NOAO SDM Document PL013

ODI Pipeline Data Flow Design

F. Valdes[1]

National Optical Astronomy Observatories
Science Data Management

V1.1: September 10, 2009
Minor edits: October 23, 2009

[1]NOAO Science Data Management, P.O. Box 26732, Tucson, AZ 85732

Table of Contents

Abstract

A distributed and data-parallel science pipeline data flow design for the WIYN One Degree Imager (ODI) is presented. This high-performance design is required to handle the large data volumes from this premier camera. The design is targeted to satisfy the pipeline science requirements set forth by Dell'Antonio, et al. [1]. The design is grounded in experience with the NOAO Mosaic Imager high-performance pipeline application [8], the MOSAIC pipeline for simplicity, which has similar processing requirements. The ODI data flow design may be implemented using the NOAO High Performance Pipeline System (NHPPS) [10] and many of the pipelines, components, and stages from the MOSAIC pipeline. This eliminates the pipeline framework effort and limits application development to ODI-specific requirements, thereby speeding delivery of a pipeline for ODI data.

Keywords: ODI, pipeline

Purpose of this Document

This document is an architectural blueprint for a high-performance science pipeline for the WIYN One Degree Imager (ODI) [3] that satisfies the pipeline science requirements for this instrument [1]. The data flow (also known as a workflow) through various pipeline services is the language of the blueprint. The internal structure of the services – the stages and computational tasks – is more of a bricks and mortar implementation detail.

The purposes and goals for this document are:

- Communicating the vision, challenges, approaches, and techniques for a high-performance ODI pipeline to various audiences including science and project boards and managers
- Presenting a technically complete plan suitable for a design review
- Providing a blueprint for developers to implement the basic pipeline units
- Identifying a fast-track implementation plan that makes use of existing software and experienced pipeline developers

1 Introduction

The WIYN One Degree Imager (ODI) is a very large format (giga-pixel) optical camera with electronic image stabilization (see Jacoby, et al. [3]). Many users will depend on a high-performance pipeline to turn their raw exposures into science ready data in a timely manner. This pipeline must make use of a cluster of processing nodes at optimal efficiency. This requires a data-parallel and distributed data flow which can be optimized for the available resources. While the pipeline framework, for example the NOAO High Performance Pipeline System (NHPPS) [10], provides the functionality to perform many data processing operations on a cluster, the pipeline designer is responsible for organizing the processing operations, algorithms, and data flow to make optimal use of this functionality within the available resources.

The ODI camera consists of an 8x8 mosaic of Orthogonal Transfer Array (OTA) detectors. Each OTA provides an 8x8 grid of independently addressable pixel cells. The cells are effectively small independent CCDs. The specifics of the cells are not critical to the pipeline application. The key point is that the data format produces 4096 image rasters with size of order 500^2 light sensitive pixels with small gaps between cells and OTAs.

The needed calibrations and data products are set forth in the *One Degree Imager Pipeline Software and Archive Science Requirements Document* [1] defined by an invited panel with the help of the WIYN Observatory director and ODI instrument team. The data flow design described in this document is based on this requirements document as well as the author's experience with similar mosaic cameras.

The data flow design is described in a narrative overview (§4), in more functional detail by a pipeline decomposition (§5), and in examples (§7). A reader may wish to read only the overview and/or the examples to get the general idea of the data flow.

A *pipeline* in the context of this data flow document is the atomic distributed unit of the data flow. It is defined more fully in section 1.1. These pipelines operate on *datasets* as defined in section 1.2.

While the data flow presented here includes considerable detail, it is still a relatively high level view of the pipeline application. The pipelines are where the actual calibration and image processing takes place and each pipeline has its own internal data flow with parallelization, though not across nodes, which are only briefly touched on. In the data flow presented in here we are mainly concerned with identifying where the various calibrations are made, that they occur in the right order, and can be performed with as much parallelism and distributed computing resources as possible.

1.1 Pipeline Applications versus Pipelines

A fundamental architectural classification for distributed parallel processing systems is whether they are *fine-grained* or *coarse-grained*. In the former very low level parallelization within the computations is used. An example of this is parallelizing and distributing program loops over a data unit such as an image. In the latter class complete programs or sequences of programs are applied to a data unit (though multi-threading in tasks may also be used). Because calibrations for large format astronomical cameras, especially mosaic cameras, can generally be done in parallel over large detector data units such as CCDs, pipelines for these instruments are primarily coarse-grained and, with lots of parallel data (called *embarrassingly parallel*), multi-threading and fine-grained parallelism is less significant. The ODI camera and data flow design is no exception and what is presented here is architecturally coarse-grained.

A pipeline system has at least three major elements. These are the (coarse-grained) execution framework, supporting components such as a calibration manager, and the instrument specific pipeline application. While the purpose of this document is to lay out the data flow for an ODI pipeline application, we note that a straightforward and quick implementation would reuse non-instrument specific components already developed and put into production by NOAO. The implementation effort would then adapt similar components of the MOSIAC pipeline application and concentrate development on the ODI specific requirements.

The key concept in this section is that the *ODI pipeline* is a pipeline application and a pipeline application consists of many *pipelines*. In this context a pipeline is a coarse-grained set of steps that accomplish a part of the data processing on a subset of the data, called a *dataset*. One may logically think of a pipeline as a method of the pipeline application and the stages in the pipeline as the statements of the method. Another analogy that applies is that a pipeline is a service in a service oriented architecture (SOA). In the NHPPS framework a pipeline executes on a single node as a service but there can be instances of the pipeline on as many nodes as desired. The data flow logic then selects the least busy pipeline/service as new datasets are to be processed.

Many distributed pipeline systems have similar concepts of a pipeline but with different nomenclature. For example, the LSST architecture calls the framework a *harness* and data parallel pipeline instances *slices*. The latter term is particularly apt when there are as many slices as pieces of atomic data, e.g. detector elements in an exposure.

The data flow through the pipelines can be represented as a hierarchy for the most part. A pipeline may call one or more child pipelines. A pipeline generally waits for the child pipelines to return results before itself completing and returning to its parent. The key point in a data-parallel pipeline application is that at any level a pipeline may decompose its dataset into pieces and call multiple instances of a particular child pipeline. The NHPPS framework allows the designer to configure the number of instances that may be called at one time and also where in a cluster the pipeline may seek a child; particularly the distinction between using the same node as the parent or any available node.

While we make a precise distinction between a pipeline application and a pipeline, the term pipeline is often used more informally, including in this document as in the *ODI pipeline*, to refer to a pipeline application because it is shorter and natural in many contexts. Also, the terms *sub-pipeline*, in analogy with subroutine, and *child pipeline* may be used to refer to a pipeline called by another pipeline.

1.2 Datasets

The elements of the data flow for a distributed and data-parallel pipeline application are *datasets*. The key generality is that a dataset can be any desired grouping of data to be operated upon by a particular pipeline. In the object oriented paradigm a dataset is an object and a pipeline is a method which operates on that type of object. The data flow description here describes how datasets are defined at various points. It is important to understand that datasets change during the processing. For example, in one pipeline a dataset may be all OTAs of a single exposure while in another pipeline it may be all data taken with a single OTA.

Another important concept is that a dataset is a set of data *identifiers* or URIs; for example filenames in an appropriate processing cluster syntax. In a typical NHPPS pipeline application a dataset object is implemented as a ”list” file which contains filenames that include a node specification. Note that it is a data flow design choice, significantly influenced by the target cluster architecture, whether data actually moves between local file systems or is available from a high-performance shared file system visible to all nodes. The use of network resource identifiers supports both common storage models and lets the pipelines generally not care about where the data is located, though the developer should consider whether it is more efficient to work on a local copy.

1.3 The MOSAIC and NEWFIRM Pipelines

There are several points to highlight concerning the NOAO MOSAIC and NEWFIRM pipeline applications ([8], [9]), collectively referred to as the NOAO pipelines. A non-technical point is recognizing the investment and experience embodied by these pipelines. There is over 20 people-years of work and experience in developing the infrastructure, computational tasks and algorithms, and applications. This does not include the work from the IRAF processing toolkit, used in do the actual computations, and from antecedent projects; in particular, the Space Telescope Science Institute’s OPUS [7] system used for their pipeline processing. An interesting synergy is that NOAO’s NHPPS framework, which evolved from OPUS, is now feeding back as a replacement for OPUS at STScI.

The NOAO pipelines have processed large volumes of mosaic data. This has proven the NHPPS framework [10], upon which these pipeline applications are layered. to be highly reliable and efficient. They also demonstrate the flexibility of the framework by implementing significantly different data flows to meet the calibrations requirements for these instruments. For example, the NEWFIRM pipeline data flow implements multiple passes of sky subtraction and masking.

For the ODI pipeline a key point is that the data flow requirements are similar in a most respects to the MOSAIC pipeline. Therefore, the experience with developing and optimizing the data flow for that pipeline can and is reflected in this ODI design. It also follows that a fast implementation path for an ODI pipeline would make use of the NHPPS and many elements of the MOSAIC pipeline.

The NOAO pipelines support a self-calibration strategy where where both standard dome calibrations and the science data are used within a basic dataset from a block of nights. The block of nights often match PI runs. The processing is then similar to the classic (ground-based) model of PI data reductions where a PI program is assigned a block of nights and the PI is responsibly for obtaining all calibrations and calibrating their data from what they take home with them. That these pipelines operate on blocks of nights and support a classic PI calibration model is an important aspect of a general community pipeline required to operate on data from heterogeneous programs assigned time on the instrument in the classic ground-based PI run mode.

However, it is important to understand that the MOSAIC pipeline goes beyond this classic PI model to the calibration library paradigm used by most observatory level pipelines. The ODI requirements document [1] mandates a calibration library capability which the NOAO pipelines also are based upon. In the NOAO pipelines any calibrations derived from the dataset being processed are entered into a calibration library, along with data quality information, and any subsequent use of these calibrations is through the calibration manager. This means even if calibrations are derived from a dataset the calibrations used for that dataset need not be those calibrations if better quality ones are available. Also if no calibration data is present or derivable from the dataset, the data can still be calibrated with data from the library obtained separately. These may be from earlier data but if reprocessing is performed this might be from chronologically later data.

The ODI requirements [1] also specify the types of calibrations which include bias/zero, dark count, flat field, fringe template, illumination pattern, and pupil pattern. These are all calibrations produced and used by the MOSAIC pipeline. Of particular note, the derivation of night-sky calibrations, such as fringe templates and illumination flat fields, is one of the most challenging tasks for a pipeline. A lot of work and experience has gone into this in the MOSAIC pipeline. Incorporating the techniques for extracting these calibrations automatically from on-sky exposures in an efficient, data-parallel data flow requires careful design.

The calibration processing data flow for an ODI pipeline, based on experience with the NOAO pipelines, is discussed further in section 2 and in the detailed data flow design which follows.

1.4 Pan-STARRS, DECam, and LSST Pipelines

The Pan-STARRS Image Processing Pipeline (PIPP) [5] and the developing Dark Energy Camera [6] and LSST pipelines [4] are other potential sources for stages in the data flow. Pipelines implemented using the NHPPS and the data flow laid out here could include processing stages using computing elements of those pipelines. The PIPP might be a useful source for unusual operations specific to the OTA technology, though nothing more complex than convolutions based on the guiding shift history is expected. In later sections discussing PSF matching (§5.13.1) we note that this operation requires some sophisticated algorithms which are currently being developed and tested by the LSST data management project. These LSST algorithms may be incorporated in the ODI pipeline at some point.

1.5 Pipeline Frameworks

There are a number of (coarse-grained) pipeline frameworks in use and under development for astronomical imaging camera pipelines. These include OPUS/NHPPS for the Hubble Space Telescope operated by the Space Telescope Science Institute, NHPPS for the NOAO mosaic cameras, the Pan-STARRS Image Processing Pipeline for the Pan-STARRS ”telescope” using a database driven scheduler, Dark Energy Camera using grid tools (e.g. Condor-G), and LSST using their own pipeline harness. Of these the first three are in production and the latter are in data challenge driven development. As a general classification, there are two types of frameworks: those developed specifically for astronomical image processing and those based on grid technology.

Something that this document attempts to demonstrate is that an *efficient* data flow requires the more sophisticated frameworks developed for astronomical pipelines than a (solely) grid-style scheduler approach. The emphasis on efficient means that while such systems can process astronomical camera data, they will take longer, given the same computing resources, than more astronomical pipeline specific frameworks. In addition, dependence on a shared computing resource such as a TeraGrid cannot be more efficient than a 100% dedicated pipeline cluster because of scheduling latency and optimized storage access. Of course, the promise of a shared grid is to provide significantly more resources so that this makes up for the advantages of a dedicated cluster of a fixed size.

As a case in point, the largest high data throughput project in optical astronomy in active development is LSST. Their data management group is developing a framework, which they call a *harness*, targeted to the needs of their pipeline. The LSST harness might be a potential ODI framework, however it is still under development. It is currently focused on the LSST-scale problem, which means a more complex framework than our experience indicates is necessary for ODI, though their vision is for a framework that would eventually support pipelines from desktops, to clusters, to grids.

As noted previously, a pipeline based on the proven and scalable NHPPS framework with the high-performance data flow design modeled on the proven MOSAIC pipeline is the lowest risk and fastest path to a high-performance ODI pipeline.

2 Calibration Processing

The ODI data flow presented here integrates calibration and science processing rather than having separate pipeline applications. The requirements document [1] may create the impression that these are separate and it is not clear how dark sky calibrations are to be created. However, what is described here is consistent with those requirements.

An integrated data flow makes use of recognizing observation types (e.g. bias, dome flat, twilight, and on-sky exposures) and pipeline control parameters. The pipeline must also recognize detector guided modes. In such an approach the higher level pipelines, which primarily orchestrate the data flow, simply by-pass inappropriate, unnecessary, or undesired lower level processing pipelines.

An integrated data flow allows multiple operational modes; in particular, it can be used as a purely calibration generating pipeline and as a purely science exposure pipeline, or model the classic PI reduction process. For example, if the pipeline application is presented with only dome calibrations then it acts as a purely calibration generation pipeline for producing master calibrations. Alternatively, a block of nights can be presented and the master calibrations derived first before processing the science exposures. Note that in both cases any derived calibrations are produced as data products for the user and archive and also made available for use by the calibration library for current and future application.

A special requirement for ODI is that datasets without detector guiding (static mode) exposures are recognized and can be used to derive night-sky calibrations (e.g. fringing and illumination). In this case the pipeline becomes both a calibration pipeline for support scientists to generate night-sky calibrations from observations made specifically for this purpose and one used for PI programs not using the guided modes.

For data which is guided the pipeline would use only library calibration data which are convolved for each exposure's shift history.

The integrated pipeline for static exposures, where night-sky calibrations are automatically generated, is essentially the one pioneered by the MOSAIC pipeline. The most challenging aspect of that pipeline was developing decision methods to automatically determine which exposures, if any, in a dataset are suitable for creating night-sky calibrations. The ODI pipeline makes use of this experience.

Addition of a data flow that creates and applies calibration data for guided mode programs is actually quite simple. The part of the data flow used to generate dark sky calibrations from static data (the SKY pipeline described later) is skipped. Instead, static library calibrations are convolved by an exposure's shift history and applied to that exposure. Because this is independent of any other exposure it can be done in purely exposure parallel processing (the EXP pipeline defined below).

3 Limiting the Parallelization

In the data flow described here one of the main forms of parallelization occurs when a parent pipeline divides its dataset into smaller datasets and requests multiple instances of a child pipeline. A common, and obvious, case is when a full field exposure is divided in OTA pieces. If all parallelization opportunities noted in the data flow are taken without limit then it is possible to end up with hundreds or thousands of datasets all trying to make progress at the same time. This leads to a loss of performance from too much multitasking where processes are swapping out and waiting for a long time to get another turn. It can also result in reaching OS limits on the number of processes.

The solution developed for the NOAO pipelines lets the data flow designer think in terms of requesting as many instances of a pipeline as makes sense for the data without worrying about the consequences of too many processes. This is accomplished by having the framework "call and return" mechanism for pipelines throttle and queue the requests.

What happens is a parent pipeline submits all of the parallel datasets, say for 64 OTAs, to as many nodes as have a desired child pipeline available. The framework then submits a smaller number, defined in terms of the number per available node. For example if the number is two and there are 8 nodes then 16 datasets, two per node, are submitted and the rest are kept in a queue. The advantage of the per node parameterization is that if different cluster sizes occur (because of machines being down, added, or run at a different site) the loading behavior remains similar. The number can be tuned but it is generally set to match the number of cores on the nodes.

As a dataset is completed by a child pipeline and the parent pipeline is notified, the framework sends another dataset in the queue to the same node. All the parent pipeline is doing is waiting for the indication that all the datasets have completed.

An advantage of sending datasets to nodes as they complete is that it optimizes getting the whole group of datasets done when the nodes complete the work at different speeds due to hardware or load differences. Also, if a node goes down it is possible to re-queue a dataset.

A related aspect of the NHPPS to point out is that there is a configurable limit on the number of pipeline managers which run on a node. This may lead people to think that this also limits the number of datasets which run in parallel. But a pipeline manager limits the number of identical stages, the basic pipeline processing unit, that can run at the same time (on the node).

There can be any number of datasets ready for a stage within a pipeline but only the available number of pipeline manager instances can be actively processing one of the datasets at that stage. However, other datasets in other stages of the pipeline can be active as well. So if there are four pipeline manager instances for a pipeline with 10 stages then, in principle, 40 processes can be running in parallel on the node. That the number of data managers for a particular pipeline is not a limit on the number of datasets being processed is why the call and return mechanism is also required. The number of pipeline managers per node is also generally set to match the number of cores. This is because most pipelines have a rate limiting stage and so that number of processes for that stage using all the available cores provides an efficient use of the node.

In summary the data flow designer and operator can control the number of parallel processes in a number of ways. One, of course, is how the designer chooses to decompose the problem into pieces and how to call pipelines. The ”submit all and wait” strategy is not the only method. Beyond the data flow design itself, the distribution of pipelines on the nodes can be configured. For example, one can have one set of nodes do the de-trending, another the stacking, and so forth. This gives a lot of operational flexibility. Then the number of pipeline managers on a node can be configured. And finally, the queued ”call and return” mechanism can define how many datasets are actually submitted at one time to a node.

4 ODI Data Flow Overview

In this section the ODI data flow design is presented as a narrative overview. The data flow begins with a dataset for some group of nights. The dataset may be limited to calibration data only, science exposures only, or both. In the first case the pipeline is used to generate calibrations for a calibration library. In the second case the science exposures are processed assuming all calibrations have been entered in the calibration library. In the third case the pipeline generates calibrations from the dataset, if possible, adds them to the calibration library, and then proceeds as in the science exposure only case. Note that in this last case the pipeline will calibrate the data whether or not the particular dataset, say from a PI run, includes calibration exposures. The logic is that the decision about the best calibration to use, whether produced from calibration data in the dataset or not, lies with the calibration manager (a component not described here).

The top level entry pipeline stages, verifies, remediates, and groups the data. It processes groups in the usual order for CCD calibrations – biases, darks, dome flats, and on-sky exposures. The on-sky exposures are ordered by static mode exposures which may be used to produce night-sky calibrations followed by guided exposures (coherent and local modes). Each group must complete before the next group begins, but within each group multiple calibration sequences or filter groupings are processed in parallel.

The dome calibration pipeline and its children make master calibrations from sequences of biases, darks, and dome flats and also check their data quality. The master calibrations are checked into the calibration library. The exposures in each sequence are processed together since they must be combined to produce a master calibration but are parallelized by OTA.

The science exposures are grouped by filter and multiple filters can be processed in parallel. Each filter dataset is broken down into individual exposures which are processed in parallel by a single exposure pipeline. Within each exposure pipeline the OTAs are processed in parallel by an OTA pipeline where most of the work, such as applying subtractive and flat field calibrations and determining the astrometric calibration, is performed. One subtlety, however, is that some global steps, meaning across multiple or all OTAs, are required for the astrometric solutions and for the scaling of pattern templates (pupil ghost and fringing) to match a particular exposure. This is done with a back and forth interaction between the exposure level pipeline and the OTA child pipelines. The astrometric calibration is a combination of cataloging sources in parallel at the OTA level, matching all the catalogs to a reference catalog and fitting an astrometric model at the global level, and updating the individual OTA exposures back at the OTA level.

For coherent and locally guided data the exposure pipeline applies all calibrations. In particular, calibrations made from dark sky observations using either specific on-sky calibration programs or general science programs are taken from the calibration library, convolved to the particular shift history, and applied to each exposure.

For static data the exposure pipeline does not apply dark sky calibrations. Instead another pipeline is called using the entire data set processed through dome flat fielding. This pipeline and its children determine whether to create dark sky exposures by stacking suitable exposures using source masking and rejection and extracting calibrations such as fringe templates and illumination flat fields. When these calibrations are created they are submitted to the calibration manager. Whether or not the pipeline is able to create calibrations is based on a careful data quality assessment. The individual exposures are then calibrated by night sky calibrators as is done in the single exposure pipeline for guided data, though no convolution is required.

The dark sky calibration pipeline is parallelized in a manner similar to the dome calibration pipeline. The multiple exposures are kept together but organized in parallel OTA pipelines. As with the single exposure pipeline there is some back and forth between the full field and parallel OTA pipelines to compute global parameters such as template scalings.

The final calibration of the OTA groups is ultimately parallelized to individual OTA using either the night sky calibration just created or a library calibrator.

In the pipeline where the dark sky calibrations have been applied to the OTA images – the single exposure pipeline for guided data and the night sky calibration pipeline for static data – the OTA images are resampled to a standard sky orientation and scale at a standard tangent point. This is done for data which can be stacked or when a standard sampling data product is produced.

Within the single filter dataset pipeline a stacking pipeline is applied after the instrumental calibrations. This only applies to data identified as stackable either by heuristics or from metadata supplied by the observing control system about dithered sequences. Generally there may be stacks for multiple fields. These are distributed in a parallel fashion so stacks of different fields can be created in parallel.

The stacking pipeline has two functions. One is to identify ”transient” sources, such as cosmic rays, and the other is to produce a deeper, gap removed, standard orientation data product. The transients are identified in a two pass manner. In the first pass an initial stack is created, resampled to the original detector sampling for each OTA, sources are identified in the difference of the OTA and stack, and the sources are added to a mask. In the second pass the OTA exposures are resampled again, with the transient sources excluded or removed by interpolation to minimize ringing from sharp cosmic rays, and then stacked a second time using a mask which includes any newly detected transients.

The stacking pipeline and its children are organized such that steps are parallelized at the appropriate points. In particular, matching a piece of the first pass stack to an OTA, doing the difference detection, and resampling the OTA with the transients removed is done in parallel for individual OTAs.

The requirements document [1] identifies two levels of processing called *Tier 1* and *Tier 2*. Stacking and use of difference methods is categorized as Tier 2. However, as just described, stacking is a fundamental method for cosmic ray removal, which is a Tier 1 calibration requirement. Because of this the stacking is done in a separate pipeline, as just described, from what is called the science pipeline.

The science pipeline takes all the calibrated data products, again within the parent single filter pipeline, and adds additional science capabilities. This includes analyzing the catalogs from the difference detections in the stacking pipeline for interesting transient sources, making source catalogs, and doing image filtering operations.

The last part of the data flow for a dataset consists of a couple of generic pipelines. These are found in both the calibration and science exposure pipelines. One is a data products pipeline that builds the final data products, for instance making multi-extension files or combining resampled OTAs into larger pieces such as a full image or, more usefully, ”slices” of a convenient size that tile easily. One type of data product, which might be only for operators, is convenient review documentation. An example of this is a ”contact” sheet of graphics (e.g. png or jpeg as in figure 3) which can be reviewed quickly in a browser.

The data product pipeline has child pipelines to handle multiple exposures in parallel. One capability these parallel pipelines provides is archive submission of the data products. This is an option because one operational model to support is having an operator or pipeline scientist review the data products prior to archiving. Rather than block the data flow, this review process is moved outside the pipeline using data products saved by the data transport pipeline described next.

The other utility pipeline runs at the end of a calibration or filter dataset. Its function is to move all data to be saved from the pipelines, where the data may be distributed throughout the processing cluster, to a desired location and format. This might consist of creating an ftp directory of data products including results which are not necessarily going into a long term archive. As well as collecting data products to be saved, this pipeline also cleans up all the remaining intermediate processing data so that the next dataset from the top level pipeline starts with sufficient disk space.

Upon completion of the last guided exposure dataset the top level pipeline, which has been waiting for all child pipelines to return, does some final clean up, logging, and so forth. It notifies the agent which submitted the dataset and is then ready to accept a dataset from another block of nights from that agent. In the NOAO pipelines that agent is the *Pipeline Scheduling Agent* [11].

5 ODI Data Flow Details and Pipeline Decomposition

In the following sections a description of each pipeline shown in figure 1 is presented. The figure illustrates a pipeline hierarchy where data roughly flows from the top down. The tabbing indicates a parent-child relationship where a pipeline calls one or more instances of another pipeline and does not complete until all children have completed.

The description of each pipeline includes the type of dataset it handles, some idea of the functions it performs, and how it interacts with parent and child pipelines. Discussion of the computational operations and stages is limited, though issues that are unique to ODI or are important from a parallel processing standpoint are noted.

5.1 TOP Pipeline

The TOP pipeline is the primary entry point for the ODI pipeline application. In principle, there can always be higher level pipelines defined in terms of collections of datasets expected by a lower level pipeline; for example, a semester pipeline could call a month pipeline, which calls a run pipeline. It is also possible to enter a lower level pipeline with an appropriate dataset; this is done when a subset of the pipeline application is configured as a ”quick reduce” pipeline at the telescope. At NOAO, datasets are submitted to the top level pipelines by a *Pipeline Scheduling Agent* driven by a *Pipeline Scheduling Queue* [11]. However, the way a dataset is generated and submitted to the TOP pipeline is an operational detail independent of the pipeline data flow design.

For the ODI pipeline application we define a TOP pipeline dataset as exposures from a block of contiguous nights. Often this is equivalent to a PI ”run”, also known as a ”campaign”. But queue or shared scheduling can also result in a dataset with a mix of PI data. There is no problem with this for the pipeline. It is up to the archive or the data distribution system to protect proprietary data. As described in the *NOAO Science Pipelines Operations Model* [11] the rationale for blocks of nights is to handle cases where there are only a few exposures for a particular filter due to weather, field availability during a night, or observing strategy.

While the default is for a block of nights, the details of the grouping are irrelevant for the TOP pipeline and a dataset may be submitted for a single night or some other grouping of data. This is because one function of the TOP pipeline is to decompose and orchestrate the dataset into smaller pieces for processing. Section 2 pointed out that a dataset might consist of only the calibration data, including targeted static exposures for dark sky calibrations, in which case the pipeline application acts as a pipeline for creating library calibrations.

The functions for the TOP pipeline are

- orchestrate the staging of the data
- perform basic data verification and remediation
- create independent datasets based on filter, observation type, night

Figure 1: The ODI Pipeline Hierarchy.

```
TOP           - orchestrate block of nights
  CAL         - orchestrate bias, dark, and dome flat sequences
    SEQ       - process a single calibration sequence and add to library
      OCL     - process and combine the OTAs from a calibration sequence
    DPS       - prepare data products, produce operator summaries
      EDP     - create single exposure data products, archive
    DTS       - transport any desired products to short term repository
  FTR         - orchestrate on-sky data from one filter
    EXP       - exposure global calibrations, split data into OTAs
      OTA     - OTA calibrations
    SKY       - create night-sky calibrations by OTA (static data only)
      PGR     - derive pupil ghost calibration from group of OTAs
        SPG   - remove pupil ghost for single OTA
      FRG     - derive fringe frame from group
        SFR   - remove fringing for single OTA
      SFT     - derive night-sky flat from group of OTAs
        SSF   - remove pupil ghost for single OTA
      RSP     - resample astrometrically calibrated OTAs
    STK       - stack overlapping exposures, mask transients
      OSK     - compare single OTA to stack to detect transients
    SCI       - analysis and special processing (aka Tier 2)
    DPS       - prepare data products, produce operator summaries
      EDP     - create single exposure data products, archive
    DTS       - transport any desired products to short term repository
```

- distribute and sequence the datasets
- clean up
- notify the triggering agent of the final status

One might expect that persisting of the pipeline data products, meaning archiving or staging to an ftp directory, might be a function of this pipeline. But instead this function is delegated to a lower level; specifically pipelines dealing with a single filter, night, etc. This allows more efficient reprocessing in the event of a system crash where only the incomplete parts of the original dataset need be processed.

The key orchestration step of distributing and sequencing datasets is the first level of parallelization. Multiple filters, each representing one or more nights depending on the number of exposures per night, are typically submitted to parallel instances of the next pipeline. The NHPPS infrastructure, with advice from the pipeline, controls how many datasets should be attempted in parallel. Experience has shown that two is a good choice to ensure good resource use.

This pipeline also identifies all exposures which may be stacked. This is done with a WCS heuristic or possibly based on information about the observations included with the data at the telescope. All exposures are assigned a tangent point for resampling. The tangent points are defined by a grid on the sky but with a constraint that overlapping exposures will use the same tangent point to avoid boundary problems.

5.1.1 Staging the Data

As defined in §1.2, a dataset is represented as a set of resource identifiers and not the actual files. For the TOP pipeline this may be a higher level abstraction of archive identifiers returned from an archive query. These identifiers are passed to an archive/pipeline data transport interface service to physically stage the data. The optimal staging strategy is strongly dependent on the pipeline and archive system architectures and the way the data is organized and served by the archive.

In the MOSAIC and NEWFIRM pipelines the staging consists of copying the multiextension FITS files (MEF) for each exposure to a data manager staging area. Other pipelines then decompose the MEF files and distribute the pieces to local storage on the nodes assigned for those pieces. For DECam the proposed approach is to provide read-only links to the actual archive store in a network file system so no data is moved at this point. For ODI it may be the case that a single exposure is stored as multiple files. It may also be more efficient to stage data in a distributed fashion at this point. For example in a cluster of four nodes data might be staged in parallel by quadrants with the archive providing multiple servers.

Since the target processing cluster is still to be defined, a detailed staging strategy and data flow is not defined further in this document.

5.1.2 Sequencing the Dataset

The exposures are sequenced in the following groupings.

- biases

- darks
- dome flats and twilight flats
- static sky exposures
- guided sky exposures

Note that twilight flat sequences are treated in the almost same way as dome flat sequences. In the calibration discussion below the term dome calibrations, in contrast to the term night sky calibrations, should be interpreted as including the on-sky twilight exposures.

The calibration grouping is processed first using the CAL pipeline. The static sky exposures are processed next by the FTR pipeline with access to any calibrations just created. Finally the guided sky exposures are also processed by the FTR pipeline with potential use of dark sky calibrations derived from the static sky exposures.

5.2 CAL Pipeline

The CAL pipeline processes bias, dark, dome flat, and twilight flat sequences to produce master calibrations. The input dataset is a collection of all dome calibration exposures from the TOP pipeline dataset. It processes calibrations types in order of bias, dark, and dome/twilight flat. Within a calibration type it processes sequences in parallel.

While this pipeline is designed to handle a dataset of all calibrations, the TOP pipeline should do more orchestration to call the CAL pipeline for all bias sequences only and then all dome and twilight flat field sequences for each filter, with multiple filters using multiple parallel instances of the CAL pipeline. This orchestration is similar to how the FTR pipeline is called and is useful in allowing the data products to be completed in smaller groupings which makes crash recovery/restarts more efficient.

5.3 SEQ Pipeline

The SEQ pipeline processes a single dome calibration sequence. It breaks up the sequence of exposures into sequences of OTAs to be processed by parallel instances of the OCL pipeline. When all the OTAs have been processed, the master calibration is added to the calibration library if the data quality characterizations from the OTA pipelines is satisfactory. Note that the calibrations are maintained as separate OTA images. This is useful since when the calibrations are later requested it comes from an OTA level pipeline that only needs a calibration for its particular OTA. Master calibrations found to be bad or of poor data quality are identified for later operator review.

5.4 OCL Pipeline

The OCL pipeline processes a dome calibration sequence from a single OTA. The is the primary processing pipeline for dome calibrations. It performs crosstalk, overscan, and bias/dark calibrations as well as dealing with bad pixels. After calibrating individual exposures it creates a master calibration from the sequence. This pipeline also evaluates the data quality of individual exposures and the sequence. One important check is for saturation in dome and twilight flat field exposures. When bad calibrations are identified they are flagged and excluded from stacks and, when too many exposures in a sequence are bad, the OTA calibration is marked as bad.

There is an additional step for some dome and twilight flat field data. If the filter suffers from a pupil ghost pattern it is removed from the master calibration so that the pattern is not flat fielded away from the science exposures, which is photometrically incorrect. Light from the pupil pattern in the science exposure must be removed by subtracted during science exposure calibration.

5.5 FTR Pipeline

The FTR pipeline processes a collection of on-sky exposures from the same filter. The dataset may contain exposures from one or more nights. Typically, the datasets have been organized to have at least a minimum number of exposures such that if that number was not obtained in one night then additional nights are included.

Some of the algorithmic steps performed in this pipeline after basic dome calibration, is evaluation of the suitability for creating dark sky calibrations from static mode exposures. These involve analysis of low resolution sky images over the whole field, consideration of various catalog criteria such as source density and source size, and the distribution of the number of non-source pixels over the detectors. The input to these algorithms makes use of cataloging information – low resolution sky maps, evaluated source sizes and density, and source masks – generated by child pipelines.

5.6 EXP Pipeline

The EXP pipeline processes a single exposure. There are typically many EXP pipelines running in parallel. This pipeline has two main functions. One is breaking down an exposure into OTA pieces for parallel OTA pipelines and the other doing astrometric and photometric calibrations across multiple OTAs.

5.6.1 Astrometric Calibration

Astrometric calibration consists of identifying stars in the exposure with those from an astrometric reference catalog. The identified stars are used to fit a transformation between pixel and celestial coordinates. The transformation is a function with sufficient degrees of freedom to account for distortions (both within the telescope/camera and from atmospheric refraction) but of low enough order to enforce the natural physical continuity of the sky, camera, and detectors.

A question that arises with wide-field mosaic cameras is how much one can parallelize this calibration. The conservative approach is to utilize the entire field to avoid discontinuities between detectors and to make use of physical constraints. This conservative, full field approach also imposes the minimum requirement on reference source density.

The degree to which constraints on the astrometric solution are applied – from full global continuity, to groups of OTAs, to independent solutions for each OTA – is selected by a pipeline configuration parameter. What this means is that an implementation must support some level of interaction between OTAs. When processing is parallelized by OTA this requires some computation in another pipeline whose dataset is a collection of OTAs. The simplest architecture is to do this with the EXP pipeline. The ODI pipeline science requirements [1] place some requirements on the astrometric solution and recognizes that solutions computed purely from reference sources in an OTA are problematic.

Astrometric solution constraints over multiple OTAs in a data flow with parallelization at the OTA level can still make use of the parallelization by appropriate design. The approach is to generate catalogs for each OTA independently after which the catalogs are collected by the EXP pipeline for analysis. When all catalogs are received, the catalogs are merged into larger catalogs at the desired OTA grouping scale. For a fully global coupling a single merged catalog is created. During an earlier stage the EXP pipeline requests a catalog of reference astrometric sources from a catalog service. This can run in parallel with other steps to absorb any latency in querying the reference catalog. The requested field is defined by the initial WCS defined at the telescope. Uncertainties in the initial WCS are accounted for by increasing the reference field size appropriately.

Once the merged exposure catalogs and the reference catalog are matched the resulting catalogs are used to define astrometric solutions. The solutions will be adaptive such that when the source density is high fairly local solutions are determined while for lower densities more global solutions are obtained. Ultimately the solutions are cast into functions for each OTA tied to a common tangent point corresponding to the optical axis. The OTA-specific calibrations are send back to the OTA pipelines to update the image WCS. An adaptive approach with global constraints is in use with the MOSAIC pipeline.

5.6.2 Photometric Characterization

A by-product of the astrometric calibration uses the observed sources matched to the catalog sources to produce some photometric characterizations such as depth. This involves finding the flux scale, expressed as a magnitude zeropoint, that best matches the reference source magnitudes to the instrumental fluxes without considerations of bandpass and source colors. This is not the same as a photometric calibration but the photometric quantities are very useful for data quality characterization. The zeropoint, along with sky levels derived from the cataloging, are also key quantities needed for stacking exposures later in the data flow.

Global and single OTA quantities are calculated. This information is also set back to the OTA pipelines for incorporation into the header metadata for the OTA images.

5.7 OTA Pipeline

The OTA pipeline processes a single OTA. This is where most of the calibration is done. Typically there are as many parallel instances of this pipeline running on as many OTAs from as many exposures as maximizes the usage of the cluster resources.

Initially the OTA pipeline processes data from an OTA exposure as a collection of single cells. After crosstalk and overscan calibrations the cells are merged into a single raster with a mask describing the gaps between the cells as well as other bad or unusable pixels. Note that for on-

Figure 2: Operations performed by the OTA pipeline.

- crosstalk
- overscan
- masking and replacement
- merging
- bias or dark subtraction
- flat fielding
- single exposure cosmic ray masking
- pattern subtraction (e.g. fringing)
- catalog generation for astrometric calibration and photometric characterization
- updating the header with astrometric and photometric calibrations from the EXP pipeline
- resampling for stacking

detector guided exposures the shift history widens the gaps as some pixels are shifted out of the active detector area. All further processing is done on these merged OTA images. The OTA image is not unlike a single 4Kx4K CCD with masked columns and rows and is the basic unit for most further processing.

A question that arises is why stop the parallelization at an OTA? In principle one could distribute data down to the level of a cell. This is partly a question of resources. However, the main consideration is the context and goals of the pipeline. For a science pipeline run on a block of exposures there is more than enough parallelization by working on a large number of exposures in parallel. This is the context assumed in this design. The efficiency of parallelization by exposure has been demonstrated with the MOSAIC and NEWFIRM pipelines.

For the case of a *quick reduce* pipeline running at the telescope in near real-time with significant resources one could optimize the throughput for a single exposure by parallelizing to a cell level. But even then disk and network transfers would likely limit the throughput.

The required list of calibrations is shown in figure 2. This data flow document is not intended to describe the nature of the various calibrations or the techniques. Rather the following discussion focuses on the impact to the data flow.

5.7.1 Crosstalk

Crosstalk between cells along rows in an OTA is a possibility. The presence and characteristics of crosstalk have to be evaluated with engineering data. The calibration either consists of scaling and subtraction if there is a linear coupling or added to a mask if no adequate correction is possible. The QUOTA experiment [2], which used different controller electronics, showed very little crosstalk. Crosstalk would likely be the first calibration performed since it depends on the raw signal levels. It needs to be done between cells as separate images so this operation precedes merging.

If there is crosstalk between OTAs, the data flow described here would need to be modified. The EXP pipeline would call parallel instances of a crosstalk pipeline as done with the MOSAIC pipeline where crosstalk between pairs of CCD occurs. The crosstalk child pipelines take datasets of OTAs which affect each other. The results of these pipelines are sent to as many OTA pipelines as needed. What is slightly unusual in this case is that the OTA pipeline returns to the EXP pipeline and not the crosstalk pipeline. To emphasize, this alternate data flow is well-understood with a working example but will only be implemented if needed.

5.7.2 Overscan

Overscan calibration uses overscan data associated with each cell. Hence this operation must be performed with data that is stored by cell in an MEF file. Overscan correction also includes a ”trim” operation where the overscan data is removed after use. Another comment is that different overscan subtraction algorithms can be applied depending on analysis of the overscan, with the characterization becoming part of the data quality metadata. Examples of this analysis are 1) the overscan is zero indicating a total dropout of signal and so the entire cell is flagged as bad and 2) the overscan shows abrupt jumps in level which requires a line-by-line overscan evaluation, and 3) the overscan varies smoothly and can be fit by a low order function.

5.7.3 Masking and Replacement

This step identifies non-photometric pixels which are defined by an input calibration library mask and various algorithms. The identified pixels are added to an output mask for the exposure. Examples of these types of pixels are bad detector pixels (from the input mask), saturation, and blooming/bleeding. The non-photometric pixels are possibly replace by interpolation. One ODI consideration is that guided exposures will smear out bad detector pixels.

5.7.4 Merging

It is more convenient for the data flow to handle data from a single OTA as a simple image rather than a collection of cells. The cells are merged into an image based on the physical layout of the OTA. The non-light sensitive areas between cells are tracked with a mask.

One complication is when the exposure is ”locally” guided with multiple regions within a single OTA. Even then it is possible to develop the processing stages based on the merged single image. Whether this is done is an implementation question with trade-offs between the file format simplifications and the complexity of tracking regions.

5.7.5 Bias or Dark Subtraction

There are minimal special considerations here. Calibration data is obtained from the calibration manager. It is likely only bias, also known as zero, calibration will be needed but dark calibration will also be supported. Dark calibrations will need to be convolved by the shift history of an exposure.

5.7.6 Flat Fielding

Flat fielding has some special ODI considerations. If the dataset consists of guided mode exposures this step involves convolving flat field calibration data obtain in static mode with the guiding shift history. Guided mode data is also corrected for night sky illumination at this point. The calibration data is obtained from the calibration manager.

For static mode exposures this step consists of only applying a dome flat field calibration. Night sky illumination calibrations are deferred to the SFT pipeline after possibly deriving the dark sky calibrations from the dataset.

5.7.7 Cosmic Ray Masking

Cosmic ray masking consists of applying a single image cosmic ray detection algorithm. This is done conservatively to not compromise the science data and because there may be a later, more sensitive, detection for data taken with multiple exposures for dithering, stacking, or simple cosmic ray splits. The detected cosmic rays are added to a mask and possibly replaced by interpolation. The latter is a pipeline option whose default behavior will be defined by a science advisor. There are no unusual data flow issues.

5.7.8 Pupil Pattern Subtraction

A pupil pattern is where reflections within the optics produce a an image of the telescope pupil on top of the in-focus image of the field. The important characteristic of this pattern is that it is dependent only on the total light, which may include light outside the imaged field, and independent of the distribution of the light. To correct for this additive reflected light requires scaling a template of the pupil pattern, produced by the calibration pipeline and stored in the calibration library, to match the pattern in a particular exposure and subtracting.

An important data flow aspect of this is that ideally the same scaling of the template applies to all parts of the pattern which fall on different OTAs. A switch in the pipeline selects whether to use independent scaling, where a scaling is derived only from the part of the pattern in the OTA, or a ”global” scale factor. A global determination is more robust since only a small part of the pattern may fall on an OTA and the pattern may be especially confused with sources in some places.

A global scale is determined by accumulating the fitting statistics independently for each affected OTA, returning this information to the EXP pipeline which computes the net statistics and scale factor, and then returning the value to the child OTA pipelines. This ”handshaking” logic has been pioneered in the MOSAIC pipeline. It does impose a ”synchronization” point in the data flow but there is no avoiding this if a global fitting of the pattern is required.

Potentially the template derived statical exposures must be convolved by the shift history of the target guided exposure, but the pupil pattern is of such low resolution that it would be a waste of time. This is to be determined by a science advisor.

5.7.9 Catalog Generation and Astrometric Calibration

The astrometric calibration and photometric characterization also involves an interaction with the parent EXP pipeline. Catalogs generated in parallel for each OTA are collected by the EXP pipeline, merged into a master catalog, sources are matched to a reference catalog, and WCS corrections derived. The corrections and global data quality information is then sent back to the appropriate OTA pipelines for header updates. An efficiency opportunity is for the global astrometric calibration in the EXP pipeline to be done in parallel while other pixel level calibration operations are being done in the OTA pipeline.

5.7.10 Resampling

Resampling the OTA data to a particular sky pixelization is required for stacking dithered or serendipitously overlapping exposures. This can be done in parallel provided all OTA pipelines share a tangent point, output pixel scale, and orientation. The desired pixelization is supplied by pipeline parameters describing a standard orientation and scale. Since there is no reason to chose any particular orientation, this is typically the standard ”north up and east left” at a uniform and rounded pixel scale such as 0.1 arcsecond/pixel.

The tangent points are determined as one of the orchestration steps of the TOP pipeline. It identifies all the data that will or can be stacked and defines tangent points. A typical choice is for the tangent points is based on a grid (say a Healpix grid at some resolution) so that archival users might have a high probability that data can be stacked from different programs without requiring an additional interpolation. However, the algorithm must require that data being processed as a dataset not use more than one tangent point for overlapping exposures.

One question to be addressed by a science advisor is whether the individual resampled exposures are to be archived data products. If this is not the case, the resampling would only be done for exposures identified by the TOP pipeline as stackable. For the MOSAIC and NEWFIRM pipelines it is currently the case that a resampled version of each exposure is archived. For NEWFIRM this is important because virtually all observations are fairly long dithered sequences and observers may want to change the way the pipeline selects and weights exposures in a stack without having to do the resampling themselves.

An implementation note is that if the data being resampled will be resampled again in the transient detection algorithm, a faster, not as scientifically desirable, resampling algorithm can be used.

5.8 SKY Pipeline

The purpose of the SKY pipeline is to evaluate whether dark sky calibrations can be derived from a dataset, if so, it derives the calibrations, and applies dark sky calibrations to the exposures. In addition, exposures are resampled to a standard orientation in the same way as done for the guided exposures in the OTA pipeline.

The SKY pipeline is run only on static mode datasets. It is an high level pipeline which orchestrates various child pipelines. The input dataset is the set of all exposures in the FTR dataset from a single OTA. So the SKY pipeline runs as many parallel instances. The input OTA data will have been bias, dome flat, and astrometrically calibrated by the EXP pipeline but not night sky calibrated or resampled.

Note that like the SEQ pipeline, the pipelines which create pupil pattern templates, fringe frames, and dark sky illumination flat fields enter the calibrations in the calibration library by OTA. This is for the same reason that when a calibration is requested for application to a science exposure it occurs in a parallel pipeline calibrating a single OTA.

5.9 PGR and SPG Pipelines

These pipelines derive a pupil ghost template from the set of OTA exposures and apply it, or a library template, to the individual images. It has logic to determine whether it is possible to derive the template from the dataset. If a template is derived it is submitted to the calibration manager.

Applying a template to remove the pattern then includes a query to the calibration manager for the template to use. This ordering allows use of either the template just derived from the dataset or another template. Another template would be selected by the manager if either one was not entered from the current dataset or another one is considered better based on data quality considerations. Note that if no calibration is available the pipeline normally returns and processing continues without this calibration.

The pupil ghost correction involves scaling a template to each individual exposure because the amplitude is a function of the sky brightness. The scale factor is determined from the statistics of unmasked (to eliminate sources) pixels. Because only part of the pattern may be in a particular OTA there is an option to use a global scaling. To derive a global scale factor involves collecting least squares statistics from the OTAs, sending them to the FTR pipeline which combines the statistics from all OTAs into a global factor, and receiving the factors. There is additional discussion about the pupil pattern and remove in section 5.7.8.

Once the template has been derived from the dataset and the scale factors for each OTA in the set determined, the actual subtraction of the scaled template from the individual images can be parallelized by calling multiple instances of the SPG pipeline. The SPG pipeline is quite simple but is needed to provide the parallelization.

5.10 FRG and SFR Pipelines

These pipelines derive a fringe template from the set of OTA exposures and apply it or a library template to the individual images. It has logic to determine whether it is possible to derive the template from the dataset. If a template is derived it is submitted to the calibration manager.

Applying a template to remove the fringe pattern then includes a query to the calibration manager for the template to use. This ordering allows use of either the template derived from the dataset or another template. Another template would be selected by the manager if either one was not entered from the current dataset or another one is considered better based on data quality considerations. Note that if no calibration is available the pipeline normally returns and processing continues without this calibration.

The fringe correction involves scaling a fringe template to each individual exposure because the amplitude is a function of the variable night sky lines. The scale factor is determined from the statistics of unmasked (to eliminate sources) pixels. There is an option to use a global scaling. To derive a global scale factor involves collecting least squares statistics from the OTAs, sending them to the FTR pipeline which combines the statistics from all OTAs into a global factor, and receiving the factors.

Once the template has been derived from the dataset and the scale factors for each OTA in the set determined, the actual subtraction of the fringe frame from the individual images can be parallelized by calling multiple instances of the SFR pipeline. The SFR pipeline is quite simple but is needed to provide the parallelization.

5.11 SFT and SSF Pipelines

These pipelines derive a dark sky flat field or illumination correction from the set of OTA exposures and apply it or a library calibration to the individual images. It has logic to determine whether it is possible to derive the flat field from the dataset. If a flat field is derived it is submitted to the calibration manager.

Applying a flat field to flatten the data to the dark sky then includes a query to the calibration manager. This ordering allows use of either the flat field derived from the dataset or another flat field. Another template would be selected by the manager if either one was not entered from the current dataset or another one is considered better based on data quality considerations. Note that if no calibration is available the pipeline returns and processing continues without this calibration.

The derivation of a dark sky flat includes logic to decide whether the stacked candidate flat field should be smoothed to produce a large scale illumination flat or whether it can be used at all scales down to the individual pixels.

Once the flat field calibration has been derived from the dataset the actual flat fielding of the individual images can be parallelized by calling multiple instances of the SSF pipeline. The SSF pipeline is quite simple but is needed to provide the parallelization.

5.12 RSP Pipeline

The RSP pipeline resamples static mode OTAs to facilitate stacking or produce a simple ”flat” image data product in a standard orientation. This pipeline is called from the SKY pipeline for each astrometrically calibrated OTA to parallelize this operation. Whether exposures that don’t stack with other exposures are resampled depends on a control parameter. The parameter may be set based on whether single, resampled data products are desired by the PI or archive. The stages are essentially the same as those applied to guided exposures in the EXP pipeline (see [**?**]).

An implementation note is that if the data being resampled will be resampled again in the transient detection algorithm, a faster, not as scientifically desirable, resampling algorithm can be used.

5.13 STK Pipeline

The STK pipeline creates stacked data products and applies information from overlapping exposures to identify and mask transient data, most notably cosmic rays and satellite trails. The dataset is a collection of all OTA's from overlapping exposures in a single field. Multiple fields are handled in parallel. The data flow in this pipeline consists of:

- matching PSFs
- making a "harsh" stack
- distributing work to parallel OTA pipelines to detect and remove transients
- making a final science stack

An implementation note is that the stacks need not be full field, full resolution, single images. The stacks can be made in appropriate tiles. A data product implementation choice is also what to do with pixels in the final stack that have no good data. This will occur primarily for saturated stars. In the MOSAIC pipeline these pixels are cosmetically replaced by interpolation.

5.13.1 PSF Matching

PSF matching of individual exposures when creating deep stacks is a requirement for the ultimate ODI pipeline. PSF matching is a complex task which is a hot topic of current research. Projects with larger budgets and teams, as for example LSST, are struggling with finding robust solutions.

In the phased implementation advocated by the pipeline science requirements panel [1], the first ODI pipeline will not include PSF matching and, with another year or two of work by other groups, it may be possible to leverage off that work for the final ODI pipeline. In the meantime, it is important to realize that even without PSF matching significant benefits and results can be obtained.

5.13.2 Harsh and Final Science Stacks

A harsh stack is one which aggressively eliminates potential transient pixels with the consequence that some good pixels are also eliminated. The rejection of pixels in a harsh stack is done both with sigma clipping and with a median of the remaining data.

The final science stack, after transient sources have been detected and the original pixel data resampled in such a way as to minimize interpolation ringing, is created by averaging pixels using only masks to reject bad data. The masks contain bad and transient sources identified by non-statistical outlier methods. Outlier rejection is not used for the final stack becase it inevitably introduces systematic photometric errors if the PSFs and positions are not well matched.

5.14 OSK Pipeline

The OSK pipeline matches the harsh stack to a single calibrated OTA in the original (not resampled) CCD data. Pixels are identified which have a significant positive flux relative to the harsh stack. The pixels are added to a mask. The OTA is then resampled again with the pixels masked and replaced (to minimize interpolator ringing).

An implementation question is whether the detected transient sources are cosmetically removed from one or both of the unresampled and resampled versions? In the MOSAIC pipeline the resampled version is ”cleaned” while the original CCD sampling data is only masked.

5.14.1 Matching the Harsh Stack to an OTA

In order to achieve maximum sensitivity to faint cosmic rays the stack is compared to the calibrated OTA data in the original pixel sampling. This requires finding the region of the stack which overlaps the original OTA and resampling to match it.

The matching should also include matching the PSFs. However, PSF matching requires sophisticated algorithms as noted earlier. The PSF matching is only significant within sources. Outside of sources transient detection will still be fully sensitive and within sources the difference detection algorithm can require higher significance to avoid identifying PSF differences as transient sources.

5.14.2 Difference Detection

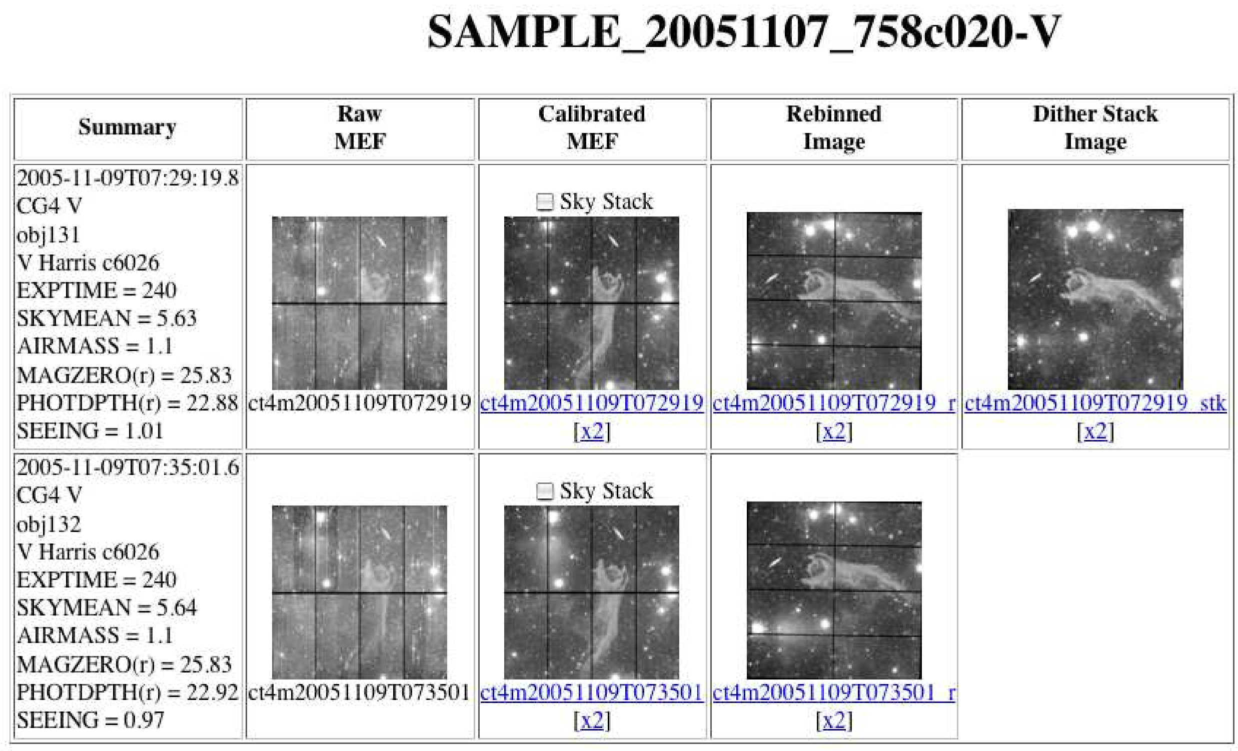
Difference detection is similar to standard source detection except that two images are used and only pixels which are significantly different are detected. Note this is more sophisticated than simply creating a difference image and running a standard source detection algorithm because the statistics of the two images are derived and considered independently.

In the absence of PSF matching, the detection algorithm includes a flux ratio parameter. When a tentative detection of a transient source is made, photometry using the same apertures on both the difference and the reference image (the stack) alone is compared. A requirement is applied on how much flux in excess of the reference image is in the difference. This algorithm has proven to be quite effective at eliminating detections of the cores of brighter stars in mismatched PSF images. This stellar core detection is by far the biggest effect with non-PSF matched data. The algorithm is effective at identifying transients within fainter sources while still finding particularly bad cosmic rays near the cores of bright stars.

5.15 SCI Pipeline

This is the parent pipeline for additional science processing of the final calibrated individual and stacked exposures. The dataset is the collection of all files processed in the FTR pipeline to this point. It primarily farms out individual exposures and stacks to instances of science subpipelines.

This pipeline and its children add the Tier 2 capabilities identified in [1] which are not already incorporated in other pipelines. One example is user requested filtering. Another possibility is generating final source catalogs.



SAMPLE_20051107_758c020-V

Summary	Raw MEF	Calibrated MEF	Rebinned Image	Dither Stack Image
2005-11-09T07:29:19.8 CG4 V obj131 V Harris c6026 EXPTIME = 240 SKYMEAN = 5.63 AIRMASS = 1.1 MAGZERO(r) = 25.83 PHOTDPTH(r) = 22.88 SEEING = 1.01	ct4m20051109T072919	Sky Stack ct4m20051109T072919 [x2]	ct4m20051109T072919_r [x2]	ct4m20051109T072919_stk [x2]
2005-11-09T07:35:01.6 CG4 V obj132 V Harris c6026 EXPTIME = 240 SKYMEAN = 5.64 AIRMASS = 1.1 MAGZERO(r) = 25.83 PHOTDPTH(r) = 22.92 SEEING = 0.97	ct4m20051109T073501	Sky Stack ct4m20051109T073501 [x2]	ct4m20051109T073501_r [x2]	

Figure 3: Example piece of MOSAIC pipeline review html page.

5.16 DPS Pipeline

The DPS pipeline receives a list of all files produced by the pipeline for the parent FTR dataset. It distributes data from each exposure to multiple EDP pipelines. It creates summary documentation for later review. One form of documentation is HTML pages with data quality metrics and image graphics at various scales. Figure 3 shows an example from the MOSAIC pipeline. On the left is some exposure information with data quality measures. Then there are graphics for various data products produced by the pipeline. Links provide access to headers and higher resolution graphics.

5.17 EDP Pipeline

The EDP pipeline produces the data products for a single exposure. This includes setting the data formats and required header information. Stacks are associated with the first exposure in the stack and are also handled by this pipeline for uniformity. The data products may be sent to the archive by this pipeline. Note that archive submission and data transport are potentially different functions.

5.18 DTS Pipeline

The DTS pipeline collects data from the pipeline and transports it to a desired destination. This is where ftp staging of non-archived but PI desired data products is handled. Note the distinction with archived data products which are standard products of the pipeline and are delivered to the archive by the EDP pipeline.

This pipeline also deletes all files from the dataset which are not saved.

6 Error Handling

Defining what happens when an error occurs adds even more complexity to the already complex parallel and distributed data flow. For example, what if one OTA has bad data or if an algorithm fails to converge? One doesn't want the pipeline to stop but one also needs to define what should happen. In this version of the data flow design we only note that this is an important consideration.

One useful contribution to this topic is that the MOSAIC pipeline has identified, often from experience, and addressed many error conditions and adopted error handling strategies. These strategies will enter into an implementation of the ODI pipeline which is builds on the MOSIAC pipeline.

7 Examples

Two examples are presented in this section to demonstrate how dataset grouping, sequencing, and parallelization work in the data flow. The first example describes the processing for a dataset consisting of 85 exposures from two nights. Figure 4 shows an annotated and compressed list of the exposures using names indicative of the night, type of exposure, filter, and field with the brackets for ranges. These names are intended to make it easier to follow the example and are not something expected for user data.

The second example describes the state of the pipeline application at an instant in time as displayed (see figure 6) in a pipeline monitor GUI . The example illustrates how a dataset is broken down into other datasets and that each dataset is a unique pipeline instance with independent state.

7.1 Setting up the data flow in the TOP pipeline

The 85 exposures submitted to the TOP pipeline go through some setup steps before the calibration and science processing actually begins. The first step is staging the exposure files referenced by input dataset list. How this is done depends on the interface between the data source (an archive or data stream from the telescope), the raw data format, and optimizations that might be made by staging pieces of the data in an parallel or distributed manner.

Whether or not the data is fully staged, the TOP pipeline will access the key observation meta-data for setting up the data flow. In order to do this it must recognize some things like exposure types and filters. This involves checking if this information is present and understandable. It

Figure 4: Starting TOP dataset.

```
n1.bias1.[1-10]   - night 1 first bias sequence of 10
n1.bias2.[1-10]   - night 1 second bias sequence of 10
n1.rdome1.[1-10]  - night 1 r-band dome flat sequence of 10
n1.Bdome1.[1-10]  - night 1 B-band dome flat sequence of 10
n1.rfield1.[1-5]  - night 1 r-band dither on field 1
n1.rfield2.[1-5]  - night 1 r-band dither on field 2
n1.rfield[3-8].1  - night 1 r-band single exposures of fields 3-8
n1.Bfield1.[1-5]  - night 1 B-band dither on field 1
n2.bias1.[1-10]   - night 2 bias sequence of 10
n2.Bdome1.[1-10]  - night 2 B-band dome flat sequence of 10
n2.Vdome1.[1-10]  - night 2 V-band dome flat sequence of 10
n2.Bfield2.[1-5]  - night 2 B-band dither on field 2
n2.Bfield2g.[1-5] - night 2 B-band dither on field 2 (guided)
```

might also involve some remediation. An example of remediation would be standardizing the filter names which might have been setup or entered differently at the telescope; this is a situation that unfortunately occurs with many NOAO instruments.

After verifying the key metadata the TOP pipeline groups the submitted dataset into into the 7 datasets shown in figure 5. Datatset1 is all the bias exposures. Dataset[2-3] are dome flat fields grouped by filter. Dataset[5-6] are static on-sky exposures grouped by filter. Dataset7 are guided on-sky exposures grouped by filter.

The science exposures within a filter may be further grouped by night depending on the number of exposures taken on each night of the block. A minimum number of exposures for a dataset is a pipeline parameter which we set to 15 for this example. If a night has less than 15 exposures it is grouped with the next night, and so on. In the groupings from figure 5 dataset6 shows two nights treated as one dataset. In this example there is no case of separating all data from a filter into multiple nights since there are no filters with 15 or more exposures per night.

The TOP pipeline does another kind of setup consisting of defining standard resampling tangent points and overlapping exposures. Overlapping exposures are identified using both a clustering heuristic based on the telescope pointing information and keywords set at the telescope indicate dither sequences. In this example 8 distinct fields are found. Fields 1 and 2 have 5 exposure dithers taken in two filters on two nights. Fields 3 to 8 are single pointings. All the exposures have tangent points assigned from a grid of tiling tangent points; for example from a Healpix tiling with a resolution of about 1 degree. For fields 1 and 2 the algorithm ensures that even if individual exposures might be nearer to different tangent points, because they are dithers they will all have the same tangent points. In the example, 4 exposures in field 1 are nearer to (2:35,32:00) and the other 6 are nearer to (2:35,33:00). The algorithm assigns all 10 to exposures in field 1, from both nights and both filters, a tangent point of (2:35,33:00).

Figure 5: Grouping the TOP dataset.

```
Biases           Dome Flats        Static Exposures  Guided Exposures
===============  ================  ================  =================

dataset1         dataset2          dataset5          dataset7
---------------  ----------------  ----------------  -----------------
n1.bias1.[1-10]  n1.rdome1.[1-10]  n1.rfield1.[1-5]  n2.Bfield2g.[1-5]
n1.bias2.[1-10]                    n1.rfield2.[1-5]
n2.bias1.[1-10]                    n1.rfield[3-8].1

                 dataset3          dataset6
                 ----------------  ----------------
                 n1.Bdome1.[1-10]  n1.Bfield1.[1-5]
                 n2.Bdome1.[1-10]  n2.Bfield2.[1-5]

                 dataset4
                 ----------------
                 n2.Vdome1.[1-10]
```

7.2 Master Biases

The TOP pipeline first submits dataset1 to the CAL pipeline. The CAL pipeline breaks up its dataset into three bias sequence datasets which are submitted to parallel instances of the SEQ pipeline. Each instance gets a list of the 10 exposures taken as a single sequence.

Consider the first group of ten, `n1.bias1.[1-10]`, which are identified as n1.Bias1. The SEQ pipeline handling this single bias sequence creates 64 subdatasets, each having data from the 10 exposures from a single OTA. These are passed to 64 parallel instances of the OCL pipeline.

The OCL pipeline calibrates and stacks the calibrated images into a master calibration image for the OTA. For bias exposures the calibrations are primarily just overscan subtraction. A data quality characterization, both for the input OTA images and for the final stack is also carried out.

When all the OTA master bias calibrations from the first sequence are completed the SEQ pipeline submits the set, call them n1.Bias1.[1-64], to the calibration library along with the derived data quality.

When the three bias sequences have been completed by the SEQ pipelines there are 192 master bias OTA images along with associated files such as data quality masks and variance arrays. These are passed to the DPS pipeline which groups them into three datasets, one for each master bias, and sends them to parallel instances of the EDP pipeline. The EDP pipeline builds the data products, which might be MEF versions of the calibration, mask, and variance arrays as well as graphics at one or more scales. Building the data products includes setting all metadata as required by the archive and end users. The files might be something like n1.Bias1, n1.Bias1_dq, n1.Bias1_var, and n1.Bias1_png.

The data products are sent to the archive as selected by a pipeline parameter. Note that data products sent to the archive are in user/archive friendly formats which are different than what is stored in the calibration library which is optimized for access by independent OTA pipelines.

After all the EDP pipelines produce and archive the master bias data products, including the graphics files, the DPS pipeline extracts information about them and builds summary and review documentation. This is a web page with postage stamp graphics that link to higher resolution versions.

After the DPS pipeline is done the CAL pipeline calls the DTS pipeline with the list of all files generated within the CAL pipeline. The DTS pipeline selects, based on pipeline parameters, what to save and what to do with them. For example the file n1.Bias1 is placed in an ftp directory accessible by the PI as defined by metadata associated with the headers. Any files not saved are then deleted.

7.3 Master Dome Flats

The TOP pipeline waits for all the master biases to be created. When this has happened it submits the dome flat field datasets (datasets[2-4]) to parallel instances of the CAL pipeline each of which breaks up its dataset into sequences. For example dataset3, consisting of all the B-band dome flat exposures, goes to an instance of the CAL pipeline where it is separated into two sequences, one from night 1 and one from night 2. The sequences are processed by parallel instances of the SEQ pipeline which each process the OTAs in parallel.

The processing of the `n1.Bdome1` dome flat sequence, for example, is similar to that described for the master biases. The main difference is that the OCL pipeline applies calibrations appropriate for a dome flat. The OCL pipeline handling the first OTA would request a master bias for that OTA, say n1.Bias1.1, and apply it to each dome flat exposure. Dome flat processing may also require removing a pupil pattern signature.

The data quality checking for dome flat exposures is more extensive than biases. This includes checking count levels for saturation and comparing against earlier master dome flats. As with the biases, the individual master dome flats for each OTA are entered into the calibration library as well as producing data products for the archive and user.

7.4 Static On-sky Exposures

When the TOP pipeline has completed the dome calibration exposures it submits the unguided science exposures for each filter to parallel FTR pipelines. In our example two instances of the FTR pipeline are called for dataset5 (r-band from night 1) and dataset6 (B-band from the merged nights).

Let's consider dataset5. This has 15 exposures which are submitted in parallel to 15 EXP pipelines. Following the data flow for exposure `n1.rfield1.1`, the EXP pipeline does some full-field setup such as getting a reference catalog of sources centered at the pointing of (2:34:24,32:05:05) with a radius of 0.5 degrees from the USNO-B1 catalog.

The exposure data is split into 64 OTA datasets, where each OTA dataset consists of 64 cells, for parallel OTA pipelines to process. In one OTA pipeline, say the first OTA, a master bias (`n1.Bias1.1`) and r-band dome flat calibration (`n1.rFlat1.1`) are obtained from the calibration manager. All the various calibrations up through dome flattening, described in §5.7, are performed. As described in that section, part of the processing is by cell and then the cells are merged into an simpler OTA image, say `n1.rfield1.1.1`.

One of the steps in the OTA pipeline is generation of a source catalog at a depth useful for astrometric calibration. When this catalog is completed, it is sent to the parent EXP pipeline. When the EXP pipeline has received 64 catalogs it merges them into a master source catalog and matches the sources against the astrometric reference catalog. WCS solutions are performed against these matches. Then the WCS solutions, as well as photometric characterization based on the matched sources, are sent back to the OTA pipelines to be incorporated into the calibrated OTAs. For instance, a magnitude zero point of 26.33 is associated with the exposure.

When the 15 first night r-band exposures have returned from the EXP pipeline, the set of exposures is next processed by the SKY pipeline for possibly deriving dark sky calibrations. This has many steps which we will simply described as follows. The 15 exposures are stacked in detector coordinates using object masks generated by the OTA pipeline along with any bad pixel masks. If the exposures have been sufficiently dithered and cover several fields the stack will consist of sky estimates at all pixels. From this stack an iterative process first extracts a pupil ghost pattern (if necessary), subtracts it from each exposure, produces a new stack, extracts a fringe pattern (if necessary), subtracts it from each exposure, produces yet another stack, and divides it into each exposure as dark sky illumination calibration. If at each step it could not produce a suitable calibration a library calibration is used. If it does produce a calibration it is entered into the calibration library, again by OTA, for use by the dataset and any later datasets.

This complex data flow in the SKY pipeline includes places where there are 64 parallel OTA specific pipelines (PGR, FRG, SFT) and 960 (= 64 * 15) parallel single exposure / single OTA pipelines (SPG, SFR, SSF).

A final step in the SKY pipeline is the RSP pipeline. This receives a dataset with the 960 OTA images that have now been calibrated (if possible) for pupil pattern, fringing, and dark sky illumination. They were also astrometrically calibrated earlier in the EXP pipeline and the resampling tangent points were assigned by the TOP pipeline. Depending on pipeline configuration parameters, none, only overlapping, or all exposures are resampled to an pipeline defined standard orientation. In our pipeline example, the exposures are oriented with north up and east left with a scale of 0.1 arcseconds per pixel. The resampling uses sinc interpolation using the data quality masks to minimize ringing from bad data. The resampling, which for a high quality interpolator like a sinc function is slow, is done in parallel by OTA and exposure.

Of the 15 exposures in dataset5 there are two fields, each with five exposures that overlap. All the pieces, the OTAs both resampled and not, are passed to two parallel instances of the STK pipeline. Dataset `n1.rfield1` has 5 * 64 * 2 science images along with masks and variances. The set of resampled pieces are median stacked with masking and outlier rejection to form a harsh stack defining the static sky during the set of observations. Remember that by design all the OTA images have a common sampling on the sky and so the stacking requires only integer shifts of the pixel arrays.

The 320 OTA pieces in the original detector sampling are submitted in parallel to the OSK pipeline along with the harsh stack. Remember that it is pointers to the data that are referenced in a dataset so it is not like copying the harsh stack 320 times.

The harsh stack is cropped and resampled to match the OTA under consideration in an OSK pipeline. Sources are detected in the difference of the original single OTA and matching piece of the harsh stack. The detected difference (transient) sources are added to a mask for that particular exposure and OTA.

The unresampled OTA image is resampled again using the mask and an algorithm that either replaces the sources by interpolation or uses and interpolator that can ignore bad data to avoid ringing.

The STK pipeline has been waiting for all the OTAs to be cleaned of transients. When this is accomplished a final science quality stack is created. This makes use of the information in the masks to eliminate bad data and the good data is averaged without outlier rejection.

Once the stacks, say `n1.rfield1_stk` and `n1.rfield2_stk`, has been created the FTR pipeline moves on. The data have now been fully calibrated to the Tier 1 requirements along with the creation of stacks. This calibrated data is now submitted to a science pipeline, aka the Tier 2 pipeline. This pipeline, for the single filter dataset, creates catalogs or special versions of the data products.

The final stages of the FTR pipeline are the same as described for the CAL pipeline. The data products are created, archived, and a summary report created by the DPS and EDP pipeline. This is followed by the DTS pipeline which does the final disposition of the data. In this example let's say the data products are put in a PI ftp area as well as being archived. The FTR pipeline deletes all remaining files produced by itself and children.

7.5 On-sky Guided Exposures

After the TOP pipeline has completed the static exposures, which are processed first in order to derive dark sky calibrations, the guided exposures are submitted to the FTR pipeline. In our example there is only dataset7 consisting of 5 dithered exposures from night 2 in field 2 taken in the B filter.

The FTR pipeline submits each exposure in parallel to the EXP pipeline. Each EXP pipeline further parallelizes the processing to individual OTAs. The processing in the OTA pipeline differs from the static exposure processing in three significant ways. The first is convolving (as appropriate) the calibration images from the calibration library by the shift history of the exposure. The second is that the OTA pipeline also applies any dark sky calibrations available in the calibration library. Finally, after all the calibrations are completed, including the astrometric calibration done in the same way as for static exposures, the OTA data is resampled for stacking and standard orientation data products. The data flow for the guided exposures is more streamlined since all the calibration operations are performed in the OTA pipeline unlike for static exposures where they are performed in the SKY pipeline.

Once the EXP pipeline is completed for each exposure in dataset7 the remainder of the data

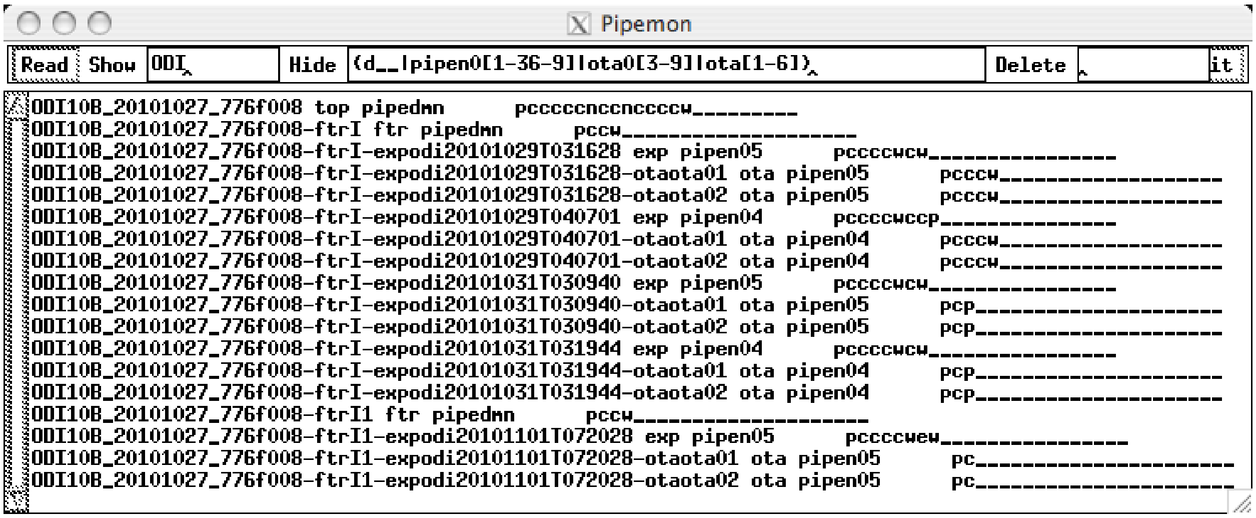


Figure 6: Pipeline Monitor Snapshot

flow is the same as for the static exposures. This means stacking in the STK pipeline, since the observations consist of a five pointing dither, Tier 2 processing, and creating and disposing of data products and intermediate files.

7.6 Finishing up in the TOP pipeline

When the TOP pipeline has processed all the data in the original dataset of 85 exposures it does a final clean up. This mostly consists of removing the raw data from the staging area(s) and removing the internal files it generated. As the final step it sends a signal to whatever agent called it, let us say by a Pipeline Scheduling Agent (PSA), that the dataset was finished. The PSA records completion of the dataset and sends another dataset to the TOP pipeline from its database of programs.

7.7 Example Pipeline State

Figure 6 provides another way to represent the data flow of the ODI pipeline. This shows the state of the pipeline at a particular instant as visualized by a simple pipeline monitor from the MOSAIC pipeline. This figure is a mock-up produced by sending status messages as if the ODI pipeline was actually running. The monitor is set to only show active ODI datasets for OTAs 01 and 02 running on cluster nodes pipen04 and pipen05. There is a lot of information encoded in this display. The salient feature we consider here is the relationship between the pipelines and datasets.

Each line in the monitor describes the state of a unique dataset being processed by a pipeline instance on a node. The whitespace delimited fields are the dataset name, the pipeline identifier, the node name, and the status flags for the stages in the pipeline.

The dataset names can be arbitrary but generally follow a convention that is easily interpretable by the operator. The convention here expands the top level dataset name with information about the pipeline hierarchy. Each hyphen delimited element starts with the pipeline identifier followed by a dataset identifier. For example

```
ODI10B_20101027_776f008-ftrI-expodi20101029T034139-otaota07
```

is a dataset from the OTA pipeline working on OTA07, which is a subpipeline of the EXP pipeline operating on exposure odi20101029T034139, which is part of an I-band dataset being processed by the FTR pipeline. The TOP pipeline dataset is from a block of nights beginning on 2010-10-27.

This example shows a TOP pipeline calling parallel FTR pipelines for all the filters in the block of nights. Two subsets from the I filter are active. Each of the FTR pipelines has called instances of the EXP pipeline for each exposure. Three EXP pipelines are active on the two nodes being displayed; other nodes are also active but not shown. Each active exposure has been decomposed in 64 OTAs, each submitted to instances of the OTA pipeline. There are two active OTA pipelines for each EXP pipeline are shown from the displayed cluster nodes.

The last part of each line shows the state of each stage in the pipeline; where some of the states are 'p' for processing, 'c' for completed, 'w' for waiting, and 'e' for error. The first letter is for the pipeline as a whole and the others are for individual stages. One point to notice is that there are no more than two 'p' stage states. This is the parallelization limiting described earlier.

8 Conclusions

There are a number of conclusions which we have tried to convey.

- The computational modules are mostly straightforward and it is the higher level data flow design which is the main challenge for a a high performance pipeline.
- How datasets are decomposed, distributed, and parallelized to achieve high performance is subtle and places requirements on a pipeline framework beyond simple job queuing.
- The experience with the NOAO MOSAIC pipeline can strongly inform the design and operation of the ODI pipeline.
- The ODI pipeline can be efficiently implemented making use of infrastructure and pipeline application components already developed at NOAO.

References

[1] I. Dell'Antonio, D. Durand, D. Harbeck, K. Olsen, J. Salzer, and P. Martin. One Degree Imager Pipeline Software and Archive Science Requirements Document. Panel Report V1.5, WIYN Observatory, Jul 2009. http://www.wiyn.org/ODI/technical/PASDR_V1_5.pdf.

[2] D. Harbeck et al. ODI and QUOTA. Report to the Board, WIYN, Oct 2006. ftp://ftp.noao.edu/pub/WIYN/QUOTA_ODI_BOARD_06B.pdf.

[3] G. H. Jacoby, J. L. Tonry, B. E. Burke, C. F. Claver, B. M. Starr, A. Saha, G. A. Luppino, and C. F. W. Harmer. WIYN One Degree Imager (ODI). In J. A. Tyson and S. Wolff, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4836 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 217–227, Dec. 2002. http://www.wiyn.org/ODI/jacoby_spie2002.pdf.

[4] LSST Project. LSST Data Management Pipelines. Web Page, LSST Corporation. http://www.lsst.org/lsst/science/datamgmt_pipelines.

[5] E. Magnier. The Pan-STARRS PS1 Image Processing Pipeline. In *The Advanced Maui Optical and Space Surveillance Technologies Conference*, 2006. http://adsabs.harvard.edu/abs/2006amos.confE..50M.

[6] J. J. Mohr et al. The Dark Energy Survey data management system. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7016 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, July 2008. http://cosmology.illinois.edu/DES/documents/SPIE2008-DESDM.pdf.

[7] J. Rose et al. The OPUS Pipeline: A Partially Object-Oriented Pipeline System. In R. A. Shaw, H. E. Payne, and J. J. E. Hayes, editors, *Astronomical Data Analysis Software and Systems IV*, volume 77 of *Astronomical Society of the Pacific Conference Series*, pages 429–+, 1995.

[8] R. Swaters and F. Valdes. The NOAO High-Performance Pipeline System: The Mosaic Camera Pipeline. SDM Pipeline Document PL009, NOAO/SDM, Nov 2006. http://chive.tuc.noao.edu/noaodpp/Pipeline/PL009.pdf.

[9] R. A. Swaters, F. Valdes, and M. E. Dickinson. The NOAO Newfirm Pipeline. *ArXiv e-prints*, Feb. 2009. http://adsabs.harvard.edu/abs/2009arXiv0902.1458S.

[10] F. Valdes, T. Cline, F. Pierfederici, B. Thomas, M. Miller, and R. Swaters. The NOAO High-Performance Pipeline System. SDM Pipeline Document PL001, NOAO/SDM, Oct 2006. http://chive.tuc.noao.edu/noaodpp/Pipeline/PL001.pdf.

[11] F. Valdes, D. Scott, N. Zarate, and R. Swaters. NOAO Science Pipelines Operations Model. SDM Pipeline Document PL012, NOAO/SDM, Jan 2008. http://chive.tuc.noao.edu/noaodpp/Pipeline/PL012.pdf.