

Winter 2016 Qserv and Webserv Release

- Major User-Facing Functionality and Interface Changes
 - Butler
 - Fixed query cancellation and responses to various legal and illegal SQL queries
- Major Non-User-Facing Functionality and Interface Changes
 - Shared scans to speed up large queries
 - Switched from zookeeper to mysql
 - Switched from mysql to mariadb
 - Improved xrdssi API
- Build and Code Improvements
 - Reworked Db module, including switching to SQLAlchemy back-end
 - Added support for distributed database and table creation/deletion
 - Added support for dynamic CSS metadata
 - Modernized Qserv code
 - Moved sphgeom to dedicated module
 - Improved Qserv build system
 - Replaced XML-RPC with in-process communication
 - Added unit tests to Webserv
 - Added support for OS X
- Research and Prototyping
 - Data Provenance
 - Data distribution and replica management
 - Secondary index
 - Technologies for Data Access and Database
 - Asynchronous ("background") queries
 - Distributed data loading

Major User-Facing Functionality and Interface Changes

Butler

A major rework of the Butler Framework started. The work include:

- added support for multiple input and output repositories
- added support for repositories without an sqlite3 registry
- added support for datasetType aliases
- improved butler configuration
- improved spatial images search
- started work on data repository selection based on version
- updated the documentation

The work will continue in Summer 16 through DM-4341 epic.

(DM-2404: DM-4544, DM-4625, DM-4682, DM-4683, DM-4365, DM4171, DM-4170, DM-3591, DM-3566, DM-3504, DM-4168, DM-3472)

Fixed query cancellation and responses to various legal and illegal SQL queries

- Added support for query cancellation
- Fixed queries involving "objectId BETWEEN", "objectId IN (...)"
- Fixed JDBC - Qserv and sqlalchemy - Qserv problems

(DM-3263, DM-1708, DM-2873, DM-2887, DM-1982, DM-3555, DM-3456, DM-4648, DM-4197)

Major Non-User-Facing Functionality and Interface Changes

Shared scans to speed up large queries

Added support for shared scans for single table scans, and synchronized scans for multiple tables joined together.

(DM-2077)

Switched from zookeeper to mysql

CSS data is now stored in mysql database instead of Zookeeper server. This reduces architectural complexity of the whole system and removes one heavy-weight component of the system. This should improve long-term stability of the system and reduce dependency on external projects.

(DM-3506)

Switched from mysql to mariadb

Switched qserv and the entire LSST DM stack from mysql to mariadb.

(DM-224, DM-5122, DM-4705, DM-4642, DM-4808, DM-4806)

Improved xrdssi API

xrdssi can now send a small amount of data (e.g. qserv result protobuf header) in the initial reply. This means an xrootd client/server round-trip can be removed from every Qserv xrootd request.

(DM-2314)

Build and Code Improvements

Reworked Db module, including switching to SQLAlchemy back-end

(DM-2513, DM-2558, DM-4648)

Added support for distributed database and table creation/deletion

First implementation of the asynchronous mechanism for dropping databases and tables on every worker node based on CSS information. New watcher service implemented.

(DM-2802)

Added support for dynamic CSS metadata

Table metadata is now retrieved directly from CSS (previously it was contained in CSS snapshot) which allows us to dynamically create/drop tables and databases without restarting czar process.

(DM-3506)

Modernized Qserv code

Passes made through the entire Qserv codebase to cut over to various C++11 features consistently and address compiler warnings. Qserv now compiles warning-free on g++ 4.9, g++ 5.1, and clang 700-1.

(DM-2956, DM-3803, DM-4757, DM-4617)

Moved sphgeom to dedicated module

sphgeom library sources was previously included directly in the Qserv source tree; now the recently-provided lsst package is used instead.

(DM-2178, DM-2946)

Improved Qserv build system

Improved packaging of shared libraries. Improved sconscripts

(DM-3447, DM-2421)

Replaced XML-RPC with in-process communication

Qserv is now implemented as a Lua extension module loaded by mysql-proxy and it runs now in the same process with proxy. This reduces architectural complexity and replaces complicated network data exchange between proxy and qserv with in-process data exchange.

(DM-4348)

Added unit tests to Webserv

(DM-3672)

Added support for OS X

(DM-3662, DM-4529, DM-3898, DM-3902, DM-4165, DM-4470)

Research and Prototyping

Data Provenance

Revisited provenance design, built a standalone proof-of-concept prototype.. Documented the data provenance architecture. The provenance can be found [here](#).

(DM-2042)

Data distribution and replica management

A prototype C++ distributed hash table package was developed, based on the design of Pastry/PAST.

(DM-2089)

Secondary index

Researched and prototyped secondary index. Identified MySQL InnoDB engine as sufficient to meet secondary index performance requirements on a single multi-core host (< 2 days to load 40 billion entries demonstrated on a 4 core laptop)

(DM-2119)

Technologies for Data Access and Database

Researched MaxScale as possible replacement of MySQL Proxy. Researched Serf, Consul, and MemSQL.

(DM-1648)

Asynchronous ("background") queries

Understood how disruptive the changes related to implementing asynchronous queries will be for Qserv.

(DM-2136)

Distributed data loading

Researched all the needs, requirements and constraints, and explored what the best architecture for a distributed loader would be.

(DM-2088)