

Report on the Fourth International Workshop on Self-Managing Database Systems (SMDB 2009)

Ashraf Aboulnaga and Kenneth Salem
University of Waterloo
{ashraf,kmsalem}@uwaterloo.ca

1 Introduction

The Fourth International Workshop on Self-Managing Database Systems took place on March 29th, 2009 in Shanghai, China, on the day before ICDE. The SMDB workshops bring together researchers and practitioners interested in making data management systems easier to deploy and operate effectively. Topics of interest range from “traditional” management issues, such as database physical design, system tuning, and resource allocation to more recent challenges around scalable and highly-available data services in cloud computing environments. The SMDB Workshops are sponsored by the IEEE TCDE Workgroup on Self-Managing Database Systems.

The SMDB 2009 program began with a keynote presentation by James Hamilton, Vice President and Distinguished Engineer with Amazon Web Services. This was followed by five technical papers, presented in two sessions. The workshop concluded with a panel discussion on “grand challenges” in database self-management. More information about the workshop program, including links to papers and presentation slides, can be found at the SMDB 2009 web site, at <http://db.uwaterloo.ca/tcde-smdb/smdb09>.

2 Keynote

James Hamilton is a Vice President and Distinguished Engineer with Amazon Web Services. His background includes 15 years in database engine development with IBM and Microsoft and 5 years in the cloud service space with Microsoft and Amazon. He is also the author of widely-read blog, at <http://perspectives.mvdirona.com/>. James’ keynote presentation was titled “Cloud-Computing Economies of Scale”. It covered hardware trends affecting cloud computing and a variety of useful techniques for scaling cloud services.

James began his presentation with a discussion of the economies of scale in networking, storage, and administration that are achievable by very large scale cloud service providers like Amazon. He observed that while the largest cost component in enterprise data centers is typically people (administrators), this is not true for very large scale cloud service providers. For these organizations, costs are dominated by hardware, cooling, power and power distribution. His figures showed power and power-related costs approaching 50% of total expenses and trending upwards, while hardware costs are trending down.

James next addressed the question of what ultimately limits the application of arbitrarily large amounts of computing power to cloud applications. He argued that one limiting factor is power, since it is likely to become

Copyright 0000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

the dominant cost component for cloud providers. The other limiting factor is communications - the need to get data to the processors. In James' words, the "CPUs are outpacing all means to feed them". He noted the widening "storage chasm" between DRAM and disks, which is encouraging larger memories and more disks. Finally, he noted the large and growing gap between sequential and random disk bandwidth.

In the last part of his presentation, James covered a variety of useful techniques for building scalable services. First, since scalable and highly-available services demand partitioning and redundancy, it is necessary to build such services on top of large numbers of unreliable, distributed hardware and software components. James argued that recovery-oriented computing [1] is the only affordable administrative model when there are many components. This involves deeply monitoring applications and treating failures as routine occurrences to be handled, when possible, by simple automated recovery actions, such as rebooting and re-imaging. Human administrators become involved only if these automated steps do not work. Speaking of the limits of auto-administration, James noted that one should never allow an automated administration system to make a decision that would not be entrusted to a junior operations engineer.

Second, there are many techniques that can be used to lessen the impact of the disk/DRAM chasm, including sequentialization of I/O workloads, the use of distributed memory caches, I/O cooling, and I/O concentration. On the subject of solid state storage devices (SSDs), James urged caution, noting that they should be used only if they produce a win in terms of work-per-dollar or work-per-joule. He argued that SSDs are "wonderful for some workloads", not plain wonderful.

Third, we should expect user data to be pulled closer to the edges of the network to support highly interactive applications and because of political and other restrictions on data movement. Conversely, we should expect aggregated data to be pulled towards the network core where it can be subjected to analysis. While growth in transactional workloads tends to be related to business growth, growth in analysis workloads is linked to the cost of the analyses. As the cost of analysis drops, more types of analyses become economically viable.

Fourth, the spread between average and peak loads is high, so unmanaged load spikes will result in overload. Admission controls or service degradations are necessary to manage overload conditions. There are considerable potential benefits around resource consumption shaping, e.g., "pushing" peak loads into load troughs.

James concluded by noting that utility computing is here to stay because of the cost advantages he had already described, and that the most difficult aspect of providing cloud computing services is "managing multi-center distributed partitioned redundant, data stores and caches", which "really is a database problem."

3 Paper Sessions

The five papers presented at the workshop dealt with a wide range of research issues related to self-managing database systems. Belknap et al. [2] presented a fully automated workflow for tuning SQL queries that is part of Oracle Database 11g. The goal of this work is to automate the invocation of the SQL Tuning Advisor that is available in Oracle for tuning SQL statements, but that had to be invoked manually by the database administrator. The work identifies the SQL statements in the user workload that would be good candidates for automatic tuning. To ensure that recommended tuning actions do indeed result in improved performance, an interesting approach is adopted in which candidate query execution plans recommended by the tuner are executed and their actual performance is measured to decide the best tuning action. Some safeguards are put in place to ensure that the tuner itself does not consume too much time and resources.

Thiem and Sattler [3] presented integrated performance monitoring in the Ingres database system. This is an approach for monitoring performance data at a high resolution from within the database system, with the goal of providing a basis for better-informed automatic tuning decisions. The collected data includes CPU time, number of I/O's, wall clock time, indexes used, and other performance metrics. A relational view over the in-memory monitoring data is provided to users and tuning tools, and the data is also written to an on-disk workload database. The authors observe that collecting the monitoring data adds little overhead to the system, but writing

it to the workload database is expensive, so a better mechanism for persisting the monitoring data is still needed.

Dash et al. [4] proposed an economic model for self-tuned caching in the cloud, specifically targeting scientific applications. The setting considered by the paper is one in which scientific data is stored in a cloud of databases, and users pose queries against this data. The focus of the paper is developing an economic model for deciding which index structures to create for the offered user queries. This model balances the utility of an index structure to user queries against the cost of creating this index structure and the amount that users are willing to pay for different levels of service.

Schnaitter and Polyzotis [5] presented a benchmark for evaluating online index tuning algorithms. Many algorithms for online index selection have been proposed, so a principled approach for evaluating them is definitely welcome. The proposed benchmark models a database hosting environment in which multiple data sets are hosted on one server and multiple workload suites are executed against these data sets. The workload suites include SQL statements of varying complexity, and the benchmark shifts between these workload suites to model dynamic workload behavior. The benchmark's effectiveness was demonstrated by using it to compare two online index selection algorithms implemented in PostgreSQL.

Finally, Hose et al. [6] presented an online approach for selecting materialized views in an OLAP environment. The proposed approach targets OLAP server middleware that uses a data cube as the logical model and translates multi-dimensional queries into SQL queries over a relational star schema. These SQL queries are typically expensive, and materialized views that pre-aggregate the data are often used to speed them up. The paper presents an approach for choosing the set of materialized views for a workload and adjusting this set over time in an online manner. This requires a query representation that identifies how pre-computed aggregates can be used to answer a query, a cost model to estimate the cost of a query if some set of aggregates is materialized, and a search algorithm to identify the aggregates to pre-compute and materialize at each stage of workload execution.

4 Closing Panel

The workshop concluded with a panel on “Grand Challenges in Database Self-Management”, with six distinguished panelists: Anastassia Ailamaki from Ecole Polytechnique Federale de Lausanne, Shivnath Babu from Duke University, Nicolas Bruno from Microsoft Research, Benoit Dageville from Oracle, James Hamilton from Amazon, and Calisto Zuzarte from the IBM Toronto Software Lab.

The panelists presented a wide range of views on the research challenges in the area of database self-management. Bruno presented the need for better modeling of query execution times (e.g., by modeling concurrency and locking) and richer workload models (e.g., taking into account stored procedures with conditionals). He mentioned other challenges such as developing benchmarks for physical design tools (the problem addressed in [5]), extending self-management techniques to the cloud, exploiting large user bases to extract data and diagnose problems, and developing a unifying theory for self-tuning database systems.

Dageville presented some of the work that Oracle has recently done on database manageability, covering areas such as statistics collection, SQL tuning, memory management, and SQL repair. Looking ahead, he saw the need for dealing with bad SQL statements from application developers with little database expertise, and for robust query execution plans. He also saw the need for working towards proactively fixing problems before they occur, for example by leveraging the experience of other users. Another important area he identified is extending manageability solution to the entire software stack, beyond just the database system.

Ailamaki focused on using self-managing database systems in e-science. She mentioned that an important challenge in e-science is the sheer scale of the data. She pointed out that in this setting there are many aspects of a database systems that need to be self-managed, such as logical and physical design, power and energy awareness, and resource utilization. The responsibilities of the database management layer also include ensuring policy compliance, allocating resources for scaling, and handling failures. She listed open problems including dynamic or underspecified workloads, non-relational data models, changes in database statistics, enabling scalability, and

finding a path for deploying research ideas to the scientists using e-science systems.

Hamilton pointed out that relational database systems are losing workloads in the cloud. Many cloud applications use database systems, but most of the complexity (e.g., partitioning) is handled above the database system. Furthermore, many new applications use non-relational data models and many interesting workloads have left relational database systems. For example, analysis workloads are being handled by Hadoop, and caching is handled by memcached and similar solutions. Database infrastructures that focus on simplicity and scalability are becoming increasingly popular. These include BigTable, Amazon SimpleDB, Facebook Cassandra, BerkelyDB, and many others. Hamilton identified several problems causing relational database systems to lose ground in the cloud: lack of scalability, excess administrative complexity, resource intensity due to un-needed features, unpredictable response times, excessively random data access patterns, opaque failure modes, and slow adaptability to new workloads. The solution he proposed involves supporting simple caching models and richer execution models, and embracing recovery-oriented computing to improve robustness.

Zuzarte observed that computing has gone from centralized mainframes to distributed client-server systems, and now back to a centralized system with dumb clients talking to a cloud. The challenges he posed include optimizing for objectives other than minimizing elapsed time or resource consumption; for example, minimizing energy consumption or cloud computing dollar costs. He also observed that the Map-Reduce paradigm can benefit from techniques developed in the context of traditional database systems such as sharing scans, progressive re-optimization, and on-the-fly indexing.

Babu advocated using an experiment-driven approach for tuning and problem determination. This approach provides simple and robust solutions to many manageability problems, but it poses several challenges such as identifying the right type of experiment for a given problem and planning the sequence of experiments to conduct. Other challenges include providing an infrastructure for running experiments (e.g., in the cloud) and combining experiment-driven management with current database tuning tools.

5 SMDB 2010

SMDB 2010 is being organized by Shivnath Babu and Kai-Uwe Sattler, and is scheduled for March 1, 2010 in Long Beach, California, immediately preceding ICDE 2010. Its scope includes emerging research areas around database self-management, such as cloud computing, multi-tenant databases, storage services, and datacenter administration. Visit <http://db.uwaterloo.ca/tcde-smdb/smdb10> for more information.

References

- [1] D. Patterson et al. Recovery oriented computing (ROC): Motivation, definition, techniques. Technical Report CSD-02-1175, Berkeley, CA, USA, 2002.
- [2] Peter Belknap, Benoit Dageville, Karl Dias, and Khaled Yagoub. Self-tuning for SQL performance in Oracle Database 11g. In *Proc. SMDB'09*, pages 1694–1700. IEEE Computer Society, 2009.
- [3] Alexander Thiem and Kai-Uwe Sattler. An integrated approach to performance monitoring for autonomous tuning. In *Proc. SMDB'09*, pages 1671–1678. IEEE Computer Society, 2009.
- [4] Debabrata Dash, Verena Kantere, and Anastasia Ailamaki. An economic model for self-tuned cloud caching. In *Proc. SMDB'09*, pages 1687–1693. IEEE Computer Society, 2009.
- [5] Karl Schnaitter and Neoklis Polyzotis. A benchmark for online index selection. In *Proc. SMDB'09*, pages 1701–1708. IEEE Computer Society, 2009.
- [6] Katja Hose, Daniel Klan, and Kai-Uwe Sattler. Online tuning of aggregation tables for OLAP. In *Proc. SMDB'09*, pages 1679–1686. IEEE Computer Society, 2009.