Large Synoptic Survey Telescope

www.lsst.org

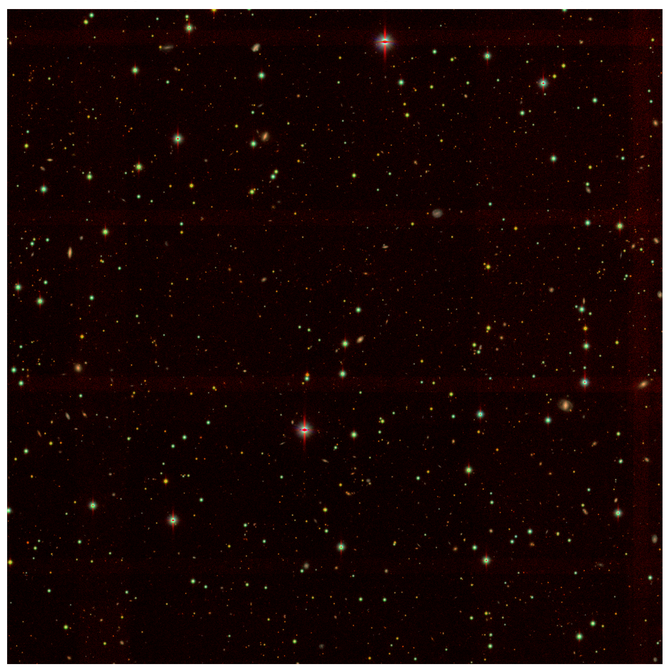
The LSST Data Processing Software Stack: Summer 2015 Release

Tim Jenness *(LSST Tucson)* for the LSST Data Management Team

Abstract The Large Synoptic Survey Telescope (LSST) is an 8-m optical ground-based telescope being constructed on Cerro Pachón in Chile. LSST will survey half the sky every few nights in six optical bands. The data will be transferred to NCSA and within 60 seconds they will be reduced using difference imaging techniques and detected transients will be announced to the community in the VOEvent format. Annual data releases will be made from all the data during the 10-year mission, with unprecedented depth of coadds and time resolution of catalogs for such a large region of sky. Here we present the current status of the data processing software, and describe how to obtain it.

The LSST Software Stack

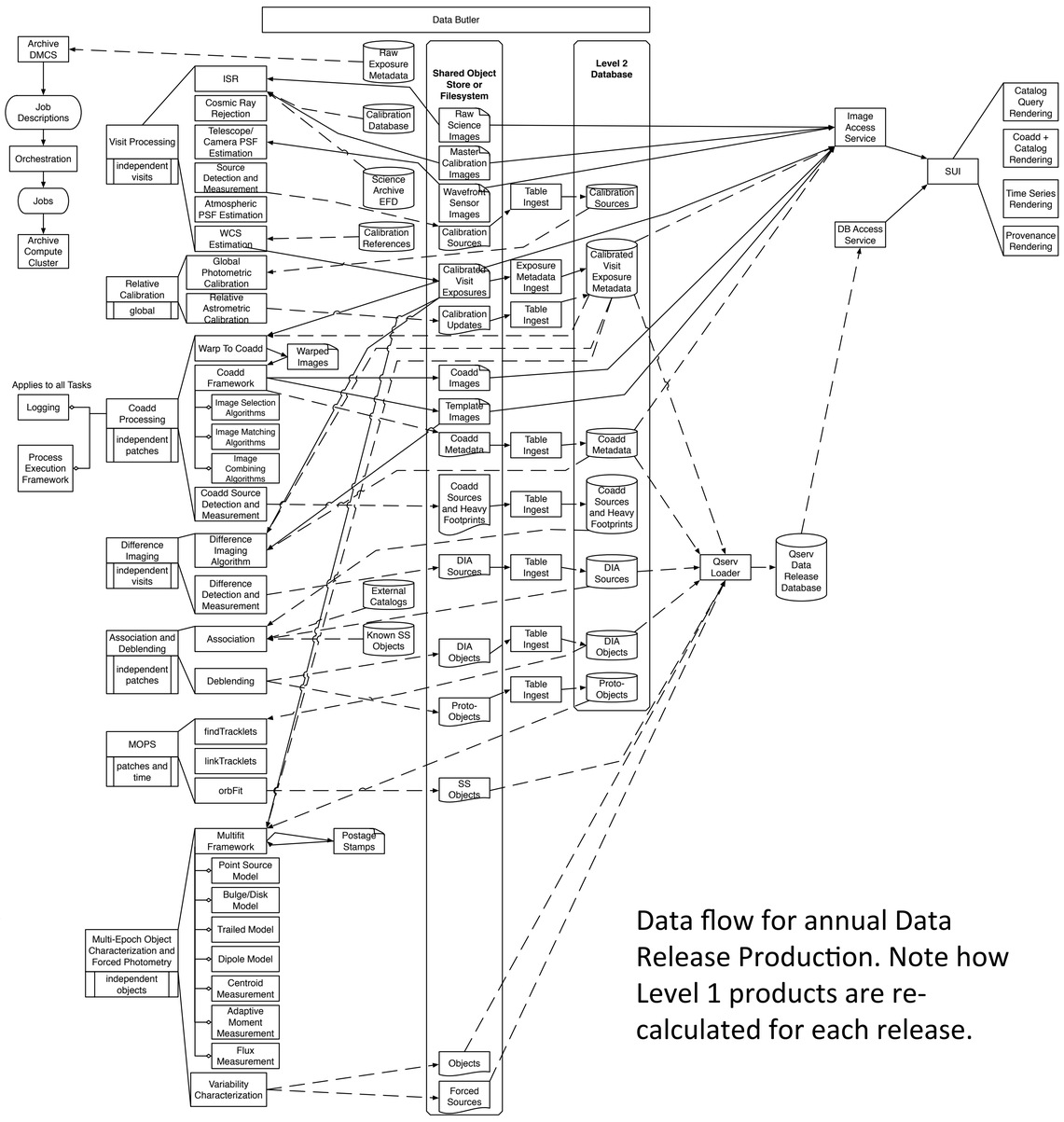
The Large Synoptic Survey Telescope will take about 15 TB of image data per night and after ten years of operations will have 15 petabytes of catalog data from the final data release, and 0.5 exabytes of image data. We are writing a suite of software packages to enable these data products to be created with sufficient quality and performance to meet the established science goals. The science pipeline software enables two key components of the data management system. The Alert Production pipelines (also known as *Level 1*) process the data from the telescope and publish alerts to the community within 60 seconds of data acquisition. Data Release Production (*Level 2*) is responsible for the annual data releases which reprocess all the data each year to generate the best possible catalogs. Both these systems are integrated with the Calibration Products Production that continuously calculates the best calibrations for the pipelines. The software also provides a toolkit for user-supplied code that can be used to efficiently and effectively analyze LSST data as part of *Level 3* processing or their own Pipelines.



Left: Three-colour (R, I, Y) composite image of an 11 by 11 arcminute patch of SDSS Stripe 82 based on Hyper Suprime-Cam engineering test data processed through the LSST stack Summer 2015 release.

Stack Summary

About 40 distinct packages.
100,000 lines of Python, 110,000 lines of C++.
Migrating to C++11 and support of Python 3.



Data flow for annual Data Release Production. Note how Level 1 products are re-calculated for each release.

Summer 2015 Release

Whilst the software is open source and can be installed at any time, LSST makes formal releases of the science pipeline software at the end of each six month development cycle in the spring and autumn. The most recent release covered the summer development cycle and was labeled *Summer 2015* and released in September 2015. Detailed release notes can be found at https://community.lsst.org/t/268; here we provide a summary.

Multi-band processing for coadds: New command-line tasks have been added for consistent multi-band processing of coadds. This new data processing flow carefully combines source measurements taken in multiple bands to guarantee consistent deblending across all bands, including when carrying out forced photometry, thereby enabling reliable color measurement, and ensuring that all sources are measured in each band, regardless of the bands where they are detected.

Upgraded astrometry calculation: Previously astrometry was calculated using astrometry.net code and related catalogs distributed by LSST. To improve flexibility in the code the astrometry fitter is now pluggable and now includes an alternative implementation.

Support for PSFex: PSFEx is currently the state of the art external package for point spread function (PSF) determination, used in projects such as DES. LSST wrappers were created such that PSFEx could be used as a plugin in place of the built in PSF determiner.

Aperture correction: Aperture corrections are now applied consistently throughout the stack.

More efficient handling of large footprints: A footprint defines the pixels associated with a particular source or blended sources. This release saw significant improvements in performance when using very large footprints.

Enable use of deblended heavy footprints in coadd forced photometry: Given the new multi-band processing for coadds we now have a reference catalog that is consistent across all bands. This catalog allows the use of the source's heavy footprints (A heavy footprint is a footprint that includes the pixel values) to replace neighbors with noise in forced photometry, thus providing deblended forced photometry and consistent deblending across all bands. This provides much better colors for blended objects as well as measurements for drop-out objects that do not get detected in the canonical band. This functionality has been enabled for forced coadd photometry.

Significant improvements in the table class: The AFW package has a native C++ implementation of a class for manipulating table data for handling the results of detections and measurements algorithms. This release comes with some major enhancements to the internals of *afw.table* and, in particular, much better support for compound fields (such as Right Ascension/Declination tuples).

Device independent displays: DS9 is no longer hard-wired into the software and the choice of display tool is now user configurable. The intention is for the next release to include support for the Firefly visualization tool. (See talk O10.1)

Stack Components

daf The Data Access Framework is responsible for mediating between the archive resources and the application writer. The pipeline code has a completely abstract view of file I/O and only has to know how to deal with data objects representing fundamental types such as exposures and tables. Currently FITS is the internal format but the system is designed such that the internal format could be changed to HDF5, for example, and no changes would have to be made to the science pipeline code. This abstraction of the files from the code protects us against shifts in format preferences.

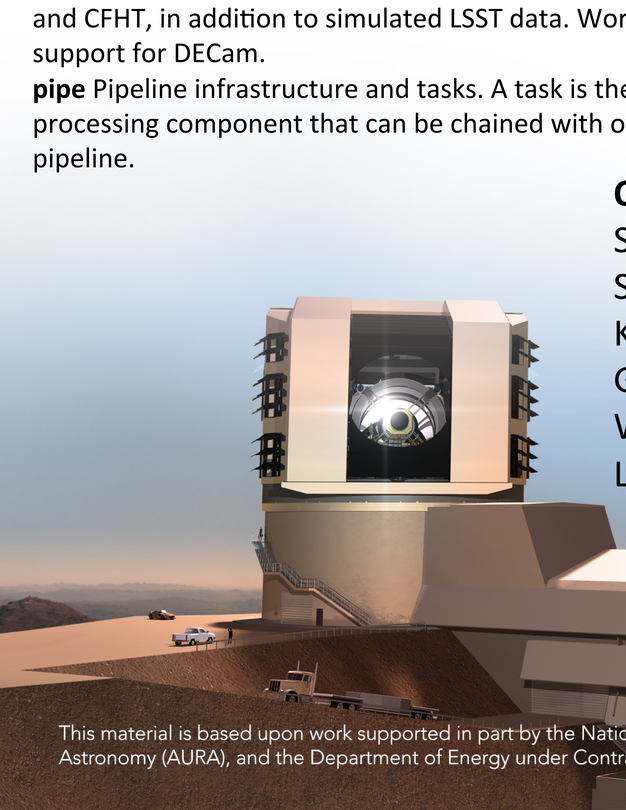
afw The Astronomy FrameWork provides the core classes for manipulating exposures and catalogs, including detecting sources and world coordinate handling.

ip These are the image processing classes, including packages for instrument signature removal and image differencing.

meas The measurement packages include code for determining source properties and correcting astrometry and photometry.

obs These classes provide instrument-specific knowledge to the software system, providing information to the data access framework to teach it how to interpret data from a range of optical cameras. The *obs* packages currently support data from some instruments on Subaru and CFHT, in addition to simulated LSST data. Work is ongoing to add support for DECam.

pipe Pipeline infrastructure and tasks. A task is the name for a core processing component that can be chained with other tasks to build a pipeline.



Obtaining the Software

Source code: https://github.com/LSST
Summer 2015 release: http://ls.st/0qg
Known to work on CentOS 6 and 7 & OS X Yosemite & Mavericks.
GCC 4.8.3 or later, or Apple clang 6, required.
We are experimenting with a CernVM-FS binary distribution for Linux.

More Information

LSST Design Overview: arXiv:0805.2366
LSST Science Book: arXiv:0912.0201
LSST Data Products: http://ls.st/LSE-163
DM Applications Design: http://ls.st/LDM-151
Key Numbers: http://lsst.org/scientists/keynumbers
Talks: DM Overview: O3.1; IPAC Firefly: O10.1

This material is based upon work supported in part by the National Science Foundation through Cooperative Support Agreement (CSA) Award No. AST-1227061 under Governing Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.