Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

IMAGER: A Parallel Interface to Spectral Line Processing

Doug Roberts

University of Illinois/NCSA

Dick Crutcher

University of Illinois/NCSA

Abstract. We report on the development and use of the IMAGER system at NCSA. IMAGER is an interface to parallel implementation of imaging and deconvolution tasks of the Software Development Environment (SDE) of the NRAO. The interface is based on the MIRIAD interface of the BIMA array and it allows for interactive and batch operation. The parallelization is carried out by distributing independent spectral line channels across multiple processors. The use of this system on the SGI Power Challenge array at NCSA is presented. For most problems, the speed-up with number of processors is nearly linear. We present results from our use of IMAGER on a data reduction of multiple-pointing BIMA observations of molecular gas around the "Sickle" near the Galactic center. We also report on the general use and timing results of this system at NCSA.

1. Introduction

There are many problems in observational astronomy that can be solved only with high performance computers. Because computers are the optical element of a synthesis telescope, radio synthesis data reduction has been one of the most computer intensive operations in observational astronomy. The sizes of data sets can frequently be large. In the some cases, such as pulsar observations, short time samples lead to large data sets. In the common case of radio spectral line observations, large numbers of frequency channels lead to large amounts of data. It is not uncommon with such instruments such as the VLA and the BIMA telescopes to have spectral line data sets in excess of a gigabyte. Astronomers need access to fast processing to allow the analysis of such large data sets. It is especially important to have analysis capabilities that allow astronomers to use different methods of non-linear deconvolutions in a timely basis to properly interpret their observations.

The analysis of spectral line data, in which each channel is independent from every other channel, is an embarrassingly parallel problem. Such problems have little or no data dependence and require very little communication between individual processors. Thus, an additional motivation for this project is the need to demonstrate that parallel execution of embarrassingly parallel problems can be accomplished in a straightforward manner. In order to prototype a solution to spectral line data reduction and to provide a substantial performance increase from conventional packages such as AIPS and MIRIAD, the IMAGER package was developed.

2. IMAGER Implementation

We decided to implement a three dimensional data reduction package starting with existing software to carry out 2-D data reduction. This involved writing a user interface and control package to send the data and data reduction software for separate channels to individual processors and collect the results into a data cube.

We intended to use an efficient 2-D data reduction package and decided on the SDE package written by Tim Cornwell of NRAO. SDE code was written assuming that the visibility and image data could fit into the virtual memory. We developed IMAGER to run optimally on the SGI Power Challenge Array of NCSA. The total physical memory is 4 GB; thus almost all problems will fit into physical memory and will not require the program swapping to disk. SDE requires that visibility data be in UVFITS format with each channel in a separate file. Additionally, multiple pointing data sets have all pointings in a special "mosaic" database. Thus, functionality for data conversion is included in the IMAGER package.

In addition to the underlying 2-D programs, we needed to create an interface to the package that handled user inputs, documentation, and execution. We decided to use the MIRIAD shell for the interface to IMAGER. The MIRIAD interface allows inputs to be saved and recalled, help documents to be browsed, and IMAGER programs to be executed. The documentation and user input system includes normal and expert modes. The normal mode hides less frequently used inputs (and the associated documentation), and the expert mode provides access to all inputs and help.

The final part of the package carries out parsing the inputs, distributing the channels to the individual processors and accumulating the results into a data cube. History and logging information are also managed. Two large Perl scripts written to carry out these actions constitute the core of the IMAGER system.

These data reduction processes were implemented in IMAGER:

- **uvmap**—Inverting the visibility data into the image plane, without subsequent deconvolution (i.e., creating "dirty" images from visibility data).
- **clean**—Deconvolution of image plane data created with the **uvmap** program using the CLEAN algorithm.
- **mosaic**—Iterative imaging and MEM deconvolution; includes mosaics of multiple fields.

The data reduction steps can be carried out in a pipelined manner or they can be done one step at a time. For instance, a dirty cube can be created in one step (using **uvmap**) and saved; then **clean** can be run several times to different output files using different parameters, (e.g., clean iterations, flux limits). Alternately the **uvmap** and **clean** steps can be pipelined together and executed on one step. Both interactive and batch execution are supported.

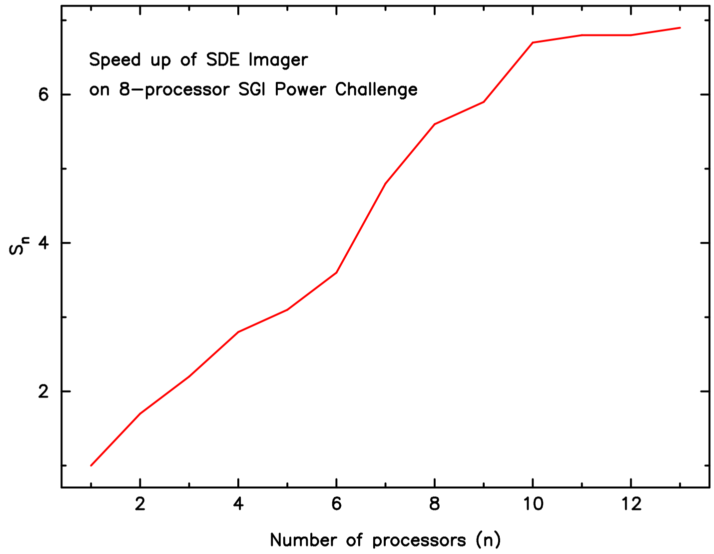

Figure 1. Speedup of IMAGER program with number of processors.

3. Results

The IMAGER system has been used by astronomers at the University of Illinois to carry out analysis of data from the VLA and BIMA telescopes. In all projects, there is an initial overhead for the data conversion into the proper format. The worst case is that of multiple-field spectral line data, in which initially data for all channels of a single pointing are in separate files (one file for each pointing). The data conversion stage separates each pointing and channel and recombines the pointings together, resulting in a separate file for each channel. In the worst case, the overhead may be 50% of the total execution time. However, the IMAGER package is intended to give astronomers the power for iterative data reduction; thus in all real cases this step is done only once and subsequent steps (those optimized for parallel execution) are carried out repeatedly.

The projects have used the SGI Power Challenge at NCSA to carry out computer processing. In addition to production scientific processing, we carried out tests in order to determine the speedup on the 8-processor SGI Power Challenge. Our test problem was the imaging (**uvmap**) and CLEAN deconvolution (**clean**) of a single pointing observation of the molecular gas (the CS line) associated with the "sickle" HII region near the Galactic center. The data were acquired with the BIMA interferometer. The visibility data set was about 300 MB and the output images were 256 × 256 pixels × 100 channels. All timing tests were carried out after the initial conversion of data formats. The number of threads were varied from 1 to 13. The speedup ($S_n = t_1/t_n$, t_1 is time for a single processor run, t_n is time using n processors) as a function of the number of threads is shown in Figure 1. The speedup is nearly linear and saturates at 7; this is one less than the number of processors because the control program is running on one processor. The slope is a bit less than one because when the control script detects that a channel has been completed, a new channel is sent, but only after a bit of delay.

Also compared in this test was the performance between the same operations carried out with the AIPS, MIRIAD, and IMAGER packages. The single processor timing between the three packages is shown in Table 1. The AIPS package was by far the slowest. However, the computational problem was carried out in AIPS in a manner that required the mapping program (HORUS) to be run once for the cube and the CLEAN deconvolution program (APCLN) run once for each *channel* (i.e., for this test 100 times). The AIPS code was written assuming a very small memory model machine; AIPS writes many scratch disk files for each program execution. The I/O caused by these scratch files dominates the execution time of the programs and is responsible for the poor performance relative to MIRIAD and IMAGER. The I/O for the various programs is given in Table 1 as well.

Table 1. Timing results between packages.

Parameter	AIPS	MIRIAD	IMAGER
Execution time (sec)	1041	247	261
I/O overhead	43%	2%	2%

4. Discussion and Future Plans

The IMAGER package is currently supported on the SGI Power Challenge array at NCSA. Astronomers are using the IMAGER system to analyze radio synthesis data. Anyone wishing further information on this can view the on-line documentation[1] or contact Doug Roberts directly.

One of the goals of this project was to demonstrate the ease of programming and use of a embarrassingly parallel data reduction package. We feel that this has been accomplished. The next step was to use this experience and design a plan for the coding of embarrassingly parallel algorithms and compute intensive algorithms of the AIPS++ project. The plan for parallelization of AIPS++ has been created. Work will begin after the beta release of AIPS++ in Jan 1997. Parallel functionality at some level should be available in AIPS++ by the time of the first full release of AIPS++ in mid-1997.

[1] http://monet.ncsa.uiuc.edu/hpcc/imager/imager.html