

# Database Meeting 2022-08-24

## Date

24 Aug 2022

## Attendees


- Igor Gaponenko Fritz Mueller Fabrice Jammes Andy Salnikov Joanne Bogart Andy Hanushevsky John Gates





## Notes from the previous meeting:

- [Database Meeting 2022-08-17](#)

## Discussion items

Discussed	Item	Who	Notes
✓	Project news	Fritz Mueller	<ul style="list-style-type: none"><li>the last DMLT was focused on the upcoming reviews</li><li>another "hot" topic is the DM all-hands meeting in Chile (preliminarily March 13-17, 2023):<ul style="list-style-type: none"><li>Yusra sent around the questionnaire for those who may be interested in participating (online versus in-person, arriving the weekend before or staying the weekend after)</li><li>It's formally for DM "construction" (not for "operation") which might be an issue for some of us.</li></ul></li></ul>
✓	NCSA to SDF (S3DF) migration	Igor Gaponenko Fritz Mueller	<p>Igor Gaponenko on the status of the test Qserv instance:</p> <ul style="list-style-type: none"><li>got 6 nodes for Qserv, finalized the configuration</li><li>unpacked and deployed a snapshot of Qserv instance <b>large6</b> that was taken at NCSA around May 17th</li><li>the snapshot does not include <b>DP02</b> (the catalog would need to be ingested)</li><li>Qserv hasn't been started as I need to do some work on the tooling</li></ul> <p>Fritz Mueller there is an interest to set the TAP service at RSP</p>

	<p>Status of the Qserv integration tests</p> <p>(technical discussion)</p>	<p>team</p>	<p><a href="#">Igor Gaponenko</a> on the current status:</p> <ul style="list-style-type: none"> <li>• a collection of tables (5 databases) within Qserv source tree within <code>itest_src</code></li> <li>• a purpose of many queries is not well understood (or documented) with documentation links pointing to TRAC             <ul style="list-style-type: none"> <li>◦ <a href="#">Fritz Mueller</a> we may still have the original TRAC pages migrated to Confluence</li> <li>◦ Action item: need to document each query</li> </ul> </li> <li>• some queries are meant to test the non-existing functionality of Qserv (some sort of the "wish list" for future improvements?)             <ul style="list-style-type: none"> <li>◦ <a href="#">Fritz Mueller</a> those might be based on the initial survey of what functionality was expected from Qserv</li> </ul> </li> <li>• some may exist for testing the home-grown SQL parser (before migrating to ANTLR4)             <ul style="list-style-type: none"> <li>◦ <a href="#">Fritz Mueller</a> some might be added as bugs were discovered in the parser, or for bugs in the query rewriter tests</li> </ul> </li> <li>• about 50% of the test queries are presently disabled (marked as <i>FIXME</i>, etc.)</li> <li>• some of those (disabled tests) are needed to cover the current functionality of Qserv</li> <li>• some might be disabled when migrating the tests to the new <i>lite</i> container or because the required functionality wasn't present in the Replication/Ingest system at the time of the migration</li> <li>• there is quite a bit of duplication between the tests (and catalogs)</li> <li>• some data ( CSV ) files are compressed, while others aren't. It's not clear why and what it's meant to test.</li> <li>• only 3 (out of 5) catalogs are presently tested</li> </ul> <p>Conclusions:</p> <ul style="list-style-type: none"> <li>• it's a bit of a mess in there</li> <li>• Qserv coverage is not complete (or excessive) in some areas</li> <li>• the <b>BIGGEST</b> problem (for myself) was with using very specific table names that imply certain semantics in the context of LSST ("Object", "Source", etc.). Although the initial motivation behind that decision is clear, this naming convention is presenting a big obstacle in understanding <b>what</b> is actually being tested in Qserv. The semantics of some LSST tables has been changed since the original Data Model.</li> </ul> <p>The proposal to be discussed:</p> <ul style="list-style-type: none"> <li>• revisit the test cases</li> <li>• eliminate duplicates and obsolete tests</li> <li>• add in the missing tests</li> <li>• come up with the Qserv-specific naming convention for the tables to reflect their role within Qserv ("director", "child", "ref-match", "fully-replicated", etc.)</li> <li>• refine the table schemas (and data) to leave only the essential columns (required for Qserv and for the referential integrity of the schemas), and add a few of the "payload" columns as needed where tests require row selection based on those values (shared scans, testing the <i>WHERE</i> clause etc.)</li> </ul> <p><a href="#">Fritz Mueller</a>:</p> <ul style="list-style-type: none"> <li>• some tables should carry the semantics (time series queries, etc.). So we do need a way to keep the semantics (at least) for some)</li> <li>• we could document those within the source tree using RST</li> <li>• we need to do a systematic revisit of the tests to see what's missing</li> </ul> <p><a href="#">Fritz Mueller</a> Add the micro dataset to be automatically deployed with Qserv before running any integration tests. This dataset could be used for basic (interactive?) testing of Qserv after it gets deployed.</p> <p><a href="#">Igor Gaponenko</a> proposed to implement the synthetic dataset generator (driven by <i>YAML</i>) as an alternative for the present collection of static test catalogs. The dataset would be generated by the Python script at the run-time of the integration test. Or, it could be pre-generated if needed. This option allows generating catalogs of any scale (number of databases, tables, columns in the tables, the number of rows). Also:</p> <ul style="list-style-type: none"> <li>• the test queries could be generated by the script accordingly</li> <li>• this technique could be also used for the small-to-mid term scalability &amp; performance testings of Qserv</li> </ul>
---	--	-------------	---

	Status of qserv-operator	<a href="#">Fabrice Jammes</a>	<p>Intermittent problems when loading DP02 into <code>qserv-dev</code> have been observed at IDF. The first class of problems was found to be caused by Google's NAT service configuration for the outbound connections. That was causing failures when pulling contributions from IN2P3 into IDF.</p> <p>The second class of problem might be caused by the worker pods restarted during the ingest. The restarts were resulting in changes in the IP addresses of the restarted pods. This was confusing the ingest workflow that was caching IP addresses of the workers at the beginning of each transaction.</p> <ul style="list-style-type: none"> <li>• <a href="#">Fritz Mueller</a> has proposed to extend the worker registration protocol (model) of the Replication/Ingest system with the DNS entries of the workers captured by the worker ingest services themselves and reported to the worker registry. Then the ingest workflow would be given an option to use the IP addresses or the DNS entries. <a href="#">Igor Gaponenko</a> has registered the following JIRA ticket addressing the issue: <div data-bbox="516 388 1302 556" data-label="Complex-Block">  <a href="#">DM-36005</a> - Jira project doesn't exist or you don't have permission to view it. </div> </li> <li>• Though, we still have the transient problem during worker's pod restarts. The DNS entries of the workers would disappear from the Kubernetes DNS service. In order to deal with this, <a href="#">Fabrice Jammes</a> would need to reinforce the implementation of the workflow to resolve the DNS entries (or IP addresses of the workers) before submitting the contributions. Should any problems be seen at this stage, the workflow could take proper actions (wait before the DNS entry would show up again, or request the new sets of the IP addresses from the Replication Controller).</li> <li>• <a href="#">Igor Gaponenko</a> an alternative approach would be to extend the Replication Controller to allow sending all ASYNC requests to the Controller and let the Controller take care of distributing these requests between the relevant workers. <ul style="list-style-type: none"> <li>◦ Action item: discuss this idea with <a href="#">Fabrice Jammes</a> and <a href="#">Fritz Mueller</a> and make a JIRA ticket if approved.</li> </ul> </li> </ul> <p><a href="#">Fabrice Jammes</a> is not seeing these problems in IN2P3 (<code>k8s</code>-based Qserv deployments) where the workers are being run on the beefy hardware. <a href="#">Fabrice Jammes</a> is going to bump (by a factor of 2) the amount of memory available to the workers in IDF (<code>qserv-dev</code>) to see if that would help to workaround the issue (prevent the restarts).</p> <p>In the meantime, <a href="#">Igor Gaponenko</a> will be working on the improved version of the worker ingest services to avoid the very origin of the problem (memory accumulation by the services).</p> <p>Also discussed a plan to improve the logger configuration for the services in the <code>qserv-operator</code>-based deployments. <a href="#">Fritz Mueller</a> has made the following JIRA ticket in tis context:</p> <div data-bbox="470 953 1258 1121" data-label="Complex-Block"> <ul style="list-style-type: none"> <li>•  <a href="#">DM-36004</a> - Jira project doesn't exist or you don't have permission to view it.</li> </ul> </div>
	Query cancellation		<p>Context:</p> <ul style="list-style-type: none"> <li>• We began discussing this problem at the previous meeting <a href="#">Database Meeting 2022-08-17</a></li> </ul> <p><a href="#">Fritz Mueller</a> reported the updates:</p> <ul style="list-style-type: none"> <li>• has discovered there is the LUA hook allows to intercept these events at the <code>lua-proxy</code> level</li> <li>• The origin of the 8 hours timeout is still not known. Setting timeouts at the level of MariaDB and the proxy didn't help</li> <li>• Theories: kernel TCP, keep-alive timeout, or something else</li> <li>• will continue investigating</li> </ul>

## Action items

