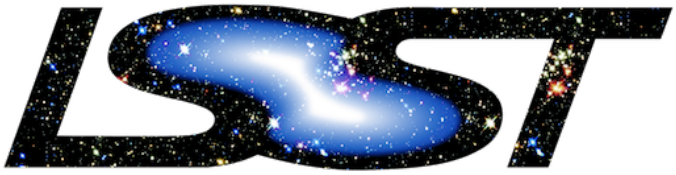

LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

LVV-P46 (2018 Qserv Large Scale Testing) Test Plan and Report

Fritz Mueller

DMTR-71

Latest Revision: 2019-07-15

Abstract

This is the test plan and report for LVV-P46 (2018 Qserv Large Scale Testing), an LSST level 2 milestone pertaining to the Data Management Subsystem.

Change Record

Version	Date	Description	Owner name
	2019-07-03	Draft	Fritz Mueller
1.0	2019-07-11	Test campaign completed. Document issued.	Fritz Mueller
1.1	2019-07-15	Fixed title capitalization	Gabriele Comoretto

Document curator: Fritz Mueller
Document source location: `https://github.com/lsst-dm/DMTR-71`
Version from source repository: e093def

Contents

LVV-P46 (2018 Qserv Large Scale Testing) Test Plan and Report

1 Introduction

1.1 Objectives

Yearly functional and scale performance testing of the Qserv distributed database system. Establishes Qserv's viability on growth curve toward full production scale.

1.2 System Overview

Qserv is a SQL-oriented MPP distributed database system built by LSST for the purpose of hosting LSST catalog data products. Qserv is tested yearly at large scale, on test datasets of ever-increasing size, to ensure that development remains on a path toward delivering a system that functions effectively at LSST release scales.

1.3 Applicable Documents

LDM-555: LSST Data Management Database Requirements
LDM-135: LSST Data Management Database Design
LDM-552: LSST Data Management Distributed Database Software Test Specification

1.4 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P46 Jira Test Plan and related Test Cycles (LVV-C81).

Section 1 provides an overview of the test campaign, the system under test (Distrib Database), the applicable documentation, and explains how this document is organized. Section 2 describes the configuration used for this test. Section 3 describes the necessary roles and lists the individuals assigned to them.

Section 4 provides a summary of the test results, including an overview in Table 1, an over-

all assessment statement and suggestions for possible improvements. Section 5 provides detailed results for each step in each test case.

The current status of test plan LVV-P46 in Jira is Completed.

1.5 References

[1] **[LDM-555]**, Becla, J., 2017, *Data Management Database Requirements*, LDM-555, URL https://ls.st/LDM-555

[2] **[LDM-135]**, Becla, J., Wang, D., Monkewitz, S., et al., 2017, *Data Management Database Design*, LDM-135, URL https://ls.st/LDM-135

[3] **[LDM-552]**, Mueller, F., 2017, *Qserv Software Test Specification*, LDM-552, URL https://ls.st/LDM-552

2 Test Configuration

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

Qserv testing at scale requires a dedicated machine cluster:

- 25 to 50 ”worker” nodes, each with on order 16 GB memory and on order 10 TB locally attached storage
- 1 to 2 ”czar” nodes, minimally provisioned as above, but preferably provisioned with more RAM and several TB of fast SSD storage

Suitable test clusters exist and have been used at both NCSA and CC-IN2P3. Some testing at scale has also been conducted with dynamically provisioned clusters on the Google cloud infrastructure.

A test dataset of appropriate size (per schedule in LDM-552) is also required.

3 Personnel

The following personnel are involved in this test activity:

- Test Plan (LVV-P46) owner: Fritz Mueller
- Test Cycles:
  - LVV-C81 owner: Fritz Mueller
    * Test case LVV-T1017 tester: Fritz Mueller
    * Test case LVV-T1085 tester: Fritz Mueller
    * Test case LVV-T1087 tester: Fritz Mueller
    * Test case LVV-T1086 tester: Fritz Mueller
    * Test case LVV-T1088 tester: Fritz Mueller
    * Test case LVV-T1089 tester: Fritz Mueller
    * Test case LVV-T1090 tester: Fritz Mueller
- Additional Test Personnel involved: None

4 Overview of the Test Results

4.1 Summary

Test Cycle **LVV-C81: 2018 Qserv Large Scale Testing**			
test case	**status**	**comment**	**issues**
LVV-T1017	Pass	Qserv docker containers installed without issue.	
LVV-T1085	Pass	Short queries executed correctly in less than required times.	
LVV-T1087	Pass	Queries executed correctly in less than required times.	
LVV-T1086	Pass	Queries executed correctly in less than the required times.	
LVV-T1088	Pass	Test results indicate the desired less-than-linear scaling rate for scans of each type, within limits of machine resource exhaustion.	
LVV-T1089	Conditional Pass	Load test excluded Object x (Source, ForcedSource) joins; see notes in ”Overall Assessment” section.	
LVV-T1090	Conditional Pass	Heavy Load test excluded Object x (Source, ForcedSource) joins and was limited by maximum shared-scan load that could be accommodated by the current test tooling; see notes in ”Overall Assessment” section.	

Table 1: Test Results Summary

4.2 Overall Assessment

A performance problem was observed with execution of Object x Source and Object x ForcedSource joins during this testing. While these joins performed per expectation in isolation, and full-table-scans performed per expectation in isolation, the combination of both under high loads resulted in significantly degraded performance.

The development team spent a good deal of time investigating this regression. It was found by running against older codebases that the regression was independent of changes to either the Qserv codebase or the underlying MariaDB database engine. Indeed, when historic code

and datasets were run on the same hardware systems which had been used for performance testing previously, at the same scale as had been run previously, this same join+scans performance degradation was evident.

The development team has concluded that changes to the operating environment (OS kernel, firmware) must be implicated, since these are the only remaining variables which could not be rolled back in the efforts to duplicate previous performance measurements. Investigations continue, and the issue is expected to be resolved before the next round of large scale testing.

Additionally, the heavy load test taxed the existing test tooling and single-master configuration their limits, due to increased result set sized, overheads from 50% more scans, limits on simultaneous database connections from the test script, etc. For the next round of testing, a more robust test harness will need to be developed, and multiple head nodes will be required to accommodate the increased result-handling loads.

In the meantime, concurrent joins were disabled during the load tests, and the existing tooling was run to as high a load as possible for the heavy load test, in order to gather/document as much meaningful performance assessment as possible in the current operating environment. Since indications are that the Qserv codebase itself is not implicated in the join performance regression, and Qserv otherwise appears to be healthy and performing per design, the load tests have been marked with a ”conditional pass”.

4.3 Recommended Improvements

- Test tooling to be revamped to support higher concurrency in database connections and to gracefully handle larger cumulative result sizes.
- Subsequent test regimes will require more than a single master node.

5 Detailed Test Results

5.1 Test Cycle LVV-C81

Open test cycle *2018 Qserv Large Scale Testing* in Jira.

2018 Qserv Large Scale Testing
Status: Done

This test cycle establishes that:

1. Qserv functional query requirements are met,
2. Qserv's shared scan infrastructure performs per design, and
3. Qserv meets query response requirements under load, with data at scale of 30% DR1 data volume.

5.1.1 Software Version/Baseline

Qserv built from git SHA 06cdeda75 (published to docker hub as qserv/qserv:travis_DM-13961)

5.1.2 Configuration

Hardware

- 50 nodes:
  - DELL PowerEdge R620 (Dell Spec Sheet) for nodes 1-25
  - DELL PowerEdge R630 (Dell Spec Sheet) for nodes 26-50
- 2 x Processors Intel Xeon E5-2603v2 @ 1.80 Ghz 4 core
- 10 MB cache, 6.4 GT/s, 80W
- Memory 16 GB DDR-3 @ 1600MHz (2x8GB)
- 2 x hard drive 250GB SATA 7200 Rpm 2,5" hotplug (OS)
- 8 x hard drive 1 TB Nearline SAS 6 Gbps 7200 Rpm 2,5" hotplug (DATA)
- 1 x card RAID H710p with 1 GB nvram

- 1 x card1 GbE 4 ports Broadcom® 5720 Base-T
- 1 x card iDRAC 7 Enterprise

Dataset Information

Table	Row Count	.MYD size [TB]	.MYI size [TB]
Object	5,662,102,056	6.86	0.15
Source	104,440,271,322	49.4	5.7
ForcedSource	515,549,769,246	16.4	12.9

Total MySQL data dir size: 93.6 TB

DR1 numbers are available in Document-16168 under “Data Releases”

Object, Source and ForcedSource are at slightly less than ~30% of DR1 level due to some empty chunks generated erroneously during the duplication phase. This difference is marginal and will not affect test results.

5.1.3 Test Cases in LVV-C81 Test Cycle

5.1.3.1 Test Case LVV-T1017 - Qserv Preparation

Open *LVV-T1017* test case in Jira.

Before running any of the performance test cases, Qserv must be installed on an appropriate test cluster (e.g. the test machine cluster at CC-IN2P3). To upgrade Qserv software on the cluster in preparation for testing, follow directions at http://www.slac.stanford.edu/exp/lsst/qserv/2015_10/HOWTO/cluster-deployment.html.

The performance tests will also require an appropriately sized test dataset to be synthesized and ingested, per the yearly dataset sizing schedule described in LDM-552, section 2.2.1. Tools for synthesis of ingest of test datasets may be found in the LSST GitHub repot at https://github.com/lsst-dm/db_tests_kpm*. Detailed use and context information for

the tools is described in https://jira.lsstcorp.org/browse/DM-8405.

It has also been found that the Qserv shard servers must have engine-independent statistics loaded for the larger tables in the test dataset, and be properly configured so that the MariaDB query planner can make use of those statistics. More information on this issue is available at https://confluence.lsstcorp.org/pages/viewpage.action?pageId=58950786.

Preconditions:

Execution status: **Pass**

Final comment:
Qserv docker containers installed without issue.

Detailed step results:

Step		Description, Results and Status
1	Description	Install/upgrade Qserv on a test cluster, following directions at http://www.slac.stanford.edu/exp/lsst/qserv/2015_10/HOW-TO/cluster-deployment.html
	Expected Result	Qserv installed
	Actual Result	Qserv installed and testing performed at CC-IN2P3, on nodes ccqserv100 - ccqserv124.
	Status	Pass
2	Description	Synthesize and load and appropriately sized test dataset per the yearly dataset sizing schedule described in LDM-552, section 2.2.1. Tools for synthesis of ingest of test datasets may be found in the LSST GitHub repot at https://github.com/lsst dm/db_tests_kpm*. Detailed use and context information for the tools is described in https://jira.lsstcorp.org/browse/DM-8405.
	Expected Result	Test dataset loaded

Actual Result	Test dataset loaded as database LSST30.
Status	Pass

5.1.3.2 Test Case LVV-T1085 - Short Queries Functional Test

Open *LVV-T1085* test case in Jira.

The objective of this test is to ensure that the short queries are performing as expected and establish a timing baseline benchmark for these types of queries.

Preconditions:
QSERV has been set-up following procedure at LVV-T1017.

Execution status: **Pass**

Final comment:
Short queries executed correctly in less than required times.

Detailed step results:

Step		Description, Results and Status
1	Description	Execute single object selection: **SELECT** * **FROM** Object **WHERE** deepSourceId = 9292041530376264 and record execution time.
	Expected Result	Query runs in less than 10 seconds.
	Actual Result	Execution time 0.23 sec.
	Status	Pass

2	Description	Execute spatial area selection from Object: **SELECT COUNT(*) FROM** Object **WHERE** qserv_areaspec_box(316.582327, −6.839078, 316.653938, −6.781822) and record execution time.
	Expected Result	Query runs in less than 10 seconds.
	Actual Result	Execution time 3.34 sec.
	Status	Pass

5.1.3.3 Test Case LVV-T1087 - Full Table Joins Functional Test

Open *LVV-T1087* test case in Jira.

The objective of this test is to ensure that the full table join queries are performing as expected and establish a timing baseline benchmark for these types of queries.

Preconditions:
QSERV has been set-up following procedure at LVV-T1017.

Execution status: **Pass**

Final comment:
Queries executed correctly in less than required times.

Detailed step results:

Step	Description, Results and Status

1	Description	Execute query: **SELECT** o.deepSourceId, s.objectId, s.id, o.ra, o.decl **FROM** Object o, Source s WHERE o.deepSourceId=s.objectId **AND** s . flux_sinc **BETWEEN** 0.3 **AND** 0.31 and record execution time.
	Expected Result	Query expected to run in less than 12 hours.
	Actual Result	Query executed in ~112 min.
	Status	Pass
2	Description	Execute query: **SELECT** o.deepSourceId, f.psfFlux **FROM** Object o, ForcedSource f **WHERE** o.deepSourceId=f.deepSourceId **AND** f . psfFlux **BETWEEN** 0.13 **AND** 0.14 and record execution time.
	Expected Result	Query expected to run in less than 12 hours.
	Actual Result	Query executed in ~5 hr.
	Status	Pass

5.1.3.4 Test Case LVV-T1086 - Full Table Scans Functional Test

Open *LVV-T1086* test case in Jira.

The objective of this test is to ensure that the full table scan queries are performing as expected and establish a timing baseline benchmark for these types of queries.

Preconditions:
QSERV has been set-up following procedure at LVV-T1017.

Execution status: **Pass**

Final comment:
Queries executed correctly in less than the required times.

Detailed step results:

Step		Description, Results and Status
1	Description	Execute query: **SELECT** ra , decl , u_psfFlux , g_psfFlux , r_psfFlux **FROM** Object **WHERE** y_shapeIxx **BETWEEN** 20 **AND** 20.1 and record execution time and output size.
	Expected Result	Query expected to run in less than 1 hour.
	Actual Result	Query executed in ~20 min, 83MB output.
	Status	Pass
2	Description	Execute query: **SELECT** COUNT(*) **FROM** Source **WHERE** flux_sinc **BETWEEN** 1 **AND** 1.1 and record the execution time
	Expected Result	Query expected to run in less than 12 hours.
	Actual Result	Query executed in ~104 min.
	Status	Pass

3	Description	Execute query: **SELECT** COUNT(*) **FROM** ForcedSource **WHERE** psfFlux **BETWEEN** 0.1 **AND** 0.2 and record the execution time
	Expected Result	Query expected to run in less than 12 hours.
	Actual Result	Query executed in ~48 min.
	Status	Pass

5.1.3.5 Test Case LVV-T1088 - Concurrent Scans Scaling Test

Open *LVV-T1088* test case in Jira.

This test will show that average completion-time of full-scan queries of the Object catalog table grows sub-linearly with respect to the number of simultaneously active full-scan queries, within the limits of machine resource exhaustion.

Preconditions:

1. A test catalog of appropriate size (see schedule detail in LDM-552, section 2.2.1), prepared and ingested into the Qserv instance under test as detailed in LVV-T1017.
2. The concurrency load execution script, runQueries.py, maintained in the LSST Qserv github repository here: https://github.com/lsst/qserv/blob/master/admin/tools/docker/deployment/ir

Execution status: **Pass**

Final comment:
Test results indicate the desired less-than-linear scaling rate for scans of each type, within limits of machine resource exhaustion.

Detailed step results:

Step		Description, Results and Status
1	Description	Repeat steps 2 through 5 below, where ”pool of interest” is taken first to be ”FTSObj” and subsequently ”FTSSrc”:
	Expected Result	At end of each pass, a graph indicating scan scaling rate and machine resource exhaustion cutoff.
	Actual Result	Object • 2 scans: ~20 min • 5 scans: ~20 min • 10 scans: ~22 min • 20 scans: ~25 min • 40 scans: ~72 min (machine resource exhaustion) ForcedSource: • 2 scans: ~ 51 min • 4 scans: ~ 61 min • 10 scans: ~270 min (machine resource exhaustion)
	Status	Pass
2	Description	Inspect and modify the CONCURRENCY and TARGET_RATES dictionaries in the runQueries.py script. Set CONCURRENCY initially to 1 for the query pool of interest, and to 0 for all other query pools. Set TARGET_RATES for the query pool of interest to the yearly value per table in LDM-552, section 2.2.1.
	Expected Result	rueQueries.py script updated with appropriate values for test iteration
	Actual Result	Appropriate edits made.
	Status	Pass
3	Description	Execute the runQueries.py script and let it run for at least one, but preferably several, query cycles.
	Expected Result	Test script executes producing log file.

	Actual Result	Script executed per design.
	Status	Pass
4	Description	Examine log file output and compile performance statistics to obtain a growth curve point for the pool of interest for the test report.
	Expected Result	Logs indicate either successful test run, providing another growth point for curve, or errors indicating machine resource exhaustion cutoff has been reached.
	Actual Result	Script executions and log contents as expected.
	Status	Pass
5	Description	Adjust the CONCURRENCY value for the pool of interest and repeat from step 3 to establish the growth trend and machine resource exhaustion cutoff for the query pool of interest to an acceptable degree of accuracy.
	Expected Result	Average query execution time for full scan queries of each class should be demonstrated to grow sub-linearly in the number of concurrent queries to the limits of machine resource exhaustion.
	Actual Result	Test indicated less-than-linear scaling rate for scans of each type, within limits of machine resource exhaustion.
	Status	Pass

5.1.3.6 Test Case LVV-T1089 - Load Test

Open *LVV-T1089* test case in Jira.

This test will check that Qserv is able to meet average query completion time targets per query class under a representative load of simultaneous high and low volume queries while running against an appropriately scaled test catalog.

Preconditions:

QSERV has been set-up following procedure at LVV-T1017

Execution status: **Conditional Pass**

Final comment:
Load test excluded Object x (Source, ForcedSource) joins; see notes in ”Overall Assessment” section.

Detailed step results:

Step		Description, Results and Status
1	Description	Inspect and modify the CONCURRENCY and TARGET_RATES dictionaries in the runQueries.py script. Set CONCURRENCY and TARGET_RATES for all pools to the yearly value per table in LDM-552, section 2.2.1.
	Expected Result	Script updated with appropriate values.
	Actual Result	Script updated without difficulty.
	Status	Pass
2	Description	Execute the runQueries.py script and let it run for 24 hours.
	Expected Result	Script runs without error and produces output log.
	Actual Result	Script ran and log files were generated per expectation.
	Status	Pass
3	Description	Examine log file output and compile average query execution times per query type; and compare to yearly target values per table in LDM-552, section 2.2.1.
	Expected Result	Average query times per query type equal or less than corresponding yearly target values in LDM-552, section 2.2.1.

Actual Result	Query through-put over 24 hours: • 625,245 Low Volume queries finished — Baseline: 604,800 • 201 Object scans — Baseline: 192 • 4 Source scans — Baseline: 4 • 4 ForcedSource scans — Baseline: 4 • 0 Object-Source joins (see ”Overall Assessment” section) — Baseline: 8 • 0 Object-ForcedSource joins (see ”Overall Assessment” section) — Baseline: 4 • 53 NearNeighbor queries — Baseline: 48 Average query times: • Low Volume queries 2.92 sec/query — Baseline: under 10 sec. • Object scans 44.5 min/query — Baseline: under 1 hour • Source scans 5.4 hr/query — Baseline: under 12 hours • ForcedSource scans 5.4 hr/query — Baseline: under 12 hours • Object-Source joins (not measured; see ”Overall Assessment” section) — Baseline: under 12 hours • Object-ForcedSource joins (not measured; see ”Overall Assessment” section) — Baseline: under 12 hours • NearNeighbor queries 38.9 min/query — Baseline: under 12 hours
Status	Conditional Pass

5.1.3.7 Test Case LVV-T1090 - Heavy Load Test

Open *LVV-T1090* test case in Jira.

This test will check that Qserv is able to meet average query completion time targets per query class under a higher than average load of simultaneous high and low volume queries while running against an appropriately scaled test catalog.

Preconditions:
QSERV has been set-up following procedure at LVV-T1017

Execution status: **Conditional Pass**

Final comment:
Heavy Load test excluded Object x (Source, ForcedSource) joins and was limited by maximum

shared-scan load that could be accommodated by the current test tooling; see notes in "Overall Assessment" section.

Detailed step results:

Step		Description, Results and Status
1	Description	Inspect and modify the CONCURRENCY and TARGET_RATES dictionaries in the runQueries.py script. Set CONCURRENCY and TARGET_RATES for LV query pool to 2020 value per table in LDM-552, section 2.2.1. Set CONCURRENCY and TARGET_RATES for all other query pools to values in next column over from current year column (or to 2020 values +10% if year is 2020) per table in LDM-552, section 2.2.1.
	Expected Result	Script updated with appropriate values.
	Actual Result	Script updated without difficulty.
	Status	Pass
2	Description	Execute the runQueries.py script and let it run for 24 hrs.
	Expected Result	Script runs without error and produces output log.
	Actual Result	Script ran, but did not finish expected number of scans due to proxy/thread crashes. This appears to be a limitation of the test script and the mysqlproxy front end.
	Status	Conditional Pass
3	Description	Examine log file output and compile average query execution times per query type.
	Expected Result	Average query times per query type equal or less than corresponding yearly target values in LDM-552, section 2.2.1.

Actual Result

Query through-put over 24 hours:

- 779,308 Low Volume queries — Baseline: 691,200
- 188 Object scans — Baseline: 288
- 4 Source scans — Baseline: 6
- 4 ForcedSource scans — Baseline: 6
- 0 Object-Source joins — Baseline: 12
- 0 Object-ForcedSource joins — Baseline: 6
- 53 NearNeighbor queries — Baseline: 72

Average query times:

- Low Volume queries 3.67 sec/query — Baseline: under 10 sec
- Object scans 44 min/query — Baseline: under 1 hour
- Source scans 4.8 hr/query — Baseline: under 12 hours
- ForcedSource scans 4.8 hr/query — Baseline: under 12 hours
- Object-Source joins (not measured; see ”Overall Assessment” section) — Baseline: under 12 hours
- Object-ForcedSource joins (not measured; see ”Overall Assessment” section) — Baseline: under 12 hours
- NearNeighbor queries 39 min/query — Baseline: under 12 hours

Status Conditional Pass