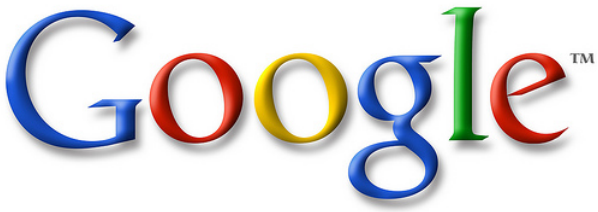
Google Spanner's Most Surprising Revelation: NoSQL is Out and NewSQL is In

Monday, September 24, 2012 at 9:12AM

Todd Hoff in Strategy, google, nosql



Google recently released a paper on Spanner, their planet enveloping tool for organizing the world's monetizable information. Reading the Spanner paper I felt it had that chiseled in stone feel that all of Google's best papers have. An instant classic. Jeff Dean foreshadowed Spanner's humungousness as early as 2009. Now Spanner seems fully online, just waiting to handle "millions of machines across hundreds of datacenters and trillions of database rows." Wow.

The Wise have yet to weigh in on Spanner en masse. I look forward to more insightful commentary. There's a lot to make sense of. What struck me most in the paper was a deeply buried section essentially describing Google's motivation for shifting away from NoSQL and to NewSQL. The money quote:

> ***We believe it is better to have application programmers deal with performance problems due to overuse of transactions as bottlenecks arise, rather than always coding around the lack of transactions.***

This reads as ironic given Bigtable helped kickstart the NoSQL/eventual consistency/key-value revolution.

We see most of the criticisms leveled against NoSQL turned out to be problems for Google too. Only Google solved the problems in a typically Googlish way, through the fruitful melding of advanced theory and technology. The result: programmers get the real transactions, schemas, and query languages many crave along with the scalability and high availability they require.

The full quote:

> *Spanner exposes the following set of data features to applications: a data model based on schematized semi-relational tables, a query language, and general purpose transactions. The move towards supporting these features was driven by many factors. The need to support schematized semi-relational tables and synchronous replication is supported by the popularity of Megastore [5].*
>
> *At least 300 applications within Google use Megastore (despite its relatively low performance) because its data model is simpler to manage than Bigtable's, and because of its support for synchronous replication across datacenters. (Bigtable only supports eventually-consistent replication across datacenters.) Examples of well-known Google applications that use Megastore are Gmail, Picasa, Calendar, Android Market, and AppEngine.*
>
> *The need to support a SQLlike query language in Spanner was also clear, given the popularity of Dremel [28] as an interactive data analysis tool. Finally, the lack of cross-row transactions in Bigtable led to*

> *frequent complaints; Percolator [32] was in part built to address this failing.*
>
> *Some authors have claimed that general two-phase commit is too expensive to support, because of the performance or availability problems that it brings [9, 10, 19]. We believe it is better to have application programmers deal with performance problems due to overuse of transactions as bottlenecks arise, rather than always coding around the lack of transactions. Running two-phase commit over Paxos mitigates the availability problems.*

What was the cost? It appears to be latency, but apparently not of the crippling sort, though we don't have benchmarks. In any case, Google thought dealing with latency was an easier task than programmers hacking around the lack of transactions. I find that just fascinating. It brings to mind so many years of RDBMS vs NoSQL arguments it's not even funny.

I wonder if Amazon could build their highly available shopping cart application, said to a be a motivator for Dynamo, on top of Spanner?

Is Spanner the Future in the Same Way Bigtable was the Future?

Will this paper spark the same revolution that the original Bigtable paper caused? Maybe not. As it is Open Source energy that drives these projects, and since few organizations need to support transactions on a global scale (yet), whereas quite a few needed to do something roughly Bigtablish, it might be awhile before we see a parallel Open Source development tract.

A complicating factor for an Open Source effort is that Spanner includes the use of GPS and Atomic clock hardware. Software only projects tend to be the most successful. Hopefully we'll see clouds step it up and start including higher value specialized services. A cloud wide timing plane should be a base feature. But we are still stuck a little bit in the cloud as Internet model instead of the cloud as a highly specialized and productive software container.

Another complicating factor is that as Masters of Disk it's not surprising Google built Spanner on top of a new Distributed File System called Colossus. Can you compete with Google using disk? If you go down the Spanner path and commit yourself to disk, Google already has many years lead time on you and you'll never be quite as good. It makes more sense to skip a technological generation and move to RAM/SSD as a competitive edge. Maybe this time Open Source efforts should focus elsewhere, innovating rather than following Google?

Related Articles

- Cloudant Labs: On Google Spanner by Mike Miller
- Thread on G+ by Jeff Dean
- SPANNER : GOOGLE'S GLOBALLY DISTRIBUTED DATABASE by Srihari Srinivasan
- On Slashdot
- On Hacker News
- F1 - The Fault-Tolerant Distributed RDBMS Supporting Google's Ad Business
- Spanner: Google's Globally-Distributed Database by Rodrigo De Castro
- Google Spans Entire Planet With GPS-Powered Database by Cade Metz
- Facebook Tackles (Really) Big Data With 'Project Prism' by

Cade Metz

- Alex Lloyd - Building Spanner
