

#### **Reprocessing CHFT Images with the LSST DM Software Stack**

Dominique Boutigny – Nicolas Chotard Laboratoire d'Annecy le Vieux de Physique des Particules

Lyon Workshop – June 12-16, 2017







- CC-IN2P3 will provide 50% of the LSST data level 2 processing together with NCSA Satellite Data Release Processing
  - Requires a plan to ramp up infrastructure and local expertise
  - Successful Data Challenge in 2013 / validation of the S-DRP approach
    - Demonstrated that LSST stack is ~at the level of SDSS software
      - 1 M CPU hours 100 TB disk storage Millions of files
    - But complexity and some technical issues
      - Need to exercise the infrastructure on a regular basis
      - Understand I/O patterns Identify bottlenecks...
- A lot of expertise at IN2P3 on CFHT / Megacam data (SNLS)
- Considerable expertise on the DM stack developper side
  - But very little connexion with science collaborations
- Launched the Reprocessing Task Force in DESC
  - Try to attract more people
  - Extend to other instrument  $\rightarrow$  DLS SuprimeCam DECam ...





- Considerable amount of data available on CADC portal
  - Started with CFHT-LS data → now taken care by SN people (Ph. Gris, D. Fouchez, J.P. Reyes, et al.)
    - Deep fields comparable to LSST depth (r = 27.7)
    - Possibility to stack a really large number of visits (~500 in deep fields)
  - Many images on standard fields (Cosmos, Aegis, ...)
    - Combine analysis
    - Comparison to higher resolution data (HST)  $\rightarrow$  deblending performance
  - Focus on Galaxy Clusters and Ultra Deep Field (SXDS)
    - Alleviates some technical difficulties
      - Narrow field with relatively small number of images
      - Allows to reprocess the same field many times
    - Obvious connexion with science Cluster group wanted to get a stack based analysis pipeline
    - Published results as references (i.e. Weighing the Giants papers)
      - Authors in DESC
      - Aim at doing better than original pipeline





• 1 X 1 deg<sup>2</sup> – 0.187 arcsec / pixel – 13.5 micron / pixel



- The focal plane is flat but significant distortion beyond 20' radius
- 90° angle in the sky is ~89° in the corners of the focal plane (from P. Astier)
- Halos and reflections
- Filters (u, g, r, i, z) several sets : 3x i 2x (u,g,r,z)











- Start from a preliminary work from Simon Krughoff in 2013
  - Initial version of the obs\_cfht package
- LSST stack is able to handle raw images and to apply bias, dark, flat, and fringe corrections
  - The CFHT Elixir platform is doing a very good job → Start from preprocessed images (downloadable from CADC)
- Initial contributions :
  - Defects masks handling
  - Focal plane distortions polynomial (from P. Astier)





- Stack Astrometry has been rather difficult for years
  - Many failures Especially in u
  - Really difficult to track what was going on
  - Experienced code regression in some (rare) occasions
  - No real estimator of the quality of the astrometric solution







- A lot of improvements in the last year(ish)
  - New matcher implemented by R. Owen (Implementation of V. Tabur, 2007 algorithm)
  - New fitter implemented by J. Bosch (mathematically correct)
  - Improved source selector
  - The real game changer is the availability of new reference catalogs and the possibility to use a different catalog for Astrometry and Photometry
    - SDSS (32 GB), GAIA (270 GB), PanSTARRS (448 GB)
    - New butler compatible format
      - Easy generation of home made catalogs
      - Easy identification of catalog subset
      - No more "mmap" usage detected as an virtual memory abuse by some batch system (Grid Engine)
- Both GAIA and PanSTARRS give satisfactory results
  - Make me think that there is something odd with the SDSS catalog





- Joincal is a simultaneous astrometric fitter running on a set of exposures
  - Developed by P. Astier on SNLS software
    - Tested on CFHT/Megacam and HSC
  - Initial hacked implementation in the stack  $\rightarrow$  simastrom
  - Now being properly integrated and further developed by J. Parejko
    - Works for both astrometry and photometry
    - Able to handle full focal plane with (semi)fixed CCD positions
- Joincal is really fast (optimized sparse matrices algebra)
- Still some fine tuning to do to avoid some factorization failures





- Use a set of 6 visits (x36 CCD) spread over 4 years (2005 2008)
- Take the first visit as a reference
- Match the other visits to the reference
- Compute angular distance between "good" star-like sources and reference
- Compare individual astrometry (meas\_astrom) to jointcal







• Repeat with a set of visits taken during the same night



→ Extremely encouraging





Align with jointcal and coadd 194 CFHT u band visits in SXDS foot print ==> Successful despite of the low S/N on individual exposures







- Reprocess clusters which are in the SDSS footprint :
  - Weighing the Giants :
    - MACSJ2243.3-0935
    - CL0016+16
  - New :
    - 3C295
- PSFex CModel Joincal forcedPhotometry on coadds shapeHSM
- A few days on CC-IN2P3 batch cluster (not automated)
  - Main problems :
    - Astrometry (until recently)
    - Satellites / very bright stars  $\rightarrow$  deblender memory overflow
      - Solved by excluding sources with too large footprint size
    - Stack stability





Compute ellipticities from second moment and correct with PSF ellipticity







- Use CModel magnitudes for stars
- Correct for Milky Way extinction
- Compare to Covey, 2007 model

 $\rightarrow$  photometry looks reasonable







The produced butler-structured catalogs have nice features but are not so easy to distribute and manipulate in the DESC context

- Complete directory structure has to be replicated
- · Stack should be installed and understood

Nicolas has developed an independent package : https://github.com/nicolaschotard/Clusters.git

- Convert stack catalogs into astropy tables
- Save in hdf5 format

Astropy tables easily expendable with new quantities :

- Galactic extinction parameters
- Photo-z from several algorithms
- etc.

Keep only the quantities of interest  $\rightarrow$  USB key distributable !

By doing so, we were able to attract contributions from the Cluster group (C. Combet, D. Applegate, Ian Dell'antonio, H. Awan, A. Bervas,...)

## The Clusters pipeline



 $\rightarrow$  The Cluster group now has a usable stack-based pipeline to start with...





## Ghosts





## r band





# i band







We would like to use the cluster data to test the Qserv database from the end user scientist point of view

- A lot of expertise at IN2P3 with Fabrice Jammes
- The idea is simple :
  - → Store catalogs in Qserv
  - → Repeat clusters analysis
  - Compare with astropy

We will also do the same thing with our PDAC instance

Nicolas has already started some preliminary work

### → Hack session on Friday





- WtG used 52 clusters in their papers and we want to add more
- LSST will have 1000's
  - Handling production manually is not acceptable

Decided to use the SLAC workflow engine developed by Tony Johnson to automatically submit jobs at CC-IN2P3 (to start)



#### Task Summary

Task Filter:	Regular Expression (?) Active in Last 30 days V Latest Task Versions V Filter Reset Defaults											
		_	and the second sec									
Last Active 🍦	Task Name 🗧	туре 🗢	*	****	1/2 +	√ •	$\times$		$\bigcirc$		$\bigcirc$	Total
2017-06-09 07:05	ClustersDM	LSST	0	0	0	0	1	0	0	0	0	1
2017-06-09 05:25	EXOProcessData	EXO	0	0	0	5688	330	0	121	0	0	6139
2017-06-09 02:35	EXONightly-RHEL6-64	EXO	0	0	1	228	0	0	113	0	0	342
2017-06-08 23:26	DC1-DM12	LSST	0	0	1	0	5	0	5	0	0	11
2017-06-08 12:44	EXOGenericMonteCarlo	EXO	0	0	3	1088	521	0	224	0	0	1836
2017-05-28 07:25	imsim_deep_pipeline	LSST	0	0	0	24	0	0	0	0	0	24
2017-05-23 03:20	CDMSSuperSystemTests	CDMS	0	0	0	169	199	0	1	0	0	369
2017-05-20 08:39	DC1-phoSim-3	LSST	0	0	0	2401	0	0	0	0	0	2401
2017-05-16 06:01	BTest02	LSST	0	0	0	1	0	0	0	0	0	1
2017-05-16 00:36	BTest01	LSST	0	0	0	4	0	0	0	0	0	4
		Totals	0	0	5	9,603	1,056	0	464	0	0	





#### Task: ClustersDM

#### xml file: (download)

<?xml version="1.0" encoding="UTF-8"?> <pipeline xmlns="http://glast-ground.slac.stanford.edu/pipeline" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:schemaLocation="http://glast-ground.slac.stanford.edu/pipeline http://srs.slac.stanford.edu/Pipeline-II/schemas/2.1/pipeline.xsd"> <task name="ClustersDM" type="LSST" version="1.23"> <notation>A simple task for running the DM for reprocessing Clusters</notation> <variables> <!-- Job site and configuration--> <var name="JOBSITE">LSST-IN2P3</var> <var name="MAXCPU">10000</var> <var name="MAXCPULONG">10000</var> <var name="BATCH OPTIONS">-l os=cl7</var> <!-- Default location from which to setup DM --> <var name="DM RELEASE">w 2017 21</var> <var name="DM\_DIR">/sps/lsst/software/lsst distrib/\${DM RELEASE}</var> <var name="DM SETUP">loadLSST.bash</var> <!-- Base directory where script files will be placed --> <var name="SCRIPT LOCATION">ReprocessingTaskForce/workflows/CFHT/clusters</var> <var name="SLAC SCRIPT LOCATION">/u/ki/boutignv/ki19/\${SCRIPT LOCATION}</var> <var name="IN2P3 SCRIPT LOCATION">/sps/lsst/dev/lsstprod/\${SCRIPT LOCATION}</var> <!-- Script names --> <var name="SCRIPT NAME">ClustersWorkflow.py</var> <var name="BATCH NAME">ClustersBatch.sh</var> <!-- Base directory for input and output data --> <var name="VISIT DIR">/sps/lsst/data/clusters/MACSJ2243.3-0935/rawDownload</var> <var name="OUTPUT DATA DIR">/sps/lsst/data/clusters/workflow</var> </variables> <!-- ingest images -->

<!-- ingest images -->
<process name="ingest" site="\${JOBSITE}">

Nicolas now has a full working version

→ Simple usable tool, ok to start with, but we need to investigate more sophisticated workflow engine (Pegasus, ...)







Ultimately we would like to be able to process a given cluster in a fully automatic way

- $\rightarrow$  Give the cluster name
- → Download data
- → Run LSST pipelines
- → Produce catalogs

We would like to have this working on any weekly release (this is not the case yet...)

We also intend to use this automatic production tool to systematically test the weeklies : production and control plots (photo-z, cluster mass, ...)





- Working on CFHT reprocessing since ~4 years
- Acquired experience with the LSST stack from a (somewhat advanced) user point of view
- Well... this was not always easy... :-)
- But DM developers are responsive (Slack is a fantastic tool)
- I am convinced that we absolutely need to further develop the connexion between DM and Science
  - This requires a significant effort from the Science collaborations
  - → I am working actively on it in DESC

## IN2P3 certainly has a special role to play

...and we are always happy to play !