Are We Experiencing a Big Data Bubble?

Fatma Özcan
IBM Almaden Research Center
San Jose, CA
fozcan@us.ibm.com

Nesime Tatbul
Intel Labs and MIT
Cambridge, MA
tatbul@csail.mit.edu

Daniel J. Abadi
Yale University
New Haven, CT
dna@cs.yale.edu

Marcel Kornacker
Cloudera
San Francisco, CA
marcel@cloudera.com

C Mohan
IBM Almaden Research Center
San Jose, CA
cmohan@us.ibm.com

Karthik Ramasamy
Twitter, Inc.
San Francisco, CA
karthik@twitter.com

Janet Wiener
Facebook, Inc.
Menlo Park, CA
jlw@fb.com

Categories and Subject Descriptors

H.4 [**Database Management**]: Systems

1. INTRODUCTION

Over the last decade, the database field has seen resurgence with the big data wave. Accelerated increase in data volumes, and modern hardware have been two major factors that brought in significant investment in new database technologies. Our field has benefited from this increased interest and focus. There is now an abundance of NoSQL, NewSQL, and SQL-on-Hadoop systems.

According to nosql-database.org, the list of NoSQL databases [6] has reached 150. Many of these systems claim horizontal scalability, and support for non-relational data. However, this high scalability usually comes at the cost of strong support for ACID transactions. Most of them only provide eventual consistency, or even worse, defer managing transactional semantics to the application layer.

Another important aspect of these NoSQL systems is the lack of declarative query interfaces. Most only support programmable APIs and do not have a query language. While this allows them to support flexible schemas and nested data types easily, it comes at the cost of physical independence and query optimization. Application programmers now have the responsibility to write their own optimized data access plans.

Yet another aspect is their restrictive feature set and specialized domain focus. Most only support object-level put and get interfaces, and have trouble providing even simple grouping and aggregation support. As these systems evolve, the need for more functionality is pushing them to consider providing more traditional database-like features. However, their ad-hoc designs prevent their wider adaptability and extensibility.

It is important to note that NoSQL systems are very popular with application programmers, despite the above-mentioned issues. Because they offer simple to use programmable APIs, flexible schemas, and high scalability. Application programmers can implement, deploy, and scale out their applications very easily and quickly.

NewSQL systems [2, 5] have emerged as a new class of relational database management systems to match the scalable performance of NoSQL systems without giving up ACID guarantees and a SQL-based declarative query interface. They achieve high performance and scalability by offering clean-slate architectural redesigns that take better advantage of modern hardware platforms such as shared-nothing clusters of many-core machines with large or non-volatile in-memory storage.

On the analytics side, MapReduce emerged as the platform for all analytics needs of the enterprise. It even fueled a major controversy about parallel databases. Hive [9] was the first SQL-on-Hadoop solution which provided an SQL-like interface, and used the underlying MapReduce infrastructure for scheduling and data movement. Hadapt [1] tried to take the best of MapReduce and databases for efficient SQL processing over Hadoop data. Over the last year, there was a major shift in SQL-on-Hadoop systems to more database-like architectures, such as Impala [4], Presto [7], and Tajo [8]. There are now even solutions that use existing parallel database technology, such as HAWQ [3].

All these advances prove the importance of the basic principles of database systems, which is the result of several decades of database research. Most of these systems seem to be coming full circle back to core database techniques. Yet, it is also important to note that traditional databases failed to provide the needed features, such as high horizontal scalability and flexible schemas, which fueled the development of these new technologies in the first place.

In this panel, we would like to explore the implications of these systems and the role of database researchers in shaping the next decade of data management solutions.

2. PANEL QUESTIONS

- Are all these systems addressing many niche markets, or is there too much overlap and redundancy?
- Without standard APIs or languages, is it chaos out there? Do we need new standards?
- Did we all give up on declarative languages, query processing and optimization?

SIGMOD'14, June 22–27, 2014, Snowbird, UT, USA.
Copyright 2014 ACM 978-1-4503-2376-5/14/06 ...$15.00.
http://dx.doi.org/10.1145/2588555.2610531.

- Why are so many application developers willing to give up on ACID transactions? Is there something different about modern applications?
- There is a renewed interest in SQL, especially in the enterprise. Is it a sufficiently rich language to support the big data ecosystem, or do we need specialized languages?
- Is there something fundamentally new in these systems, or are we reinventing the wheel?
- What can we learn from NoSQL, NewSQL, and SQL-on-Hadoop systems?
- How will the data management market look like in 5 years? How many of these systems will survive?
- What is the opportunity and the responsibility of database researchers to shape this future?

3. REFERENCES

[1] A. Abouzeid, K. Bajda-Pawlikowski, D. J. Abadi, A. Rasin, and A. Silberschatz. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. *PVLDB*, 2(1):922–933, 2009.

[2] M. Aslett. How Will the Database Incumbents Respond to NoSQL and NewSQL? The 451 Group, 2011.

[3] Pivotal HAWQ. `http://pivotalhd.docs.gopivotal.com/getting-started/hawq.html`.

[4] Impala. `github.com/cloudera/impala`.

[5] Michael Stonebraker and Samuel Madden and Daniel J. Abadi and Stavros Harizopoulos and Nabil Hachem and Pat Helland. The End of an Architectural Era (It's Time for a Complete Rewrite). In *VLDB*, pages 1150–1160, 2007.

[6] C. Mohan. History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla. In *EDBT*, 2013.

[7] Presto. `http://prestodb.io/`.

[8] Tajo. `http://tajo.incubator.apache.org/`.

[9] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Anthony, H. Liu, and R. Murthy. Hive - A Petabyte Scale Data Warehouse Using Hadoop. In *ICDE*, pages 996–1005, 2010.