CHANNEL

This article is more than **1 year old**

Google File System II: Dawn of the Multiplying Master Nodes

A sequel two years in the making

Cade Metz Wed 12 Aug 2009 // 02:12 UTC

UPDATED As its custom-built file system strains under the weight of an online empire it was never designed to support, Google is brewing a replacement.

Apparently, this overhaul of the Google File System is already under test as part of the "Caffeine" infrastructure the company announced earlier this week.

In an interview with the Association for Computer Machinery (ACM), Google's Sean Quinlan says that nearly a decade after its arrival, the original Google File System (GFS) has done things he never thought it would do.

"Its staying power has been nothing short of remarkable given that Google's operations have scaled orders of magnitude beyond anything the system had been designed to handle, while the application mix Google currently supports is not one that anyone could have possibly imagined back in the late 90s," says Quinlan, who served as the GFS tech leader for two years and remains at Google as a principal engineer.

But GFS supports some applications better than others. Designed for batch-oriented applications such as web crawling and indexing, it's all wrong for applications like Gmail or YouTube, meant to serve data to the world's population in near real-time.

"High sustained bandwidth is more important than low latency," read the original GPS research paper. "Most of our target applications place a premium on processing data in bulk at a high rate, while few have stringent response-time requirements for an individual read and write." But this has changed over the past ten years - to say the least - and though Google has worked to build its public-facing apps so that they minimize the shortcomings of GFS, Quinlan and company are now building a new file system from scratch.

With GFS, a *master node* oversees data spread across a series of distributed *chunkservers*. Chunkservers, you see, store chunks of data. They're about 64 megabytes apiece.

The trouble - at least for applications that require low latency - is that there's only one master. "One GFS shortcoming that this immediately exposed had to do with the original single-master design," Quinlan says. "A single point of failure may not have been a disaster for batch-oriented applications, but it was certainly unacceptable for latency-sensitive applications, such as video serving."

In the beginning, GFS even lacked an automatic failover scenario if the master went down. You had to manually restore the master, and service vanished for up to an hour. Automatic failover was later added, but even then, there was a noticeable service outage. According to Quinlan, the lapse started out at several minutes and now it's down to about 10 seconds.

Which is still too high.

"While these instances - where you have to provide for failover and error recovery - may have been acceptable in the batch situation, they're definitely not OK from a latency point of view for a user-facing application," Quinlan explains.

But even when the system is running well, there can be delays. "There are places in the design where we've tried to optimize for throughput by dumping thousands of operations into a queue and then just processing through them," he continues. "That leads to fine throughput, but it's not great for latency. You can easily get into situations where you might be stuck for seconds at a time in a queue just waiting to get to the head of the queue."

GFS dovetails well with MapReduce, Google's distributed data-crunching platform. But it seems that Google has jumped through more than a few hoops to build BigTable, its (near) real-time distributed database. And nowadays, BigTable is taking more of the load.

"Our user base has definitely migrated from being a MapReduce-based world to more of an interactive world that relies on things such as BigTable. Gmail is an obvious example of that. Videos aren't quite as bad where GFS is concerned because you get to stream data, meaning you can buffer. Still, trying to build an interactive database on top of a file system that was designed from the start to support more batch-oriented operations has certainly proved to be a pain point."

Next page: **The trouble with file counts**

Page: **1** 2 Next →