Using HTCondor to Calibrate and Archive HST and JWST Data

Matthew Burger &
The STScI Data Processing & Archive Services (DPAS) Branch

05.23.2018
expanding the frontiers of space astronomy

We help humanity explore the universe
with advanced space telescopes and archives
Our Strategic Goals

Excel in the science operations of NASA's current and future astrophysics flagship missions

Advance state-of-the-art astronomical data, archives, and tools for scientific discovery

Make the world's astronomical information accessible to all
Science Operations for the Hubble Space Telescope (HST)

- Proposal (observation) selection
- Planning and scheduling
- Data calibration
- Data archives
HST on Twitter @spacetelelive

Space Telescope Live @spacetelelive

What is @HubbleTelescope looking at? – Reference images from @STScI Digitized Sky Survey and Sloan Digital Sky Survey Data Release 13 (not actual Hubble images)

Low Earth Orbit
spacetelescopelive.org
Joined August 2016
Born on April 24, 1990

Tweet to Space Telescope Live

I am looking at the Galaxy NGC0613 with Wide Field Camera 3 for Dr. Peter Erwin.
spacetelescopelive.org/2018-05-21T12:...
Spacecraft & Science Operations for the James Webb Space Telescope (JWST)
(Scheduled for launch in Spring 2020)

- Proposal (observation) selection
- Planning and scheduling
- Data calibration
- Data archives
- Mission Operations Center
- Wavefront Sensing and Control
MAST Observations: Millions of observations from Hubble, Kepler, TESS, GALEX, IUE, FUSE, and more (to include JWST and WFIRST).

Virtual Observatory: Search thousands of astronomical data archives from around the world for images, spectra, and catalogs.

Hubble Source Catalog: A master catalog with a hundred million measurements of objects in Hubble images.

MAST Catalogs: Access to catalog data such as Gaia and TESS Input Catalog, with more coming soon.

All data is publicly available except for most HST data < 6 months old and JWST data < 12 months old.

http://mast.stsci.edu/
The data management system controls the flow of science and engineering data through the data calibration pipelines and into the archive (MAST).

Multiple data pipelines (workflows):

- New science from the telescope (four active instruments on HST)
- Reprocessed science data due to changes in software or calibration data
- Engineering data including spacecraft meta-data and jitter
- Mission schedules (planned observations)
- Spacecraft ephemerides
Data Calibration

**GOAL:** Convert an image from raw pixel counts on the detector to a source flux (number of photons emitted).

To solve this problem, one needs to account for:

- Detector bias: counts in a detector readout when no exposure is taken
- Dark current: spurious counts during an exposure due to noise in the detector
- Flat-field: pixel-to-pixel variations in detector efficiency
- CTE correction: pixel-to-pixel charge-transfer efficiency
- Combined/drizzled exposures: Long observations are broken into multiple exposures and combined in processing to increase the signal-to-noise ratio
Raw Data

Advance Camera for Surveys (ACS) image of M101 (Pinwheel Galaxy)

Two amplifiers with different bias levels
Final Product

Multiple images are combined to form the final product.

Cosmic ray hits have been removed.

This shows both CCDs in the instrument (previous slides only had the lower one).
• HST previously used an in-house workflow manager called OPUS
• OPUS featured
  • Distributed processing on a pool of machines (shared disk access)
  • A blackboard paradigm for distributing work
• OPUS was reevaluated under a trade-study in 2011 to choose the workflow management system for JWST
  • HTCondor was chosen for use with custom-made OWL (Orchestrated Workflow Layer) add-on
  • The HST Mission Office decided to allow OPUS to be phased out for HST processing in favor of HTCondor/OWL
• OPUS was gradually phased out and was fully retired in Feb. 2018.
Why HTCondor?

• Better performance and flexibility for large processing runs
• Capability to add machines when needed, then release them for normal operational loads
• A more maintainable, reliable system in the future
  • OPUS was developed in-house, but the expertise had left
  • Huge HTCondor user base (with conferences and everything)
• OPUS could not handle the large data processing needs of JWST. Converting HST to HTCondor had the advantages of
  • Operators and developers only need to know one system
  • Use HST to gain HTCondor expertise before JWST launch and operations
What and Why is OWL?

- OWL = Orchestrated Workflow Layer
- HTCondor manages compute resources extremely well
- Lacks services for managing and tracking the data being processed
- What OWL provides
  - A job-tracking database table (the blackboard) that captures every step in the workflow populated and updated by HTCondor job hooks
  - Template-driven workflow generation ("DAGs on-the-fly") using the Jinja2 template engine
DAGs-on-the-Fly

- DAG templates

```
# Job definitions.
JOB ZFITS sdp_edt2fits_{{ dataset }}.job
JOB RF sdp_bestref_{{ dataset }}.job
JOB BC sdp_before_calib_{{ dataset }}.job
JOB CA sdp_calibration_{{ dataset }}.job
JOB MD sdp_astrodizzle_{{ dataset }}.job
JOB AC sdp_after_calib_{{ dataset }}.job
JOB INGEST_SCI archive_submit_{{ dataset }}.job
JOB PVW sdp_preview_{{ dataset }}.job
JOB CL sdp_clean_{{ dataset }}.job

# Relationships.
PARENT ZFITS CHILD RF
PARENT RF CHILD BC
PARENT BC CHILD CA
PARENT CA CHILD MD
PARENT MD CHILD AC
PARENT AC CHILD INGEST_SCI
PARENT INGEST_SCI CHILD PVW
PARENT PVW CHILD CL

# Condor priorities
PRIORITY ZFITS 50
PRIORITY RF 55
PRIORITY BC 60
PRIORITY CA 65
PRIORITY MD 70
PRIORITY AC 75
PRIORITY INGEST_SCI 85
PRIORITY PVW 80
PRIORITY CL 90
```

sdp_psearch_wf3.dag
DAGs-on-the-Fly

- DAGs are created for specific datasets
  (here WFC3 datasets idkv02010 and idkv02020)
What and Why is OWL?

- OWL = Orchestrated Workflow Layer
- HTCondor manages compute resources extremely well
- Lacks services for managing and tracking the data being processed
- What OWL provides
  - A job-tracking database table (the blackboard) that captures every step in the workflow populated and updated by HTCondor job hooks
  - Template-driven workflow generation (“DAGs on-the-fly”) using the Jinja2 template engine
  - A web app (OWL GUI) for monitoring dataset processing status and other system features
<table>
<thead>
<tr>
<th>Fileset</th>
<th>Process Name</th>
<th>Status</th>
<th>Rescue</th>
<th>Start Time</th>
<th>Completion Time</th>
<th>Controls</th>
<th>State</th>
<th>Exit Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>iccz13q0q</td>
<td>INGEST</td>
<td></td>
<td></td>
<td>2018-05-11 18:33:50.0</td>
<td>1970-01-01 00:00:00.0</td>
<td></td>
<td>Starting</td>
<td></td>
</tr>
<tr>
<td>icd903010</td>
<td>REPRO_CL</td>
<td></td>
<td></td>
<td>2018-05-11 18:33:50.0</td>
<td>1970-01-01 00:00:00.0</td>
<td></td>
<td>Starting</td>
<td></td>
</tr>
<tr>
<td>iccz13ptq</td>
<td>INGEST</td>
<td></td>
<td></td>
<td>2018-05-11 18:33:12.0</td>
<td>1970-01-01 00:00:00.0</td>
<td></td>
<td>Exited</td>
<td>0</td>
</tr>
<tr>
<td>iccz13psq</td>
<td>INGEST</td>
<td></td>
<td></td>
<td>2018-05-11 18:33:01.0</td>
<td>1970-01-01 00:00:00.0</td>
<td></td>
<td>Running</td>
<td></td>
</tr>
<tr>
<td>iccz13pwq</td>
<td>INGEST</td>
<td></td>
<td></td>
<td>2018-05-11 18:32:50.0</td>
<td>1970-01-01 00:00:00.0</td>
<td></td>
<td>Running</td>
<td></td>
</tr>
<tr>
<td>iccz13pqq</td>
<td>INGEST</td>
<td></td>
<td></td>
<td>2018-05-11 18:32:22.0</td>
<td>1970-01-01 00:00:00.0</td>
<td></td>
<td>Running</td>
<td></td>
</tr>
</tbody>
</table>
## OWL GUI (Data Flow For a Single Dataset)

### Fileset | Process Name | Status | Start Time | Completion Time | Lasted Rescued | State | Exit Code | Owner
--- | --- | --- | --- | --- | --- | --- | --- | ---
idx08030 | CL | Exited | 2018-05-11 17:07:09.0 | 2018-05-11 17:07:10.33 | None | Exited | 0
idx08030 | PWV | Exited | 2018-05-11 17:05:01.0 | 2018-05-11 17:07:01.373 | None | Exited | 0
idx08030 | INGEST_SCI | Exited | 2018-05-11 17:03:01.0 | 2018-05-11 17:04:52.457 | None | Exited | 0
idx08030 | AC | Exited | 2018-05-11 17:02:48.0 | 2018-05-11 17:02:49.967 | None | Exited | 0
idx08030 | MD | Exited | 2018-05-11 17:00:06.0 | 2018-05-11 17:02:39.637 | None | Exited | 0
idx08030 | 2FITs | Exited | 2018-05-11 14:48:45.0 | 2018-05-11 14:48:57.177 | None | Exited | 0
Managing Datasets

• OWL allows us to specify and feed data processing runs into the system
  • Data Processing Queue (DPQ) database table holds workflow requests
Queue Processing

**DESCRIPTION:** Shows DPQ_DB.DpQueue table entries for workflows that have not yet been started by the Shoveler task, or for informational message workflowTypes that communicate status of processing where a workflow could not yet be started for a fileset. The Actions column provides a trash icon that, when selected, will result in deletion of the table row from the DPQ_DB.DpQueue database table. This can be used to remove informational message lines that have been looked into and are no longer needed. Once the Shoveler picks up an entry from the DPQ_DB.DpQueue table, the row on this display will disappear, and migrate to the DPQ_DB.DpQueueHist table, and be visible on the History tab.
Managing Datasets

• OWL allows us to specify and feed data processing runs into the system
  • Data Processing Queue (DPQ) database table holds workflow requests
Managing Datasets

• OWL allows us to specify and feed data processing runs into the system
  • Data Processing Queue (DPQ) database table holds workflow requests
  • The DPQ is populated by Pollers which watch target directories for new files. When new files are found, workflows are added to the DPQ table.
<table>
<thead>
<tr>
<th>Start/Stop</th>
<th>Status</th>
<th>Poller</th>
<th>Last Update</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>repro_manager</td>
<td>2017-09-26 11:04:06.6070000</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>rescue_me</td>
<td>2017-09-26 11:05:26.6630000</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>SHOVELER</td>
<td>2017-09-13 18:19:42.6470000</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>CRDS</td>
<td>2017-09-25 19:35:13.0600000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>CSUM_COS</td>
<td>2017-09-26 11:05:18.6400000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>CSUM_STI</td>
<td>2017-03-17 18:56:40.1270000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>dan</td>
<td>2017-09-26 11:05:19.0500000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>dlg</td>
<td>2000-01-01 00:00:00.0000000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>DP_AC6/owl</td>
<td>2017-09-26 11:05:58.0500000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>DP_CO8/owl</td>
<td>2017-09-26 11:05:51.9200000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>DP_STI/owl</td>
<td>2017-09-26 11:05:55.4870000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>DP_WF3/owl</td>
<td>2017-09-26 11:05:58.0870000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>jlt</td>
<td>2000-01-01 00:00:00.0000000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>prnm</td>
<td>2017-09-26 11:05:17.3830000</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>trnn</td>
<td>2000-01-01 00:00:00.0000000</td>
<td>60</td>
</tr>
</tbody>
</table>
Managing Datasets

• OWL allows us to specify and feed data processing runs into the system
  • Data Processing Queue (DPQ) database table holds workflow requests
  • The DPQ is populated by Pollers which watch target directories for new files. When new files are found, workflows are added to the DPQ table.
  • The Shoveler governs the order and rate at which DPQ entries are sent through the OWL workflow template generator to be transformed into executing DAGMan jobs on the HTCondor pool
• The rescue server receives rescueDAG requests to re-try a failed workflow
# A Failed Workflow

<table>
<thead>
<tr>
<th>Fileset</th>
<th>Process Name</th>
<th>Status</th>
<th>Start Time</th>
<th>Completion Time</th>
<th>Lasted Rescued</th>
<th>State</th>
<th>Exit Code</th>
<th>Remote Host</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN1448</td>
<td>dan_receipt</td>
<td>Exited</td>
<td>2017-09-20 04:26:41.0</td>
<td>2017-09-20 04:26:41.17</td>
<td>None</td>
<td>Exited</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Showing 1 to 1 of 1 entries
Future STScI Operations

★ The Hubble Space Telescope is the most successful scientific observatory in history
★ Operations are expected to continue into the mid 2020’s
★ The James Webb Space Telescope will be the premier observatory of the next decade
★ Expected to launch spring 2020
★ 6.5 meter segmented mirror optimized for the infrared
★ HTCondor + OWL provide a unified system for the data processing pipelines, allowing for efficient calibration and archiving operations.