Symposium on Cloud Computing*. ACM. 5.

Verbitski, A., A. Gupta, D. Saha, M. Brahmadesam, K. Gupta, R. Mittal, S. Krishnamurthy, S. Maurice, T. Kharatishvili, and X. Bao. (2017). "Amazon aurora: Design considerations for high throughput cloud-native relational databases". In: *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM. 1041–1052.

Verbitski, A., A. Gupta, D. Saha, J. Corey, K. Gupta, M. Brahmadesam, R. Mittal, S. Krishnamurthy, S. Maurice, T. Kharatishvilli, *et al.* (2018). "Amazon aurora: On avoiding distributed consensus for i/os, commits, and membership changes". In: *Proceedings of the 2018 International Conference on Management of Data*. 789–796.

VMware, E. "Understanding Memory Resource Management in VMware ESX 4.1".

Voorsluys, W., J. Broberg, S. Venugopal, and R. Buyya. (2009). "Cost of virtual machine live migration in clouds: A performance evaluation". In: *IEEE International Conference on Cloud Computing*. Springer. 254–265.

Waldspurger, C. A. (2002). "Memory resource management in VMware ESX server". *ACM SIGOPS Operating Systems Review*. 36(SI): 181–194.

Weikum, G., A. Moenkeberg, C. Hasse, and P. Zabback. (2002). "Self-tuning database technology and information services: from wishful thinking to viable engineering". In: *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier. 20–31.

Weissman, C. D. and S. Bobrowski. (2009). "The design of the force.com multitenant internet application development platform". In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 889–896.

Wu, C., J. Faleiro, Y. Lin, and J. Hellerstein. (2019). "Anna: A kvs for any scale". *IEEE Transactions on Knowledge and Data Engineering*.

Wu, W., Y. Chi, H. Hacigümüş, and J. F. Naughton. (2013). "Towards predicting query execution time for concurrent and dynamic database workloads". *Proceedings of the VLDB Endowment*. 6(10): 925–936.

Wu, W., X. Wu, H. Hacigümüş, and J. F. Naughton. (2014). "Uncertainty aware query execution time prediction". *Proceedings of the VLDB Endowment*. 7(14): 1857–1868.

Extended Events Overview. (2019). https://docs.microsoft.com/en-us/sql/relational-databases/extended-events/.

Xiong, P., Y. Chi, S. Zhu, H. J. Moon, C. Pu, and H. Hacgümüş. (2015). "SmartSLA: Cost-sensitive management of virtualized resources for CPU-bound database services". *IEEE Transactions on Parallel and Distributed Systems*. 26(5): 1441–1451.

Xiong, P., Y. Chi, S. Zhu, J. Tatemura, C. Pu, and H. HacigümüŞ. (2011). "ActiveSLA: a profit-oriented admission control framework for database-as-a-service providers". In: *Proceedings of the 2nd ACM Symposium on Cloud Computing*. ACM. 15.

Yoon, D. Y., N. Niu, and B. Mozafari. (2016). "DBSherlock: A performance diagnostic tool for transactional databases". In: *Proceedings of the 2016 International Conference on Management of Data*. ACM. 1599–1614.

Yun, H., G. Yao, R. Pellizzoni, M. Caccamo, and L. Sha. (2013). "Memguard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms". In: *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2013 IEEE 19th*. IEEE. 55–64.

Zaharia, M., M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. (2010). "Spark: Cluster computing with working sets." *HotCloud*. 10(10-10): 95.

Zhan, Z.-H., X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li. (2015). "Cloud computing resource scheduling and a survey of its evolutionary approaches". *ACM Computing Surveys (CSUR)*. 47(4): 1–33.

Zhang, N., P. J. Haas, V. Josifovski, G. M. Lohman, and C. Zhang. (2005). "Statistical learning techniques for costing XML queries". In: *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment. 289–300.

Zhang, Y., G. Prekas, G. M. Fumarola, M. Fontoura, Í. Goiri, and R. Bianchini. (2016). "History-based harvesting of spare cycles and storage in large-scale datacenters". In: *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*. No. EPFL-CONF-224446. 755–770.