

An On-the-fly Calibration Pipeline for the HST WFPC-2 using OPUS

Michael S. Swam, Daryl A. Swade

Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218

Abstract. An On-the-Fly Calibration System (OTFC) is being developed for selected instruments of the Hubble Space Telescope (HST) with a primary goal of providing HST Archive users with the most up-to-date calibration in their archive retrievals. The first instrument selected for this project is the Wide-Field/Planetary Camera-2 (WFPC-2), and a calibration pipeline for the WFPC-2 has been developed for OTFC using the OPUS pipeline architecture. This paper will describe the design of the OTFC WFPC-2 pipeline and the many benefits of using OPUS, including distributed multi-processing capabilities, reliable and robust function (the HST pre-archive calibration pipeline has been using OPUS for almost 3 years), high levels of code re-use, ease of integration of third-party products, and short development timescales.

1. Focus

The goals driving the development of an On-the-Fly Calibration system at the Space Telescope Science Institute (STScI) are covered elsewhere (Lubow & Polizzi 1999). This paper will focus on the technical details of the OTFC pipeline design.

The WFPC-2 was chosen as the first instrument for implementation due to its maturity and its stable calibration software base. Other HST instruments will be added to the system in the future, but WFPC-2 was selected first so that the OTFC support software could be developed with the fewest instrument-driven complications.

2. What is OPUS?

The OPUS architecture, described in detail in other papers¹ was designed and developed by the Data Processing Team at STScI. OPUS provides the framework and tools for building and operating a data reduction system that can spread processing across multiple nodes in a distributed cluster of machines. It supports parallel processing using the blackboard² paradigm of interprocess

¹http://www.dpt.stsci.edu/dpt_papers/opus_bib.html

²<http://www.stsci.edu/software/OPUS/bb.html>

communication. In addition to processing HST data at STScI, other missions are now using the OPUS software for their pipelines, including FUSE, Integral, and in the future, AXAF and SIRTf. OPUS can address the needs of a variety of missions since it is already written and released (Swade & Rose 1999), is consistently maintained by STScI, and is easily adapted to different requirements.

3. Why OPUS?

The decision to use OPUS for the OTFC pipeline was driven by several main points. The existing, pre-archive calibration pipeline that transforms WFPC-2 telemetry from HST into calibrated datasets for the HST Archive uses OPUS, and has for almost three years. The system has been robust, and provides a graphical user interface for the management and monitoring of the data processing stream, which is very useful in environments with a large data volume.

Another main driver was the concern for system scalability. The parallel, multi-processing capabilities of OPUS provide room for scaling the hardware architecture to meet the demands of the pipeline. Providing a user-friendly service like on-the-fly calibration will virtually guarantee that requests for recalibrated archive data increase. OPUS allows a system designer to add processors and disks with minimal reconfiguration to the system, resulting in a true, scalable architecture. Concerns about the performance of an OTFC system dictate that scalability be addressed, since unacceptable performance could reduce usage.

4. The Pipeline

The OTFC WFPC-2 pipeline was targeted for a relatively short development timescale (less than 1 year). The short timescale accentuated the benefits of code re-use that would be gained by using OPUS, along with existing STSDAS (Space Telescope Science Data Analysis System) tasks. Under OPUS, it is easy to string together a series of programs, shell scripts, and tools, to form a pipeline. Each stage in the OTFC pipeline performs a specific function, and the separation of work allows for greater parallel processing. Each of these pipeline tasks was implemented in a UNIX shell script. Work is accomplished using standard UNIX system commands, OPUS software tools, and STSDAS tools. The pipeline tasks are:

POLWF2 Polls for incoming OTFC requests from DADS (the Data Archive Distribution Service) at STScI.

CPYWF2 Copies the files listed in the OTFC request from a DADS disk to the OTFC system. This may be removed in the future when OTFC and DADS are merged into a single cluster of machines.

TRLWF2 Convert the observation trailer file (a log file of messages from the pre-archive pipeline processing) from FITS format to ASCII, so that OTFC pipeline messages can be appended.

KEYWF2 Update the header keywords of the files in the dataset by running the STSDAS ‘getref’ and ‘upref’ tasks. The getref³ task reads several databases at STScI to obtain a list of the most up-to-date reference files, calibration switches, and keyword fixes. This stage is crucial for producing better calibration products than those currently stored in the HST Archive. The ‘upref’ tool takes this list of changes (documented in the “delta” file), and applies them to the files in an observation dataset.

F2GWF2 Because the current STSDAS CALWP2 executable requires input in GEIS format, the FITS files retrieved from DADS are converted to GEIS.

CALWF2 The STSDAS calibration executable, CALWP2, is run against the dataset and its updated keyword values, producing new calibration products.

G2FWF2 The GEIS output from CALWP2 is converted back to FITS format, along with the ASCII trailer file and the delta file. DADS only distributes FITS files to the end-user.

RETWF2 Copies the FITS files in the dataset back to the DADS system, via FTP, as dictated in the OTFC request.

RSPWF2 Generates an ASCII OTFC response file for DADS, describing the additions that were made to the fileset by the OTFC pipeline, and indicating the status of the OTFC processing.

The pipeline consists of 9 shell scripts totaling 553 lines of code, with much reuse between modules in the areas of setup and error handling. The existence of generic OPUS software tools and STSDAS tools made this possible. The STSDAS tools perform the bulk of the processing, while the OPUS tools facilitate passing the datasets through the OTFC pipeline. A working prototype of the pipeline was up and running within two weeks and evolved into the production system.

Automated error detection was added to the pipeline in the form of three short UNIX shell scripts. These scripts are reconfigured for each pipeline stage using the resource file feature of OPUS (explained below). They detect errors in processing and send along information to DADS so that the end-user will be informed of a delay in receiving part of their archive retrieval.

5. Benefits of OPUS

The entire OTFC pipeline could be constructed as a series of shell scripts WITHOUT using OPUS, but the benefits of OPUS are numerous. Each pipeline stage can obtain values from environment variables passed in by OPUS from an ASCII setup file, called a resource file. These resource files facilitate changing task parameters without changing code, and allow the same shell script to be used for processing different kinds of data. Parameters including disk directories, file

³<http://ra.stsci.edu/bps/cdbsd/doc/getref.html>

masks, and other instrument-specific information are passed into the script, so that different versions of the script or large IF-THEN-ELSE blocks are avoided.

OPUS provides distributed, parallel, multi-processing capability, so that any stage of a pipeline can be run on any node in the cluster, as long as each node can see the needed data disks. Multiple copies of processes can be running on a single node, or spread across multiple nodes. By processing observations in parallel, the machines in the cluster are more likely to be under a balanced load (CPU and I/O), improving their throughput efficiency. Experiments and experience at STScI have shown that this parallelism increases the throughput of the system when processing a set of observations. Processing of any single observation takes a bit longer, but the staggered pipeline design (much like those used in pipelined CPU chip architectures) provides better throughput, since processing of different observations in different pipeline stages can overlap. For OTFC, the calibration stage (CALWF2) is the bottleneck in the pipeline. Running 3-5 copies of CALWF2 per processor has demonstrated⁴ better throughput for the system. Of course, there is a point of diminishing returns, where adding more copies of processes result in more swapping behavior than additional useful work. Experimentation provides the best measure for this threshold where performance starts to degrade.

OPUS multi-processing also allows built-in redundancy, so that if a dataset manages to crash one of the OTFC processes, other copies of the same process are running that can continue to operate on new datasets. This redundancy also applies to a full system crash. If hardware problems bring down one of the processors in a cluster, processes distributed to other machines through OPUS can continue to run and maintain a level of system throughput.

6. Conclusions

At this time, the OTFC WFPC-2 pipeline is in formal testing and will be fielded for in-house retrievals within STScI in January 1999. After a period of satisfactory in-house testing, the WFPC-2 instrument team at STScI will decide when it is appropriate to open up WFPC-2 OTFC retrievals to the full scientific community.

The next phase of the project will be to integrate the recalibration of Space Telescope Imaging Spectrograph (STIS) data into OTFC. The reuse from the WFPC-2 pipeline code achieved in this integration effort is expected to be greater than 90 percent.

References

- Lubow, S. & Pollizzi, J. 1999, this volume, 187
Swade, D. A. & Rose, J. F. 1999, this volume, 111

⁴http://www.dpt.stsci.edu/otfc/benchmark_cal.html