

Robust Query Processing in the Cloud

Goetz Graefe^{*1}, Allison Lee^{*2}, and Caetano Sauer^{*3}

1 Google – Madison, US. goetz.graefe@gmail.com

2 Snowflake – San Mateo, US. allison@snowflake.com

3 Salesforce – München, DE. caetano.sauer@salesforce.com

Abstract

The Dagstuhl Seminar on “Robust Query Processing in the Cloud” (24101), held from March 3 to March 8, 2024, brought together researchers from academia and industry to discuss robustness in database management systems. This seminar was a continuation of previous seminars on the topic of Robust Query Processing, where we focused in particular on cloud computing and also discussed aspects that have not been addressed by the previous instances of the seminar. This article summarizes the main discussion topics, and presents the summary of the outputs of five working groups that discussed: i) robustness benchmarking, ii) economics of query processing in the cloud, iii) storage architectures, iv) out-of-memory query operators, and v) indexing for data warehousing.

Seminar March 3–8, 2024 – <https://www.dagstuhl.de/24101>

2012 ACM Subject Classification Information systems → Data management systems; Information systems → Storage architectures

Keywords and phrases database, execution, hardware, optimization, performance, query

Digital Object Identifier 10.4230/DagRep.14.3.1

1 Executive Summary

Goetz Graefe (Google – Madison, US)

Allison Lee (Snowflake – San Mateo, US)

Caetano Sauer (Salesforce – München, DE)

License © Creative Commons BY 4.0 International license
© Goetz Graefe, Allison Lee, and Caetano Sauer

The Dagstuhl Seminar on “Robust Query Processing in the Cloud” (24101) assembled researchers from industry and academia for the fifth time to discuss robustness issues in database query performance, this time with a focus on Cloud Computing. The seminar gathered researchers around the world working on indexing, storage, plan generation and plan execution in database query processing, and in cloud-based massively parallel systems with the purpose to address the open research challenges with respect to the robustness of database management systems. Delivering robust query performance is well known to be a difficult problem for database management systems. All experienced DBAs and database users are familiar with sudden disruptions in data centers due to poor performance of queries that have performed perfectly well in the past. The goal of the seminar was to discuss the current state-of-the-art, to identify specific research opportunities in order to improve the state-of-affairs in query processing, and to develop new approaches or even solutions for these opportunities, building upon successes of the past Dagstuhl Seminars [2, 3, 5, 1, 4, 6]. The organizers (Goetz Graefe, Allison Lee, and Caetano Sauer) this time attempted to have a

* Editor / Organizer



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Robust Query Processing in the Cloud, *Dagstuhl Reports*, Vol. 14, Issue 3, pp. 1–8

Editors: Goetz Graefe, Allison Lee, and Caetano Sauer



DAGSTUHL
REPORTS Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

focused subset of topics that the participants discussed and analyzed in more depth. From the proposed topics on algorithm choices, join sequences, storage architectures, database utilities, modern storage hardware, cloud database economics, and benchmarking for robust query processing, the participants formed five work groups: i) robustness benchmarking, ii) economics of query processing in the cloud, iii) storage architectures, iv) out-of-memory query operators, and v) indexing for data warehousing. Upon choosing the topics of interest, the organizers then guided the participants to approach the topic through a set of steps: by first considering related work in the area; then introducing metrics and tests that will be used for testing the validity and robustness of the solution; after metrics, the focus was on proposing specific mechanisms for the proposed approaches; and finally the last step focused on the implementation policies. At the end of the week, each group presented their progress with the hope to continue their work towards a research publication. The reports of work groups are presented next.

References

- 1 Peter A. Boncz, Yannis Chronis, Jan Finis, Stefan Halfpap, Viktor Leis, Thomas Neumann, Anisoara Nica, Caetano Sauer, Knut Stolze, and Marcin Zukowski. SPA: economical and workload-driven indexing for data analytics in the cloud. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 3740–3746. IEEE, 2023.
- 2 Renata Borovica-Gajic, Stratos Idreos, Anastasia Ailamaki, Marcin Zukowski, and Campbell Fraser. Smooth scan: Statistics-oblivious access paths. In Johannes Gehrke, Wolfgang Lehner, Kyuseok Shim, Sang Kyun Cha, and Guy M. Lohman, editors, *ICDE*, pages 315–326. IEEE Computer Society, 2015.
- 3 Renata Borovica-Gajic, Stratos Idreos, Anastasia Ailamaki, Marcin Zukowski, and Campbell Fraser. Smooth scan: robust access path selection without cardinality estimation. *VLDB J.*, 27(4):521–545, 2018.
- 4 David Justen, Daniel Ritter, Campbell Fraser, Andrew Lamb, Nga Tran, Allison Lee, Thomas Bodner, Mhd Yamen Haddad, Steffen Zeuch, Volker Markl, and Matthias Boehm. POLAR: adaptive and non-invasive join order selection via plans of least resistance. *Proc. VLDB Endow.*, 17(6):1350–1363, 2024.
- 5 Martin L. Kersten, Alfons Kemper, Volker Markl, Anisoara Nica, Meikel Poess, and Kai-Uwe Sattler. Tractor pulling on data warehouses. In Goetz Graefe and Kenneth Salem, editors, *DBTest*, page 7. ACM, 2011.
- 6 Lukas Vogel, Daniel Ritter, Danica Porobic, Pinar Tözün, Tianzheng Wang, and Alberto Lerner. Data pipes: Declarative control over data movement. In *13th Conference on Innovative Data Systems Research, CIDR 2023, Amsterdam, The Netherlands, January 8-11, 2023*. www.cidrdb.org, 2023.

2 Table of Contents

Executive Summary

Goetz Graefe, Allison Lee, and Caetano Sauer 1

Working groups

Benchmarking Robustness Using Perturbed Workloads

Manos Athanassoulis, Carsten Binnig, Yannis Chronis, John Cieslewicz, Sudipto Das, Stefan Halfpap, Allison Lee, Bernhard Seeger, and Nga Tran 4

From ASAP to JUST Pricing: The Economics of Robust Query Performance in the Cloud

Thomas Bodner, Peter A. Boncz, Goetz Graefe, Viktor Leis, Danica Porobic, and Caetano Sauer 4

The Query Exchange: Auctioning and Bidding on Query Workloads for a Competitive Cloud Database Market

Thomas Bodner, Peter A. Boncz, Goetz Graefe, Viktor Leis, Danica Porobic, and Caetano Sauer 5

SAUNA – Saved Aggregates Unleash Novel Acceleration

Nicolas Bruno, Campbell Fraser, Kyoungmin Kim, Anisoara Nica, Immanuel Trummer, and Juliane Waack 5

Coping with Out-of-Memory Situations in Distributed Join Processing

Matthias Böhm, Periklis Chrysogelos, Thanh Do, Jan Finis, Alfons Kemper, Thomas Neumann, Knut Stolze, and Marcin Zukowski 6

The 5 minute rule for the cloud: When are caches needed?

Andrew Lamb, Angelos Christos Anadiotis, Kira Isabel Duwe, Lucas Lersch, Boaz Leskes, Daniel Ritter, Kai-Uwe Sattler, and Pinar Tözün 7

Participants 8

3 Working groups

3.1 Benchmarking Robustness Using Perturbed Workloads

Manos Athanassoulis (Boston University, US), Carsten Binnig (TU Darmstadt, DE), Yannis Chronis (Google – Sunnyvale, US), John Cieslewicz (Google – Mountain View, US), Sudipto Das (Amazon Web Services – Seattle, US), Stefan Halfpap (Technische Universität Berlin, DE), Allison Lee (Snowflake – San Mateo, US), Bernhard Seeger (Universität Marburg, DE), and Nga Tran (InfluxData – Boston, US)

License © Creative Commons BY 4.0 International license
 © Manos Athanassoulis, Carsten Binnig, Yannis Chronis, John Cieslewicz, Sudipto Das, Stefan Halfpap, Allison Lee, Bernhard Seeger, and Nga Tran

The robustness of database systems is an important and extensively discussed topic in academia and industry. While various new approaches have been proposed to improve the robustness of database systems – including new robust query operators such as smooth scan – surprisingly, there is still no method that allows us to holistically analyze and quantify the robustness of a database system. In this paper, we define the very first method that allows us to quantify the robustness of a database system. Intuitively, our method analyzes whether small changes in the execution of a workload lead to only small observed changes in the performance of the database system. Building on this robustness definition, we propose a new benchmark framework that tests the robustness of a database system. This framework can take any existing workload as input (e.g., TPC-H or TPC-DS) and by injecting small perturbations targeting, for instance, query and data characteristics, quantify the robustness of a system. Importantly, along with the benchmark, we propose a new robustness metric that allows us to quantify robustness at the system level and enable comparison across systems or versions of the same system.

3.2 From ASAP to JUST Pricing: The Economics of Robust Query Performance in the Cloud

Thomas Bodner (Hasso-Plattner-Institut, Universität Potsdam, DE), Peter A. Boncz (CWI – Amsterdam, NL), Goetz Graefe (Google – Madison, US), Viktor Leis (TU München – Garching, DE), Danica Porobic (Oracle Switzerland – Zürich, CH), and Caetano Sauer (Salesforce – München, DE)

License © Creative Commons BY 4.0 International license
 © Thomas Bodner, Peter A. Boncz, Goetz Graefe, Viktor Leis, Danica Porobic, and Caetano Sauer

We propose the concept of performance SLAs (Service Level Agreements) for SQL workloads, as an enabler for new pricing models in cloud databases. We identify two new research areas necessary for making this possible: (i) reliable methods to determine SLA pricing and associated financial risks and profits, (ii) technical innovations in cloud database engines that take into account query workload deadlines, and employ elastic resource allocation to make these deadlines. In all, we think that this new model, on the one hand finally offers customers a real handle on workload SLAs, while on the other hand enables both customer and provider cost saving as well as reduced hardware resource usage (and thus reduced carbon footprint).

3.3 The Query Exchange: Auctioning and Bidding on Query Workloads for a Competitive Cloud Database Market

Thomas Bodner (Hasso-Plattner-Institut, Universität Potsdam, DE), Peter A. Boncz (CWI – Amsterdam, NL), Goetz Graefe (Google – Madison, US), Viktor Leis (TU München – Garching, DE), Danica Porobic (Oracle Switzerland – Zürich, CH), and Caetano Sauer (Salesforce – München, DE)

License © Creative Commons BY 4.0 International license
© Thomas Bodner, Peter A. Boncz, Goetz Graefe, Viktor Leis, Danica Porobic, and Caetano Sauer

In today’s cloud analytics market, customers choose providers to serve their query workloads for an extended period of time. In contrast, an efficient choice per query (or per workload) would enable and encourage innovation for efficient database systems. Building on shared storage in the public cloud, we propose the concept of a query exchange. A query exchange implements a competitive market, brokering queries to the provider with the lowest bid for execution satisfying the customer requirements. Using open table formats and appropriate access governance, alternative providers can read and process the same data sources. Relying on specifications of data, metadata and workload SLAs, the query exchange aligns customers’ incentive to lower costs with providers’ incentive to process queries for which they have a competitive advantage, thus achieving efficient, scalable, and robust query performance in the cloud.

3.4 SAUNA – Saved Aggregates Unleash Novel Acceleration

Nicolas Bruno (Microsoft – Redmond, US), Campbell Fraser (Google – Mountain View, US), Kyoungmin Kim (EPFL – Lausanne, CH), Anisoara Nica (SAP SE – Waterloo, CA), Immanuel Trummer (Cornell University – Ithaca, US), and Juliane Waack (Snowflake – Berlin, DE)

License © Creative Commons BY 4.0 International license
© Nicolas Bruno, Campbell Fraser, Kyoungmin Kim, Anisoara Nica, Immanuel Trummer, and Juliane Waack

The Dagstuhl Seminar “Database Indexing and Query Processing” in March 2022 produced the SPA paper [1], which presented a general framework for improving the performance of scanning modern column stores. SPA introduced several interesting building blocks, such as incremental building of intermediate structures and using economic principles to guide the construction (and deconstruction) of such structures. We explored several methods inspired by that framework to improve the performance of query processing in a robust way. More precisely, we explored the following ideas, in a new framework SAUNA – Saved Aggregates Unleash Novel Acceleration.

Single table aggregate result caching. We can see the original block pruning technique as creating index structures that can answer boolean aggregate queries about whether any tuple exists in a block. We generalize the approach to return the actual results from evaluating single table aggregates with filters, resulting in an incremental/partial materialized view algorithm that follows the workload.


Using single table aggregate results to prune blocks . When single table aggregates return MIN/MAX values, these values can be used to generate runtime constraints over each block, and conceptually reason whether the original filter plus the constraint results in a contradiction, which results in novel ways to do block-level pruning.

Block pruning for fact-dimension table joins where dimension tables change slowly over time. We extend the index structures on the blocks of a fact table to include precomputed information about joins over small dimension tables that change rarely. In that way, we complement bloom filters with more precise information that can skip blocks that are guaranteed not to join with any row in the dimension tables. When executing a query plan, we can keep track of block ids that remain after each operator, detect the join operators that are selective and prune most of the blocks compared to sub-operators, and adaptively index the patterns to our index (join patterns as keys and lists of block ids as values). This idea can be easily extended to cover arbitrary query plans and operators including aggregations.

Reinforcement learning with partial configurations . State-of-the-art methods for automated database tuning rely on reinforcement learning. To find optimal solutions, such approaches need to try out various tuning options. This can lead to non-robust performance from the user’s perspective. E.g., to find optimal index configurations, current reinforcement learning methods must create and drop indexes. This can lead to significant performance variations, even when running the same query repeatedly. Considering partial configurations, e.g., indexes that cover only data subsets, can help to improve robustness in such scenarios. On the other hand, benchmarking configurations that differ only in their configuration for small data subsets may make it harder to reliably identify optima. To explore those tradeoffs, a proof-of-concept prototype was created, using reinforcement learning to find optimal index configurations in a search space of partial configurations. First experimental results show that this approach still finds optimal indexing solutions while improving performance robustness very significantly.

3.5 Coping with Out-of-Memory Situations in Distributed Join Processing

Matthias Böhm (TU Berlin, DE), Periklis Chrysogelos (Oracle Switzerland – Zürich, CH), Thanh Do (Celonis Inc. – New York, US), Jan Finis (Salesforce – München, DE), Alfons Kemper (TU München – Garching, DE), Thomas Neumann (TU München – Garching, DE), Knut Stolze (Ocient – Jena, DE), and Marcin Zukowski (Snowflake – San Mateo, US)

License  Creative Commons BY 4.0 International license
© Matthias Böhm, Periklis Chrysogelos, Thanh Do, Jan Finis, Alfons Kemper, Thomas Neumann, Knut Stolze, and Marcin Zukowski

A major challenge in distributed query processing in the cloud is the handling of mispredicted input cardinalities and skew. In order to mitigate this problem, our group set out to devise graceful strategies for unexpected out-of-memory situations in distributed joins, group-bys, and window aggregates. After detailed discussions of existing system architectures and operational challenges, we defined a framework for detecting and mitigating such out-of-memory situations. At its core, we create virtual partitions, and optimize configurations including the mapping of virtual partitions to nodes, build-side broadcasting, probe-side materialization, as well as the number of utilized nodes. Furthermore, our group explored a broad list and classification of other unexpected performance problems of user-facing issues (e.g., LIKE predicates, UDFs, and APIs), query compilation issues (e.g., generated huge queries and expression trees), as well as runtime challenges (e.g., parallelism for pipelines with small inputs, loads, tail latency, resource isolation, large strings).

3.6 The 5 minute rule for the cloud: When are caches needed?

Andrew Lamb (InfluxData – Boston, US), Angelos Christos Anadiotis (Oracle Switzerland – Zürich, CH), Kira Isabel Duwe (EPFL – Lausanne, CH), Lucas Lersch (Amazon Web Services – East Palo Alto, US), Boaz Leskes (MotherDuck – Amsterdam, NL), Daniel Ritter (SAP SE – Walldorf, DE), Kai-Uwe Sattler (TU Ilmenau, DE), and Pinar Tözün (IT University of Copenhagen, DK)

License © Creative Commons BY 4.0 International license

© Andrew Lamb, Angelos Christos Anadiotis, Kira Isabel Duwe, Lucas Lersch, Boaz Leskes, Daniel Ritter, Kai-Uwe Sattler, and Pinar Tözün

Many modern cloud database systems use disaggregated architectures, separating the computations and the underlying object storages (e.g S3). Traditionally, database systems have used caches to improve performance to improve the data on local SSDs, essentially following the so called 5-minute rule of thumb to decide when to cache in-memory or read directly from local storage. As data moves to object stores, which have highly unpredictable tail latency and explicit costs per access, the question naturally arises whether these new disaggregated architectures need comparable caches. In practice, many systems do use caches between object storage and compute, however caches can introduce new overall system robustness challenges (e.g. cache misses) as well as add non-trivial expense both in terms of engineering and operational overhead. In this paper, we review the requirements that lead to object store caching layers, analyze the design space, and propose new rules of thumb to help system designers determine under what circumstances they should introduce caches instead of reading directly from cloud object storage.

Participants

- Angelos Christos Anadiotis
Oracle Switzerland – Zürich, CH
- Manos Athanassoulis
Boston University, US
- Carsten Binnig
TU Darmstadt, DE
- Thomas Bodner
Hasso-Plattner-Institut,
Universität Potsdam, DE
- Matthias Böhm
TU Berlin, DE
- Peter A. Boncz
CWI – Amsterdam, NL
- Nicolas Bruno
Microsoft – Redmond, US
- Yannis Chronis
Google – Sunnyvale, US
- Periklis Chrysogelos
Oracle Switzerland – Zürich, CH
- John Cieslewicz
Google – Mountain View, US
- Sudipto Das
Amazon Web Services –
Seattle, US
- Thanh Do
Celonis Inc. – New York, US
- Kira Isabel Duwe
EPFL – Lausanne, CH
- Jan Finis
Salesforce – München, DE
- Campbell Fraser
Google – Mountain View, US
- Goetz Graefe
Google – Madison, US
- Stefan Halfpap
Technische Universität
Berlin, DE
- Alfons Kemper
TU München – Garching, DE
- Kyoungmin Kim
EPFL – Lausanne, CH
- Andrew Lamb
InfluxData – Boston, US
- Allison Lee
Snowflake – San Mateo, US
- Viktor Leis
TU München – Garching, DE
- Lucas Lersch
Amazon Web Services – East
Palo Alto, US
- Boaz Leskes
MotherDuck – Amsterdam, NL
- Thomas Neumann
TU München – Garching, DE
- Anisoara Nica
SAP SE – Waterloo, CA
- Danica Porobic
Oracle Switzerland – Zürich, CH
- Daniel Ritter
SAP SE – Walldorf, DE
- Kai-Uwe Sattler
TU Ilmenau, DE
- Caetano Sauer
Salesforce – München, DE
- Bernhard Seeger
Universität Marburg, DE
- Knut Stolze
Ocient – Jena, DE
- Pinar Tözün
IT University of
Copenhagen, DK
- Nga Tran
InfluxData – Boston, US
- Immanuel Trummer
Cornell University – Ithaca, US
- Juliane Waack
Snowflake – Berlin, DE
- Marcin Zukowski
Snowflake – San Mateo, US

