Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

Status and Future Plans for Parallelization of AIPS++

Douglas A. Roberts, Richard Crutcher

National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, Urbana, IL 61801

Wes Young, Athol J. Kemball

National Radio Astronomy Observatory, P.O. Box O, Socorro, NM, 87801

Abstract. During the past year, the parallel group of AIPS++ has been actively incorporating parallel processing into the most computationally intensive aspects of AIPS++. We report here the status of the AIPS++ parallelization effort and plans for future development. The biggest success of the past year of AIPS++ parallelization has been the implementation of the parallel algorithm applicator class. This class implements parallelism using the Message Passing Interface (MPI). MPI is a portable system that allows data and instructions to be sent to remote processors (either on the same machine or on different machines). New classes derived from this parallel base class will carry out parallelization with a minimal effort from application programmers. This solution addresses most embarrassingly parallel problems, notably spectral line processing. We are also investigating tuned libraries (starting with FFT's), notably the Sgi/Cray Scientific Library (SCSL).

1. Goals of AIPS++ Parallelization

The goal of the AIPS++ parallelization project is to provide an easy to use, high performance processing system that allows astronomers to processes large datasets. Specifically our project has the following goals:

1. Give astronomers at least an order-of-magnitude increase in computational power.
    - This processing power can be used to process data sets larger than currently possible.
    - The increased capability can also be used on moderate-sized data sets to explore the use of a variety of algorithms (some of which may be computationally expensive).
2. Use a familiar interface.
    - Since we are building the parallel system within the AIPS++ system[1] we should use the same interface.

[1] `http://aips2.nrao.edu/aips++/docs/html/aips++.html`

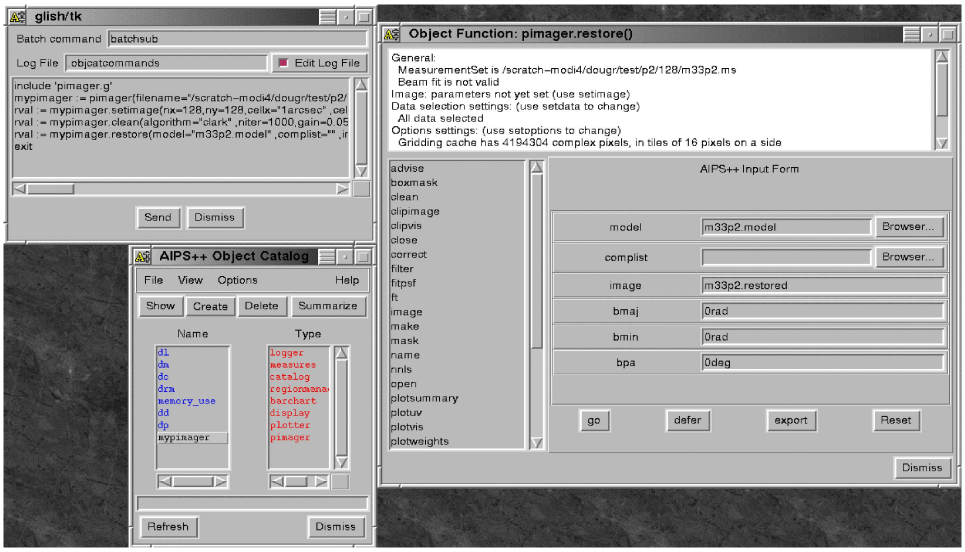

Figure 1. GUI for batch submission.

- Astronomers who use the parallel system for some portion of their processing should notice almost no difference in the interface (even though some of their processes are being carried out on a batch system).

3. Get as much code parallelized in a short time as possible.
   - This priority motivates us to implement a parallel infrastructure into the existing code.
   - Also we should make it easy for a programmer to use the parallel infrastructure.

2. State of AIPS++ Parallelization

An Algorithm-Applicator scheme has been chosen to deal with embarrassingly parallel (e.g., spectral line) problems. The Applicator is the controller, which sets up the problem, transports data to the Algorithm processes using the Message-Passing-Interface (MPI)[2]. MPI is a portable system that allows data and instructions to be sent to remote processors (either on the same machine or on different machines). Wes Young (see page 506) presented a demonstration of our first implementation of the algorithm applicator class to carry out spectral line deconvolution.

A graphical interface to the parallel system and batch processing has been implemented (see Fig. 1). The user only has to tell the application the number of processors to use. Submission to the batch queue is carried out with only one additional step.

---

[2]`http://www-unix.mcs.anl.gov/mpi/index.html`

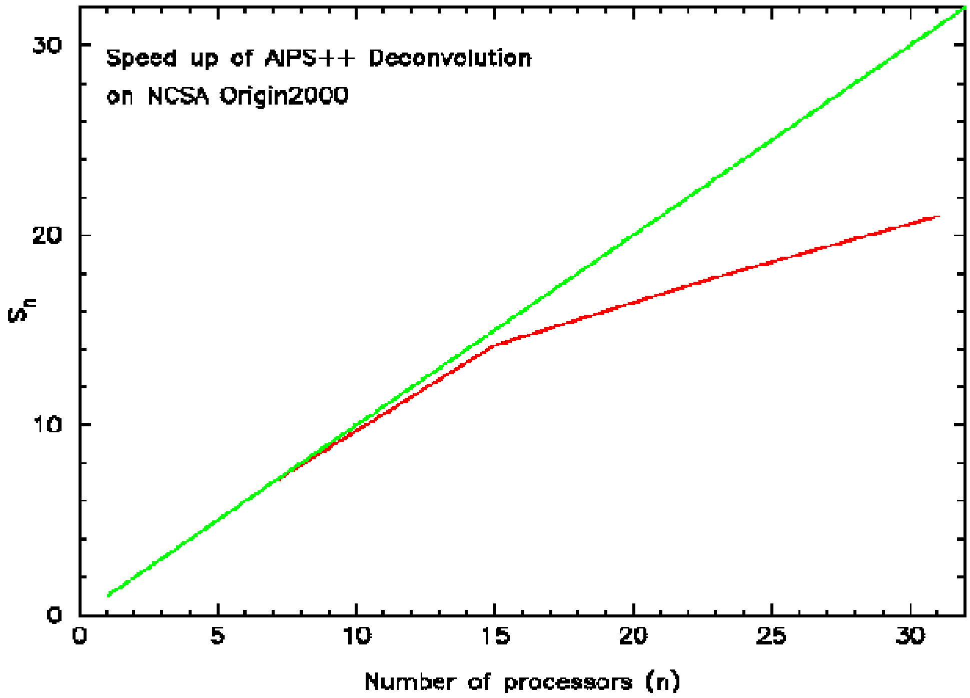

Figure 2. Clark CLEAN speed up of M33 HI 512 × 512 pixels by 100 channels using AIPS++ Applicator/Algorithm classes.

Figure 2 shows the speed up of a deconvolution as a function of processors. Currently the parallel system shows good scaling up to 16 processors.

- Lower degree of speed up beyond 32 processors may be due to I/O inefficiencies.
- Collaboration with NCSA enabling technologies team is important to obtain I/O performance statistics of the AIPS++ code.

3. Uses of Parallel AIPS++ in the Short Term

Power users can begin to use the parallel system on the NCSA computers after the first of the year. The parallel system will include parallelization of various spectral line processing: gridding, imaging, deconvolution (various flavors of CLEAN and MEM). The parallel infrastructure should work on a generic multiple processor machines.

4. Longer Term Goals

Our future goals include parallelizing the AIPS++ IMAGER application to carry out imaging, (using parallelized gridding and FFT's) and parallel deconvolution. Also, we will be parallelizing a wide-field imaging algorithm (where the assumption that the sky can be represented at a single tangent plane breaks down). Finally, we will be increasing the user support to help astronomers, who need the large computational resources of NCSA, use the parallel AIPS++ system on the Origin2000. The first step of the increased user support has been to increase the network bandwidth from the VLA to NCSA by putting NCSA on the NRAO intranet.

Future projects also include a port of AIPS++ to NT, in order to take advantage of the large NT cluster now available at NCSA and increasingly available to astronomy departments because of their modest prices. We are also collaborating with the Pablo group[3] at the University of Illinois to investigate the I/O patterns of AIPS++. Their group has instrumented our code and is now working to identify I/O patterns and statistics. This will be important to show where better performance due to increased I/O would be possible. The next generation of the MPI standard (MPI-2) includes a standard for MPI I/O. The MPI I/O standard is finalized and implementations are now available. We intend to explore its use as a complement to the parallel processing development.

We also are looking into integration of batch communication into the Glish IPC. Continue to use the parallel algorithm applicator to implement more algorithms in parallel.

In addition to post-observation processing, our project may include parallel on-line processing. The JCMT has decided in principle to use AIPS++ to process the backend of the new 1 GHz correlator. The correlator will create a data rate of about 10 MB/sec (up to about 3.8 GB could be collected in a 6 minute observation). They will have a network of about 8 to 16 workstations to carry out the processing.

[3] `http://www-pablo.cs.uiuc.edu/`