

# Database Meeting 2015-09-09

## Date

09 Sep 2015

## Attendees

- [Brian Van Klaveren](#), [Fabrice Jammes](#), [John Gates](#), [Vaikunth Thukral](#), [Andy Salnikov](#), [Unknown User \(npease\)](#), [Andy Hanushevsky](#), [Jacek Becla](#)
- [Kian-Tat Lim](#) for part of the meeting

## Discussion items

### Fabrice's visit

- tentative dates for Fabrice visit at SLAC: Nov 2-13
- one of the tasks to focus on during that time: dockerizing Qserv, running Qserv integration tests inside CI
- thought: invite someone from SQUARE (Josh?) and/or NCSA (Bill?) for ~3 days during that time

### docker

- Fabrice making great progress with running Qserv in docker, see eg DM-3199
- SL on IN2P3 cluster too old to support decent version of docker, working on upgrading it

### gcc version

- planning to stop carrying about gcc 4.4 and start using newer C++ features (in qserv)

### Query size

- ok to limit the query definition (sql) to 2 MB. Currently allowing unlimited, but complicates xrootd code

### query status

- can rely on the fact it is possible to send multiple queries inside the same session. So after chunk query is dispatched, just send queries requesting status. Response can be inside metadata (which is faster due to less overhead)

### clean exit from xrootd

- need proper destructors to be called, e.g., to do proper cleanup in mysql in-memory tables
- difficult because of multiple threads running
- idea: force unload() of our plugin

### retrying failed chunk-queries

- we need to mark query results (add something like queryId and retryId to rows or blocks of rows), so that we don't end up with duplicates when we resubmit partially failed chunk-query

### large results

- difficult because we want/need to load balance across czars based on result size, which is unknown when we dispatch query
- maybe introduce dedicated czars for handling large results?
- or save results locally on worker nodes and then decide how to handle results based on result size
- likely it will not be an issue, probably @2GB for large queries is ~ near limit, and we won't have many such queries finishing at the same time
- btw, mysql imposes 1 GB limit for results, eg., one can't fetch more than 1 GB using "select \* from results"

### interactive / background queries

- queries that are on shared scans don't necessarily have to be all "background" queries. They can be interactive, especially useful if we are streaming partial results

### shared scans

- planning to dynamically skip chunks if query is too expensive (and come back to skipped chunks in next scan). This will require keeping track of processed chunks per query (bitmap)
- issue to think about: L3 brings chunk that is mark as "empty" in LSST production tables. This is problematic because we will have that chunkId turned off