Quantcast File System (QFS)

Faster, more efficient Hadoop processing

Big data requires big investments – hard disks for storage, computers for processing, datacenters to house them, electricity to run and cool them, and people's time to write jobs (and wait for them to finish). These costs multiply quickly, making efficiency paramount.

In 2006, Quantcast was a young startup setting out to measure the web's audiences, using an early version of Hadoop, and needing a more efficient, cost-effective data-processing stack. We began to invest in efficiency innovations for Hadoop, including a streamlined file system that is now production hardened and ready to be shared with the community.

The **Quantcast File System (QFS)** is a distributed file system developed as an alternative to Apache Hadoop's HDFS. It's 100% open-source, plug-in compatible with Hadoop, and delivers significantly better performance for batch data processing while simultaneously offering substantial disk-space savings (and hence computer, datacenter and electricity savings too).

QFS Key Innovations

More Processing. Less Hardware. More compact data storage means fewer hard drives to purchase and power. Faster data throughput means faster results. Here's how QFS delivers both:

- **Reed-Solomon (RS) error correction.** Unreachable machines and dead disk drives are the rule rather than the exception on a large cluster. Therefore, tolerating missing data is critical. HDFS uses triple replication, which expands data storage requirements 3x. QFS uses Reed-Solomon encoding, a commonly used error correction technique for CDs and DVDs, which offers superior data recovery power and yet only requires a 50% data expansion. Thus, QFS requires only half the storage of HDFS for equivalent capacity.

- **Faster writes.** Leaner data encoding doesn't just save disk space, it means less data to write. Since every job on QFS writes only half as much physical data, it puts half as much load on the cluster. Jobs write data faster, and more can run at the same time.

- **Faster reads.** A hard drive is the slowest component in a cluster, with a top read speed of about 50 MB/s. HDFS reads each data block from a single drive and therefore inherits the same speed limit. QFS reads every block from six drives in parallel, making its top theoretical read speed 300 MB/s. This translates into a significant speed boost for real-world jobs.

- **Direct I/O.** The fastest way to read data from (or write data to) a drive is in large, sequential bursts. Normal file I/O APIs permit the operating system to buffer data and swap disk time between different processes, which breaks up big, efficient bursts into small, inefficient ones. QFS uses low-level APIs that enable more control to ensure that disk access stays optimal.

- **Fixed memory.** QFS is implemented in C++ and carefully manages its own memory within a fixed footprint. That means fast operations, with no interrupts for garbage collection. It's also a good neighbor to other processes on the same machine, as it never asks the OS for more memory at the risk of swapping and extra disk activity. QFS's memory management helps keep performance high and administration simple.

quantcast

Benchmark Results

Our benchmarks show how performance compares with Hadoop's HDFS at scale on our own systems. The metaserver benchmark exercises the hub of the file system, which manages directory structure and oversees operations. QFS's lean native implementation demonstrates clear advantage.

The 20 TB benchmark compares performance of read-only and write-only Hadoop jobs on a real-world cluster, using different file systems. The write job ran 75% faster using QFS due to having less data to write. The read job ran 47% faster, primarily because better parallelism shortened the delays caused by straggling workers.

Metaserver (Head Node)

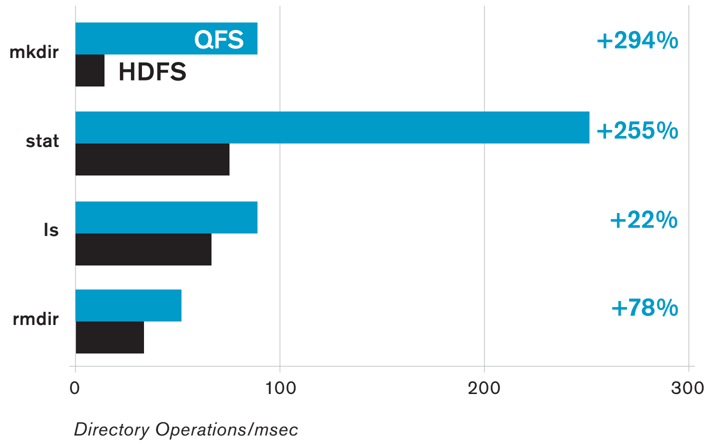

Figure 1: Comparison of directory management on QFS metaserver and HDFS head node, based on total throughput achieved by 512 clients building, inspecting, or deconstructing a balanced tree totaling 75.7 million directories. Dual E5-2670, 64GB RAM, HDFS version 2.0.0-cdh4.0.0 (Cloudera 4.0).

20TB Hadoop Job

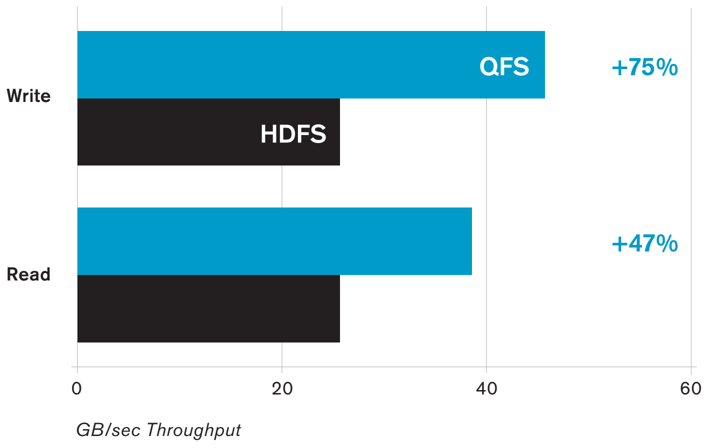

Figure 2: File system benchmark based on average throughput reading or writing 20TB uncompressed data across 6500 drives on a mixed network of 1–10Gbps links. Apache HDFS version 1.0.2.

Proven Reliability

Reliability is critical for a file system and earned only through time and hard work in a suitably demanding environment. Quantcast is such an environment. We receive over 40 terabytes of new data daily, and our daily map-reduce processing can exceed 20 petabytes.

QFS has grown up with us. It evolved from the Kosmos File System (KFS), which we adopted in 2008 and started using for secondary storage. We learned many lessons and made many improvements over the years. In 2011, we committed fully when we stopped using HDFS and migrated our primary processing to QFS. Since then we've read or written over 4 exabytes to it. We're confident it's ready for other users' production workloads.

Features at a Glance

- Fault-tolerant petabyte-scale storage
- Feedback-directed chunk allocation
- Coordinated restarts
- Unix-style permissions
- User logging
- Direct I/O
- Fixed-footprint memory management
- Efficient multi-client append mode
- Reed-Solomon error correction
- File replication
- Hadoop compatible

Get QFS

- Free
- Open source under Apache 2 license
- Binaries prebuilt for 64-bit Centos 6.2, Debian 6.0, OSX 10.8, Ubuntu 11.04

Download

http://quantcast.github.com/qfs

Questions

qfs@quantcast.com