

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.



Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/1978542.1978546

<http://cacm.acm.org/blogs/blog-cacm>

Stonebraker on NoSQL and Enterprises

Michael Stonebraker discusses several reasons why NoSQL has not caught on with enterprise users.



Michael Stonebraker
"Why Enterprises Are Uninterested in NoSQL"

<http://cacm.acm.org/blogs/blog-cacm/99512>

Sept. 30, 2010

According to a recent ReadWriteWeb blog post by Audrey Watters, 44% of enterprise users questioned had never heard of NoSQL and an additional 17% had no interest. So why are 61% of enterprise users either ignorant about or uninterested in NoSQL? This post contains my two cents' worth on the topic.

At a recent trade show I attended that highlighted NoSQL engines, there were many Web developers, mostly from startups. However, I was struck by the absence of enterprise users. Hence, my (totally unscientific) experience confirms the basic point of the above blog post.

Moreover, in my experience, most information among enterprise users occurs by word of mouth. Hence, if they don't hear about something, it is because their professional network does not pass the word along. In other words, an interested enterprise professional generates additional interest. Non-

interest generates the behavior seen in the above blog post. So why is enterprise interest lacking?

To get more color on the situation, I contacted a very senior technical guru at a large enterprise who is responsible for looking at new database management system (DBMS) technology for his company. I asked him how interested he was in NoSQL and, in effect, how interested his company was. He reported "no interest." I asked him why.

He first said the vast majority of his company's applications are classifiable as online transaction processing (OLTP) where there are frequent small updates to a database of structured records or data warehouses/data marts that assemble historical business data for ad hoc query by analysts. Although there are other applications around the "edges," such as document management, these are not considered important.

He then made one comment about OLTP, one about warehouses, and one general comment. These follow.

No ACID Equals No Interest

Much of the OLTP data kept by this company is mission critical. Screwing it up causes people to lose their jobs. In

his world, ACID is the gold standard for updates to shared datasets. Any system that does not support real transactions is considered a nonstarter in his OLTP environment.

Even if a dataset can get by with single-record transactions now (a common feature of NoSQL DBMSs), he is unwilling to guarantee that it will never need multi-record transactions in the future. Put differently, his company assumes that ACID may be required in the future for any OLTP dataset, and nixes non-ACID systems.

A Low-Level Query Language is Death

Data warehouses are subject to frequent ad hoc queries like "Tell me whether pet rocks are selling better than Barbie dolls in the south?" Ted Codd's pioneering paper, "A Relational Model of Data for Large Shared Data Banks," in 1970 advocated a user interface whereby one stated what data he required instead of writing an algorithm to fetch relevant data from disk. In the subsequent 40 years of DBMS activity, high-level languages, like SQL, have been shown to offer ease of programming for such ad hoc data warehouse inquiries. My enterprise guru's company is rarely interested in the algorithmic record-at-a-time interfaces seen in most NoSQL products, as they are seen as a throwback to the days of IMS and CODASYL.

NoSQL Means No Standards

His company has a large number of databases (apparently more than 10,000), and the company is clearly concerned

with the number of different kinds of interfaces their application programmers have to learn. Hence, standards are important to a large enterprise.

Seemingly, there are north of 50 NoSQL engines, each with a different user interface. Most have a data model, which is unique to that system, along with a one-off, record-at-a-time user interface. My enterprise guru was very concerned with the proliferation of such one-offs. In contrast, SQL offers a standard environment.

I want to close this blog post with a single comment: “Those who do not understand the lessons from previous generation systems are doomed to repeat their mistakes.” In other words, “Stand on the shoulders of those who came before you, not on their toes.”

Disclosure: Michael Stonebraker is associated with four startups that are either producers or consumers of database technology. Hence, his opinions should be considered in this light.

Comments

This blog post makes me wonder why I pay \$100 a year to ACM.

Are you seriously going to sit there and disregard a very viable set of database options just because one person in one enterprise environment says he's uninterested? Or are you pushing your own agenda in the disguise of public opinion?

How do we teach the up-and-coming professionals that they should use the best tool for the job when presumably one of the top DB guys in the industry is waging a war on new technologies in the database field? I say presumably, because your continual dismissal of NoSQL solutions will render you irrelevant.

—Srdjan Pejic

Srdjan,

I am in no position to defend the author but it seems to me that what he is writing here is not NoSQL bashing. This article is a valuable thing; it is making clear to any NoSQL vendor what the barriers are that need to be overcome.

I work for an ISV that sells software to large enterprises and the issues raised here are the issues that would prevent us from using NoSQL. Our customers want to write their own reports using existing data warehouses; they want a RDBMS that fits into their exist-

ing support model.

“How do we teach the up-and-coming professionals that they should use the best tool for the job...” You do that by teaching them to use the best tool for the job; the point is that NoSQL is not going to be the best tool for the job as long as these barriers remain. “The job” is rarely just the application itself; data lives on forever, and enterprises want to use data everywhere and NoSQL vendors need to embrace that reality if they want to be enterprise players.

—Jamison M.

Srdjan,

At the top of the article it was made clear that it isn't “just one person”: “44% of enterprise users questioned had never heard of NoSQL and an additional 17% had no interest. So why are 61% of enterprise users ignorant about or uninterested in NoSQL?” Not to mention the fact that ACM has featured many articles enthusiastic about NoSQL, does that validate your \$100 a year?

In addition, it is quite clear that to an enterprise, NoSQL options are not “viable” for exactly the reasons stated.

I'd have to say, though, that the disclaimer at the bottom of this article is uncalled for, especially since similar disclaimers have not appeared on articles by proponents of NoSQL solutions (who are also financially invested in that tech).

—Jay Wright

Srdjan,

This is why Stonebraker is waging a counterargument to NoSQL: The average NoSQL fan lacks the ability to compare and understand relational database performance vs. NoSQL alternatives.

Nowhere has Mike ever stated, “For specific large dataset problems, SQL continues to outperform NoSQL.” Instead, I've seen him advocate for specific solutions to specific problems. CStore becomes Vertica, H-Store becomes Volt, and those who know better chose Postgres over MySQL.

In my personal growth, I came to understand that most of my startup's scalability problems had been solved before. Any time we started to get excited about Cassandra, BigTable, Dryad-LINQ, Pnuts, or K-V stores like Redis, Tokyo Cab, Couch, or Mongo, a more reasoned voice in our team was able

to educate everyone else that a typical relational SQL solution was still quite scalable while offering far superior consistency or isolation. We saw time and time again that NoSQL hype can easily trend toward uninformed religion.

There are very few people working on problems that really need to care about NoSQL or consistency-relaxed alternatives. Stonebraker's opinion is necessary to seriously question the NoSQL fanboy's understanding; he advocates different flavors of database solutions for different problems. That fact stands in stark contrast to your accusation that he ignores the best tool for the job, or is being rendered irrelevant.

—Jeff Vyduna

Jeff,

First of all, please do not assume I am a NoSQL “fanboy.” Also, how is it that you're sure I lack the ability to “compare and understand relational database performance vs. NoSQL alternatives,” as you put it?

A survey by *InformationWeek* is not a good representative of opinion. Most would say it's actually biased to favor established players like Microsoft and Oracle, so basing an article on those numbers is dubious at best.

Second, since you seem to have not read my comments carefully, I was complaining about the influence of this post by this author on “the best tool for the job” paradigm.

If your startup determines that basing your data store on a relational database is the best way to go, I will fully support you in that choice. Personally, I know that requirements my projects have fit better with a data model based on a K-V store like Mongo for stuff other than e-commerce. The e-commerce portion will go into something like Postgres, because the need for consistency is greater. Again, best tool for the job.

Jay, Why weren't any of these many enthusiastic articles referenced here as a counterpoint? Could it be Mike has an agenda against NoSQL solutions?

As for the stats reference, refer to what I wrote above about *InformationWeek*.

—Srdjan Pejic

Michael Stonebraker is an adjunct professor at the Massachusetts Institute of Technology.

© 2011 ACM 0001-0782/11/08 \$10.00