History Repeats Itself:
Sensible and NonsenSQL Aspects of the NoSQL Hoopla

C. Mohan
Dr. C. Mohan

- IBM Almaden Research Center, San Jose, CA.
- IBM Fellow, 1997
  - IBM's highest technical position.
- ACM Fellow, 2002
- IEEE Fellow, 2002
- ACM SIGMOD Innovations Award, 1996
Dr. C. Mohan

- R* : Distributed RDBMS
- Starburst : ORDBMS
- Lotus Notes : GroupWare

ARIES Algorithm

- **WAL:** Push “Log Updates” to disk, then push “Data Updates” to disk.
- **FAILURE.**
- **RECOVERY:**
  - Check “Log Updates” in disk.
  - **REDO:** Redo all actions up until crash.
  - **UNDO:** Roll Back uncommitted Transactions.
About NoSQL Systems:

“In rushing to develop these systems to overcome some of the shortcomings of the relational systems, many good principles of the latter, which go beyond the relational model and the SQL language, have been left by the wayside.” [3]
Dr. Michael Stonebraker

PAPER: The End of an Architectural Era (It's Time for a Complete Rewrite)

Moreover, at the time relational DBMSs were conceived, there was only a single DBMS market, business data processing. In the last 25 years, a number of other markets have evolved, including data warehouses, text management, and stream processing. These markets have very different requirements than business data processing.” [4]
PAPER: The End of an Architectural Era (It's Time for a Complete Rewrite)

Get Rid of:
- Relational Data Model
- SQL
“Of late, it has become fashionable to discredit RDBMSs, and a significant chunk of the technologies that have been laboriously thought about and worked out over the last few decades. Some inconvenient/inadequate features of RDBMSs in certain contexts have been used as arguments to 'throw the baby with the bath water' while coming up with alternatives. As some of us anticipated, many features which were initially considered unnecessary/undesirable are now being retrofitted to the NoSQL systems, in many cases in ad hoc and simple-minded ways which could lead to problems, if not now, in due course of time.” [3]
NoSQL: Who?

- Triggered by Web 2.0 companies: Amazon, Facebook, Google, and Yahoo.
- Variety of NoSQL systems: Key-Value stores, Document stores, Graph stores, column-oriented stores.
- **Examples:** BigTable, Cassandra, Dynamo.
- Helped by Open-Source Movement.

Cite: [6]
NoSQL: Why?

- Web 2.0 companies deal with unstructured data. Relational Model is too rigid.
- No support for JSON data format.
- Programmers have to learn JavaScript AND SQL.
- Less need for data consistency.
- Must use Cloud Services. Must scale faster than RDBMS.
- Open-Source support is cost-effective.

Cite: [6]
Dr. C. Mohan

- “The goal is to avoid the pitfalls with starting out with simple designs and then, after realizing their inadequacies, adding layers of patches to attempt to provide advanced functionality” [3]
3 Painful Mistakes

- IBM System/38's table-wide latching. Latch conflicts became a problem.

- Lotus Notes had no transaction or logging system. Adding transactions and log-based recovery was painful.

- DB2 & Sybase: Reducing granularity of locking from page size to something smaller was a problem.
Current NoSQL Mistakes

- Lack of **standard API** makes migration difficult.
- Developers are now responsible for **Query Optimization**.
- **Complex data models** make migration difficult among NoSQL DBMSs.
- **Scaling Document stores** is difficult because large documents expose inefficient replication and locking.
- **Looser Transaction and Consistency semantics** will confuse developers.
Current NoSQL Mistakes

- Lack of **standard API** makes migration difficult.
- Developers are now responsible for **Query Optimization**.
- **Complex data models** make migration difficult among NoSQL DBMSs.
- **Scaling Document stores** is difficult because large documents expose inefficient replication and locking.
- **Looser Transaction and Consistency semantics** will confuse developers.
Future?

• In the Past, popular alternatives to RDBMSs were eventually absorbed into the RDBMS infrastructure.

• Example: OODBMS and XML DBMS.

• It is likely that a lot of NoSQL features will also be absorbed into the RDBMS infrastructure.
PRO / CON

• There are more specialized markets now.

• Specialized products for specialized markets would be faster than general product.

• Some RDBMS concepts are still relevant to most specialized markets. Not everything should be thrown away.

• Loss of standardization is a problem.
REFERENCES:


• [3]:
  C. Mohan:
  History repeats itself: sensible and NonsenseQL aspects of the NoSQL hoopla. EDBT 2013: 11-16

• [4]:
  Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos, Nabil Hachem, Pat Helland:
  The End of an Architectural Era (It's Time for a Complete Rewrite). VLDB 2007: 1150-1160

• [5]: http://www.csail.mit.edu/user/1547
REFERENCES:

- [6]: https://en.wikipedia.org/wiki/Apache_Cassandra