Data Access Hangout 2014-12-01

Date

01 Dec 2014

Attendees

- Serge Monkewitz
- Unknown User (xiuqin)
- Tatiana Goldina
- Unknown User (zhang)
- John Gates
- Fabrice Jammes
- Kian-Tat Lim
- Andy Salnikov
- Jacek Becla

Goals

- Discuss Data Archive related topics

Discussion items

Status of Qserv setup at IPAC

- new issues related to c-style comments around query text (see: DM-1601)
- request account for Jacek at ipac (if easy) (Xiuqin will)
- setup up qserv at SLAC not urgent if we can get a server at NCSA in the next 2-3 months
- long term: need tools to quickly publish based on the tip of the master

Test data set(s)

- Serge exporting data set used for IPAC-based qserv to NCSA

SUI-DataAccess testing in W15:

- SUI - Qserv now
- SUI - Image metadata ~January

Preserving comments in Metadata store?

- preserve (next-to-value) comments from FITS headers? Yes!
- preserve values for the "comment" keyword?
    - long term: yes. Short term: need to look how it is stored (John)

Duplicate keywords

- caused problems at IPAC in the past. Made assumption: take the last keyword. It worked for one project, not for the other
- document what we support (we take first or last etc)
- long term: give users tool that massage the headers if they are in non-supported format

Rebuilding headers

- tools don't have to rebuild exactly, enough if we rebuild the meaning

Image metadata

- exposure* tables in Qserv: distributed on each node (or maybe a small central shared file system)
- small scale db server, for all kinds of metadata including L3 image metadata
- Level 1. Metadata goes into metadata store within 24h
- try foreign tables to have everything in one place. If needed, it should be OK to make a copy of DR metadata (but hopefully DataCat will not require that, it requires it now)

Data set descriptions

- In the short term, need a mechanism to describe existing data sets at NCSA from past data challenges. Long term: will do that programmatically
- Official source of truth should be database
- Better to not rely on any text file, don't want file-based implementation
- So, build a form where people submit information. Input gets stored in database
    - allow updates etc, log all entries in case we need to recover
    - storing all that in lsst10 mysql is fine
- Filling the form will be required, if you don't fill the form, your data set will be thrown away
- This might grow into L3 import form

- What to ask on the form: location, type, purpose, responsible user, level of priority (scratch, short term keep, long term keep), backup or not
- Dependencies between data sets (like sym links) - should be taken care of automatically