Astronomical Data Analysis Software and Systems X
ASP Conference Series, Vol. 238, 2001
F. R. Harnden Jr., F. A. Primini, and H. E. Payne, eds.

The Networked Telescope: Progress Toward a Grid Architecture for Pipeline Processing

Raymond Plante, Dave Mehringer, Damien Guillaume[1], Richard Crutcher[1]

National Center for Supercomputing Applications

Abstract. Pipeline processing systems for modern telescopes are widely considered critical for addressing the problem of ever increasing data rates; however, routine use of fully automated processing systems may discourage the typical user from exploring processing parameter space or trying out new techniques. This issue might be particularly important with regard to radio interferometer data in which the post-calibration processing required to create an image for scientific analysis is not well defined. We describe our architecture for the BIMA Image Pipeline which attempts to address this issue. The pipeline by default is automated and uses NCSA supercomputers to carry out the processing; however, as we further develop the system, we will progressively add tools that enable the astronomer to guide the automated processing. This same system can also be used by the astronomer to create new processing projects using data from the archive. We report our progress on this system, highlighting the design features that allow for greater user interaction, including a research-oriented data model, web-based portal interfaces, and use of the AIPS++ toolkit. The ultimate goal is to evolve the system into a flexible, computational grid for processing radio interferometer data.

1. Vision: The Data Life Cycle within a Grid-based Ecosystem

The BIMA Data Archive was built to deliver data automatically in real–time from the BIMA interferometer to a repository at NCSA where it can easily be accessed from NCSA supercomputers for high–performance processing or delivered to astronomers via the Web for local processing (Crutcher 1994, Plante & Crutcher 1997). We are now in the process of expanding the archive system to support automated calibration and construction of images from the raw visibility data, a process traditionally done interactively by the investigating astronomer. Pipeline processing of modern astronomical data is a problem well-suited to a Grid environment because of the inherent distributed nature of the hardware, software, data, and people involved. As part of NCSA's efforts to build Grid infrastructure, we have adopted a Grid model for implementing the BIMA Image Pipeline.

[1]Astronomy Department, University of Illinois Urbana–Champaign

Our vision of a grid for radio astronomy starts by viewing the flow of data not strictly as a pipeline but rather as a cycle. The cycle begins when an astronomer turns an idea into an observing proposal and plan. When the data are generated by the telescope, they can be transfered through a variety of channels to multiple, distributed archives, and then processed using multiple, distributed compute engines. Further analysis by the astronomer often incorporates information from a variety of network-based services (that need not be tightly integrated with the rest of the Grid). Finally, the results and data are published to feed ideas into new research projects. Much of our previous work with the BIMA Data Archive and the Astronomy Digital Image Library[2] (ADIL; Plante et al. 1996) concentrated on data archiving, publishing, and delivery; we are now turning our attention to problems of distributed computing.

Our pipeline is motivated by the same issues that are driving other pipelines in use and in development today. These issues have been described in previous ADASS papers; however, a few of the motivators are worth highlighting. Ever-increasing data production rates driven by improvements in hardware threatens radio astronomy just as it does other fields. As new millimeter arrays come on-line (CARMA, ALMA), there is a greater need to understand the data as least as fast as they are being produced. Furthermore, a pipeline that operates on new data from the telescope can just as easily be applied to data from the archive; thus, value is added to the data when the pipeline enables archival research. Finally, when the pipeline incorporates high-performance computing resources, not only can we tackle larger observing projects, we can explore more of the processing parameter space. This can be important when processing parameters are not well-defined, as is typically the case in radio astronomy.

Building the pipeline within a grid environment adds value as well. Other papers in this volume describe what constitutes a "grid" and why it might be useful; however, again, a few reasons are worth highlighting. First, a grid-based pipeline can provide users flexible access to high-performance computing. It can provide the infrastructure needed to integrate data from diverse sources. People are also an important component of a grid; thus, it can cultivate a community for developing and disseminating new processing techniques. These motivators extend beyond the BIMA community, which is why we have been collaborating with NRAO to develop a general blueprint for a data grid for radio astronomy.[3]

2. Architecture: A Portal for Guided Automation

One approach to building a pipeline might be to take the tools one uses to process data interactively, wrap them up, and connect them together so that they can run automatically. We see our approach as the exact opposite: we want to build a system that is inherently automated and then extend it to add ever increasing amounts of interactivity, resulting in an architecture that might be described as "guided automation." Here are some ways we want to allow users to interact with the pipeline: (a) **prior to observations:** the astronomer can

[2] `http://adil.ncsa.uiuc.edu/`

[3] `http://monet.astro.uiuc.edu/papers/COBRA/NGRA_white_paper.pdf`

override default processing parameters to better suit the scientific goals of the project; (b) **during observations:** the astronomer can monitor the telescope and data via the web; (c) **after observations:** the astronomer can browse the archive's holdings using customizable displays; (d) **prior to processing:** the astronomer can create his/her own scripts for reprocessing archival data; (e) **during processing:** optional viewers can be opened up to monitor, and possibly steer, the deconvolving process.

One of the challenges to enabling all these diverse features is delivering interfaces to users over the network. This is a problem for most any type of grid; thus, NCSA has been developing a framework for *scientific portals*. A scientific portal can be thought of as a collection of network-based services and documents integrated into a single, customizable web environment for the purpose of conducting scientific research. NCSA has recently released its first version of such a framework called the *Open Portal Interface Environment* (OPIE).

3. Data Management through Data-centric Programming

In many ways, the hardware and scientific processing software are the simpler parts of the pipeline. The rest of the system is about information management; thus, metadata have an important role: they drive the data through the system. We encode the metadata used to archive, process, and deliver datasets to astronomers in XML. Our schema[4] is based on a research-oriented data model that organizes data into hierarchical collections (i.e., projects, experiments, trials, and datasets). Users can browse these metadata via the web; XSLT is used to turn XML into HTML on-the-fly. The goal of the HTML rendering is to make the relationships between data and collections intuitively clear.

Our use of XML has resulted in software that is very *data-centric* (as opposed to process-centric, see Guillaume & Plante 2001). We found XML to very useful for rapid modeling of our data; object-oriented software built around entities in our model followed naturally. In this regard, our application has two important challenges: first, our model is large (we currently define about 100 objects), and second, we know it will evolve as we add new capabilities. The cost of these challenges on software can be quite high; however, our use of the Quick software package controls this cost. Not only can the package convert XML documents into intelligent Java objects, but (through our contributions to the package) it can also convert the schema definition document into custom Java classes (see Guillaume & Plante 2001).

Our pipeline is "metadata–driven;" this means that the task of processing the data is a matter of converting metadata into processing instructions. This can be handled quite effectively using XSLT: instead of converting XML to HTML, we convert it into high level Glish scripts (see §4.) that select and configure predefined processing recipes. XML is also effective for storing state information for processing that may proceed incrementally over a period of months or years as more data becomes available.

---

[4] `http://monet.astro.uiuc.edu/BIP/xml/bimaarch_dtd.txt`

4. Data Processing with AIPS++

The processing is carried out using AIPS++, which employs the Glish scripting language (Scheibel 2000) to glue processing objects together. Its event-driven programming model (combined with the toolkit nature of AIPS++) makes it ideal for building automated processing in a distributed environment. An important role for NCSA, as a member of the AIPS++ development consortium, is to enable support for parallel processing on a range of mildly to massively parallel machines, with a particular emphasis on Linux clusters. The Intel Itanium-based supercluster that will be brought on–line at NCSA this year will handle the bulk of the imaging and deconvolution chores for the pipeline, while smaller machines will handle the serial processing.

The pipeline's processing engine presents its own set of interesting challenges. Foremost is making effective use of parallel processing. Common forms of imaging and deconvolution can be broken down into naturally self-contained components that can be parsed out to separate processors. The simplest example is dividing the problem into independent frequency channels. A more interesting application to widefield imaging is described by Golap et al. (2001). MPI is used to orchestrate the parallel processing (Roberts et al. 2000). An important focus of current research is parallel I/O via MPI-2. Another challenge is making remote grid processing appear interactive within the AIPS++ interface. Grid processing in general is typically queue-scheduled, which can introduce latencies and unnatural barriers one would not see in traditional interactive environments; nevertheless, users will want to interact with processing in real-time, and so computing resources must be negotiated in real-time as well. Finally, a powerful pipeline can be an excellent tool for exploring processing parameter space from which we hope to develop diagnostic measures that can not only evaluate various processing strategies but also predict the most appropriate strategy for a given experiment. Such diagnostics would further improve the quality of the data products coming from the pipeline.

References

Crutcher, R. M. 1994, in ASP Conf. Ser., Vol. 61, Astronomical Data Analysis Software and Systems III, ed. D. R. Crabtree, R. J. Hanisch, & J. Barnes (San Francisco: ASP), 409

Golap, K., Kemball, A., Cornwell, T., & Young, W. 2001, this volume, 408

Guillaume, D. & Plante, R. 2001, this volume, 221

Plante, R. L. & Crutcher, R. M. 1997, Proc. SPIE, 3112, 90

Roberts, D. A., Crutcher, R. M., Young, W., & Kemball, A. J. 1999, in ASP Conf. Ser., Vol. 172, Astronomical Data Analysis Software and Systems VIII, ed. David M. Mehringer, Raymond L. Plante, & Douglas A. Roberts (San Francisco: ASP), 15

Schiebel, D. R. 2000, in ASP Conf. Ser., Vol. 216, Astronomical Data Analysis Software and Systems IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP), 39