Facebook at 13 Million Queries Per Second Recommends: Minimize Request Variance

Thursday, November 4, 2010 at 8:48AM

Todd Hoff in MySQL, Strategy, facebook

Facebook gave a MySQL Tech Talk where they talked about many things MySQL, but one of the more subtle and interesting points was their focus on controlling the variance of request response times and not just worrying about maximizing queries per second.

But first the scalability porn. Facebook's OLTP performance numbers were as usual, quite dramatic:

- Query response times: 4ms reads, 5ms writes.
- Rows read per second: 450M peak
- Network bytes per second: 38GB peak
- Queries per second: 13M peak
- Rows changed per second: 3.5M peak
- InnoDB disk ops per second: 5.2M peak

Some thoughts on creating quality, not quantity:

- They don't care about average response times, instead, they want to minimize variance. Every click must be responded to quickly. The quality of service for each request matters.
- It's OK if a query is slow as long as it is always slow.
- They don't try to get the highest queries per second out of each machine. What is important is that the edge cases are not the bad.

- They figure out why the response time for the worst query is bad and then fix it.
- The performance community is often focussed on getting the highest queries per second. It's about making sure they have the best mix of IOPs available, cache size, and space.

To minimize variance they must be able notice, diagnose, and then fix problems:

- They measure how things work in operation. They can monitor at subsecond levels so they catch problems.
- Servers have **miniature fractures** in their performance which they call "stalls." They've built tools to find these.
- Dogpile collection. Every second it notices if something is wrong and ships it out for analysis.
- Poor man's profiler. Attach GDB to servers to know what's going on, they can see when stalls happen.
- Problems are usually counter-intuitive this can never happen type problems.
  - Extending a table locks the entire instance.
  - Flushing dirty pages was actually blocking.
  - How statistics get sampled in InnoDB.
  - Problems happen on medium-loaded systems too. Their systems aren't that loaded to ensure quality of service, yet the problems still happen.
- Analyze and understand every layer of the software stack to see how it performs at scale.
- Monitoring system monitors different aspects of performance so they can notice a change in performance, drill down to the host, then drill down to the query that might be causing the problem, then kill the query, and then trace it back to the source file where it occurred.
- They have a small team, so they make very specific changes to

Linux and MySQL to support their use cases. Longer term changes are made by others.

Please watch the MySQL Tech Talk for more color and details.

Related Articles

- Good Hacker News Thread