Summer 2015 Qserv Release

Major Milestones

Improved Qserv Stability, Concurrency, Latency and Performance

Qserv stability has been dramatically improved. We demonstrated we can run Qserv under heavy load with no problems, no crashes and no memory leaks for extended periods of times (e.g. ~48 hours, we did not try longer). During the tests we continuously ran over large number of queries (over 100 simultaneous queries at any given time, with peaks above 200). We tested the limits, and uncovered one limitation which will be addressed during S16 cycle (returning very large results from a query results in overly high CPU and memory consumption)

Qserv concurrency has been greatly improved as well. It was the first time we were able to demonstrate Qserv successfully serving hundreds of simultaneous queries. In past tests the highest we successfully handled was under 10.

We demonstrated big improvements in the query scheduler. In the past scheduling and answering even the simplest query such as selecting one row out of multi-billion row table sharded across thousands of chunks required over a second. During S15 tests we demonstrated a response time below 100 miliseconds. When running a large number of concurrent queries, Qserv was able to handle simultaneous 50 low-volume queries and 5 high-volume queries, with average response time

- under 1 sec for low-volume queries (vs required 10 sec),
- ~15 min for high volume Object scans (vs required 60 min in baseline, 24 hours in KPI planned for S15),
- ~57 min for Source scans (vs required 12 hours in baseline, 24 hours in KPI planned for S15)

These tests were run on data set equivalent to 10% of the 1st Data Release Catalog.

The tests are documented in more details in S15 Large Scale Tests.

(DM-2854, DM-1961, DM-1962, DM-2189, DM-2638, DM-2806, DM-3362, DM-3390, DM-3038, DM-3262, DM-3038, DM-3451, DM-3432, DM-1706)

Major User-Facing Functionality and Interface Changes

Improved Query Coverage

The following classes of queries were previously not-supported or broken, and have been fixed in this release:

- queries with results larger than 2 MB
- queries with results that involve BINARY and BIT columns
- queries that repeat column name in SELECT, for example, "SELECT Object.id, Source.id FROM..."
- queries that involve ORDER BY for partitioned tables
- queries that selects rows by objectId from non-director table

(DM-1847, DM-1692, DM-1841, DM-1540, DM-1541, DM-1715, DM-854, DM-2858, DM-3104, DM-2864)

Added Support for Query Cancellation

Query interruptions initiated by users by pressing Ctrl-C in the mysql client are now handled by Qserv. In the past, interrupting a query with Ctrl-C in mysql client resulted in killing Qserv. We are still aware of some issues which require a redesign of Executor, this will be handled in W16.

(DM-1087, DM-1716, DM-2177, DM-1947)

Major Non-User Facing Functionality and Interface Changes

Refactored XRootD-qserv communication (xrd oss) and migrated Qserv to ssi v2

Reworked and documented interfaces between XRootD and Qserv.

(DM-2294, DM-2451, DM-2547, DM-1676, DM-2643)

Worker Management

Designed and built tools for managing workers.

(DM-1900, DM-2176, DM-2419, DM-1411, planned: DM-1901, DM-2420)

Database and Table Management

Added code/tools for managing distributed databases and tables: creating, deleting, loading data. Information about distributed databases and tables is kept in the Zookeeper based Qserv Central State System (CSS). This relies on the worker management service.

(DM-1896, DM-2385, DM-1897, DM-2622, DM-2623, DM-2624)

Query Management

Built a system for keeping track of asynchronous queries, including their status; the information is kept in mysql. It is not yet connected with the core of Qserv because additional work scheduled for W16 is needed (dynamic updates of CSS instead of static snapshot).

(DM-1488, DM-2804, Dm-2805, DM-3249, DM-3090)

Central State System v2

Revisited CSS and decided to replace zookeeper with MySQL-based solution. Implemented MySQL-based KVInterface, and added support for updates. Also added code that coordinates CSS updates with Query Metadata updates. CSS will be migrated away from Zookeeper in W16.

Multi-Node Integration Test Suite

Extended Qserv integration test suite to support multi-node testing.

(DM-998, DM-2175, DM-2412, DM-2748, DM-2679)

Bug Fixes

Fixed problems in xrootd discovered in multi-node tests

- Allow the mwfiles option to be specified by path.
- Include mwfiles as a sufficient specification for cms.space.
- Do not hold the request lock when calling a request callback method.
- Make sure to remove session reference in all defered tasks in
- Make sure messages are properly sequenced
- Async handling in XRootDMsgHandler

(DM-3261)

Fixed six different race conditions

(DM-2777, DM-2779, DM-2708, DM-2710, DM-2681, DM-2481)

Fixed several memory and connection leaks

(DM-2698, DM-2716, DM-2762, DM-2714, DM-2703)

Fixed problem with Qserv related to restarting mysql

(DM-2930)

Fixed crash in data loader related to creating tables

(DM-2417)

Fixed code related to repeated column names in SELECT, for example, "SELECT Object.id, Source.id FROM..."

(DM-854)

Fixes code related to selecting rows by objectId from non-director table

(DM-2864)

Build and Code Improvements

Qserv Configuration

Scripts for Qserv configuration have been improved and simplified. Location of mysql data directory has been migrated away from the qserv_rundir to prevent accidental deletion of the (typically very valuable) data set during reinstallation of qserv run directory. Added qserv-restart.sh

(DM-2519, DM-2595, DM-3238)

Unit Testing

Designed and built framework for unit-testing two large components of Qserv: controller module and query execution). Previously they could not be tested in isolation.

(DM-201, DM-202)

Improved Error Handling in Qserv

Implemented proper error checking in UserQueryFactory from QuerySession objects. Improved confusing error messages.

(DM-2390, DM-2885)

Improved Data Loader

Improvements to data loader which was introduced Winter 2015 release:

- engine type was corrected
- index type for overlap tables was fixed (unique --> non unique).
- the loader is now documented

(DM-2617, DM-2618, DM-2190)

Adoption of C++11

Qserv code now built with C++11 features enabled in the C++ compiler, and the code has been modified to begin to use these.

(DM-2469, DM-2497, DM-2715, DM-2729, DM-2711, DM-2712, DM-2720, DM-2953)

Code Cleanup

Improvements to logging, removed unnecessary namespace qualifiers, removed unnecessary white spaces, spellings

(DM-3110, DM-2730, DM-2430, DM-3091, DM-219)

Qserv Documentation

Revisited and updated Qserv documentation, refreshed existing contents, removed obsolete sections and documented parts previously not-documented. Wrote documentation for the new Xrootd SSI interface.

(DM-2468, DM-1676, DM-2841)

Research and Prototyping

Data Distribution

A thorough research was conducted to determine how to distribute qserv-managed data and keep track of the qserv distributed data. Centralized and distributed hash table options have been considered. The latter was selected as a more suitable solution. We built a proof-of-concept prototype. The prototype will be turned into a production version during the upcoming cycle.

(DM-2020, DM-2022, DM-1996, DM-1997, DM-1998, DM-1999, DM-2000, DM-2001, DM-2002, DM-2021, DM-2507, DM-2508, DM-2843)

Qserv Authorization

Explored how to handle authorization in Qserv.

(DM-1803)