Database Meeting 2015-11-11

Date

11 Nov 2015

Attendees

- Brian Van Klaveren, Unknown User (npease), Vaikunth Thukral, Fabrice Jammes, John Gates, Andy Salnikov, Kian-Tat Lim, Jacek Becla

Discussion items

latest code/data incompatibility

- at IN2P3 cluster: need to migrate metadata because we switched from zookeeper to mysql
- what happens when we lose CSS info
  - can't reproduce based on data
  - rely on backup / replicas
- will need tools to check consistency data < – > css

Docker

- problems with starting cmsd inside docker, AndyH provided fix
- docker work - deliverable / milestone: recipe explaining how to run integration tests inside docker on open stack using vagrant + containers

authorization

- NCSA team asked Brian for standalone dax service for testing authentication

dax for sui

- need to enable ports
- need to load / make available some consistent set of data (images + catalog)

image search

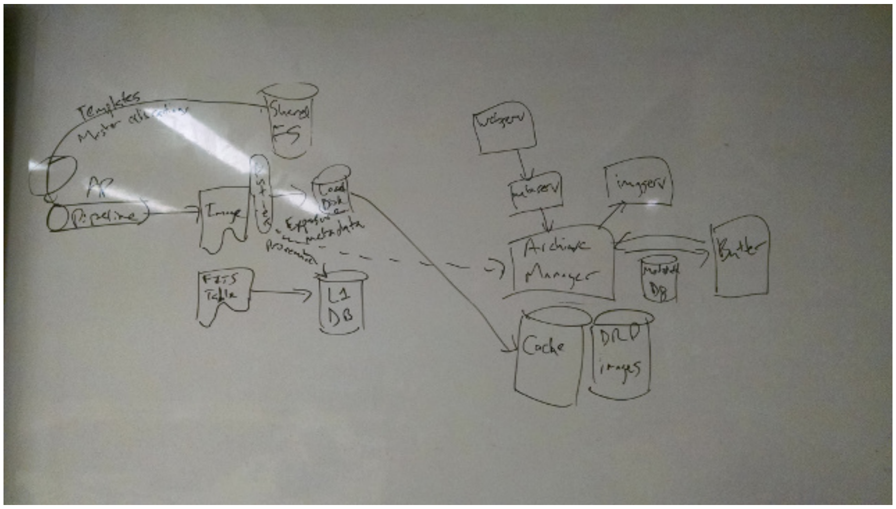
- Serge checking out memsql - in memory, supports spherical geometry. But not free, startup. Unclear what the cost is
- create story for that work, attach to epic DM-1648

greenplum

- open source now, worth checking out
- need good benchmark to really compare
- can use the queries we run for large scale tests this summer
- also, might want to checkout hsqldb (research version of voltdb)

image archive

- need to plan epics / metaepics
- writing images from AP ... local disk? shared fs? object store? Prefer shared fs over object store. Probably don't want to serve images to users from the location where AP writes. So writing to local disk might be best
- when images saved, need to save provenance. It is either via butler "put", or separate step. Better via butler, it is more atomic that way
- in butler put, it should be possible to turn off writing metadata, we have a usecase where someone might want to write just images say on a laptop
- need to understand how to hook provenance to butler
- butler probably does not need to add to provenance, hopefully not
- querying provenance: preexisting "query templates", user fills bits and pieces, we expand it to internal queries that know how to query provenance. That can be implemented through plugin to butler
- usecase: we push alerts, but fail to push images and/or catalog data. Later we reprocess that thing that failed. Obey observing order, also, have to remember processing order! So, provenance needs to keep track of processing order for AP (this is not the case for DRP, as Jim Bosch mentioned in the thread on community.lsst.org)
- archive manager
  - access via buter, webserv, ingestion daemon that ingests data from pipeline. Maybe exposed to users for ingesting images
- also, see picture of whiteboard related to that discussion:



Templates
Master calibrations
Shared FS
AP
Pipeline
Image
Butler
Local Disk
Exposure metadata
Provenance
FITS Table
L1 DB
webserv
metaserv
imgserv
Archive Manager
metadata DB
Butler
Cache
DRP images