Exp Astron (2013) 36:59–76
DOI 10.1007/s10686-012-9315-0

ORIGINAL ARTICLE

An AIPS-based, distributed processing method for large radio interferometric datasets

Stephen Bourke · Huib Jan van Langevelde · Karl Torstensson · Aaron Golden

Received: 20 February 2012 / Accepted: 3 September 2012 / Published online: 17 April 2013

Abstract The data output rates of modern radio interferometric telescopes make the traditional data reduction process impractical in many cases. We report on the implementation of a lightweight infrastructure, named AIPSLite, that enables the deployment of AIPS interferometric processing routines on distributed systems in an autonomous and fault tolerant manner. We discuss how this approach was used to search for sources of 6.7 GHz methanol maser emission in the Cep A region with the European VLBI Network (EVN). The field was searched out to a radius of 1.25 min^{-1} at milli-arcsecond spatial resolution and 1024 frequency channels with 0.088 km s^{-1} velocity resolution. The imaged data was on the order of 30 TB. Processing was performed on 128 processors of the Irish Centre for High End Computing (ICHEC) linux cluster with a run time of 42 h, and a total of 212 CPU days.

Keywords Interferometry data processing · Distributed processing · AIPS · ParselTongue · AIPSLite

S. Bourke (✉) · H.J. van Langevelde · K. Torstensson
Joint Institute for VLBI in Europe, Postbus 2, 7990AA Dwingeloo, The Netherlands
e-mail: sb@astro.caltech.edu

S. Bourke · A. Golden
Centre for Astronomy, National University of Ireland, Galway, Ireland

H.J. van Langevelde · K. Torstensson
Leiden Observatory, Leiden University, PO Box 9513, 2300RA Leiden, The Netherlands

Present Address:
S. Bourke
Department of Astronomy, California Institute of Technology, MC 249-17, 1200 East
California Blvd, Pasadena, CA 91125, USA

Present Address:
A. Golden
Department of Genetics, Albert Einstein College of Medicine, Bronx, NY 10461, USA

1 Introduction

The landscape of radio interferometric telescopes is currently in a state of flux. Existing telescopes are being upgraded and entire new telescopes are being built deploying modern digital data transport and processing equipment. These include instruments which have been substantially upgraded such as eMerlin and the Jansky Very Large Array (JVLA), instruments which have had a series of incremental upgrades such as the European VLBI Network (EVN) and Very Long Base Line Array (VLBA), as well as a slew of new instruments which are currently being built and commissioned such as Lofar, ASKAP, and MeerKat. Common to all these instruments are their capacity to produce data volume orders of magnitudes larger than their predecessors.

Two technologies are common to many if not all of the upcoming generation of interferometers which result in large data output rates: wide band receivers and high throughput correlators. Motivated by the desire for increased sensitivity as well as coverage of previously unobserved frequency ranges, wide-band receivers are becoming the norm. For continuum sources with relatively flat spectrum over the observed band, the sensitivity of the instrument scales with the square root of the observing bandwidth. While the previous generation of interferometers often had a bandwidth of less than 100 MHz, current technology allows for receivers to cover an order of 1 octave providing 1–8 GHz of bandwidth in centimeter bands. It is desirable to subdivide the receiving bandwidth in to narrow sub-channels for a number of reasons. Firstly, this allows the mitigation of radio frequency inference (RFI)—unwanted man-made signals which are often narrow-band. Secondly, for reasons described below, imaging wide fields places restrictions on the maximum channel width that is acceptable if high fidelity images are to be produced. Thirdly, offline calibration techniques allow for accurate determination of the antenna receiver bandpass gain parameters, thus allowing correction and the applications of weighting parameters to avoid sensitivity losses, though modest frequency resolution is adequate for this purpose.

The second major advance common in the upcoming instruments is the use of high throughput correlators. A correlator is the computing system responsible for the production of visibilities, the data product of interferometers. Correlators are in concept simple devices, producing complex numbers related to the Fourier domain special frequency components of the sky brightness distribution. The implementation details and their high data throughput rates however often require them to use highly specialized hardware (see [19]). The output data rate is controlled primarily by two parameters, the number of frequency points generated, and the time averaging interval. The science result may lie in these regimes, i.e., spectral line/temporal variability but more commonly the reason for a high data rate is wide field imaging and, at low frequencies, RFI mitigation.

The relevance of high time and frequency resolution to RFI is obvious—RFI often occurs on fine scales in both time and frequency so the ability to identify and excise bad data while preserving unaffected data is highly desirable. The imageable field of view of an interferometer is limited by the optical response of the instrument, the frequency resolution, and the time resolution. The latter two are controlled by the correlation parameters and if not sufficiently fine will cause features to be smeared either radially or tangentially, respectively with extent:

$$S_{bw} = \frac{\Delta\nu}{\nu} r, \qquad S_t = \frac{2\pi\Delta t}{1\ sidereal\ day} r \tag{1}$$

where S_{bw} is the smearing due to frequency bandwidth, and S_t is the smearing due to the integration time. $\Delta\nu$ is the channel bandwidth, ν is the observing frequency, r is the distance from the delay center, and Δt is the integration time. For a full treatment of these effects, see [18]. It follows that if telescopes are to be maximally exploited by imaging as much of the sky that they are sensitive to as possible, high data rates are required. Modern correlators routinely produce in the order of Megabytes per second.

The manner in which interferometric data is processed will depend on the scientific goal of the observation. In many cases, however large parts of the processing is readily parallelizable. Data calibration is to a large extent decomposable in the time domain. In this case the data can be divided in time and calibrated in parallel. If the desired end product is a data cube—a set of maps, each at a different frequency—the imaging process is also easily parallelizable by decomposing along the frequency axis. Even in the case where a continuum map of the entire frequency band is desired the imaging can be decomposed and recombined later in the process with modest additional overhead. While parallel software is commonly used at observatories (e.g. [7, 9, 14]), its prevalence in the common user software packages is low. Of the packages AIPS, Casa, Miriad, and Difmap, only Casa aims to make use of parallel hardware. Given large volumes of data, it is desirable to use High Performance Computing (HPC) methods, such as parallel or distributed processing to calibrate and image the data in a timely fashion.

Compute clusters, the most common type of distributed system, often consist of one or more head nodes which users can access, and a large number of compute nodes. Batch jobs are submitted via a batch system on the head nodes which is responsible for scheduling the job, allocating cluster nodes and running the job. Disk storage is often two tier, with shared network space accessible from all compute nodes and local storage only accessible locally by each compute node. Access to such system is commonplace, however utilizing them for interferometric data processes is non-trivial, as the existing software is largely designed to be run on a workstation. Software which is available system-wide is limited, and local storage areas are often erased on completion of a job.

2 AIPSLite, a facility for distributed AIPS processing

The Astronomical Image Processing System (AIPS) is a well established system for end to end processing of radio interferometer observations. It is written in Fortran and has evolved over more than three decades. While it is a versatile and extremely complete system, its design does not immediately adapt well to distributed clusters. The distribution is in the order of 1GB. AIPS does not support deployment on a cluster for the purpose of distributed processing. Rather, its network deployment mode is for the purpose of central administration of data areas, tape drives, and configuration areas.

ParselTongue [10] is a set of Python modules that provides a Python interface to AIPS. It has two main functions, to allow the execution of AIPS tasks, and to provide access to the underlying data, including visibility data, images, and table data. Python's flexibility, expressiveness, and extensive library combined with the data access provided by ParselTongue allow for a high level of sophistication in automating AIPS (e.g., [2, 8]).

We have developed a set of Python modules which we named AIPSLite that extends ParselTongue and allows for machines without an AIPS distribution to bootstrap themselves with a minimal AIPS environment. Data areas may be created and destroyed at will, and AIPS tasks can be downloaded and executed on the fly. All this is performed dynamically at run-time. Pre-existing AIPS files, both AIPS binaries (tasks) and data products, may be utilized. Multiple processes running on a single node can be entirely isolated from each other.

2.1 Architectural overview

An architectural overview of system can be seen in Fig. 1. The Python application that will process data first uses AIPSLite methods to create a minimal AIPS environment. In comparison to the full AIPS environment which contains in the order of 1 GB of software, this environment is less than

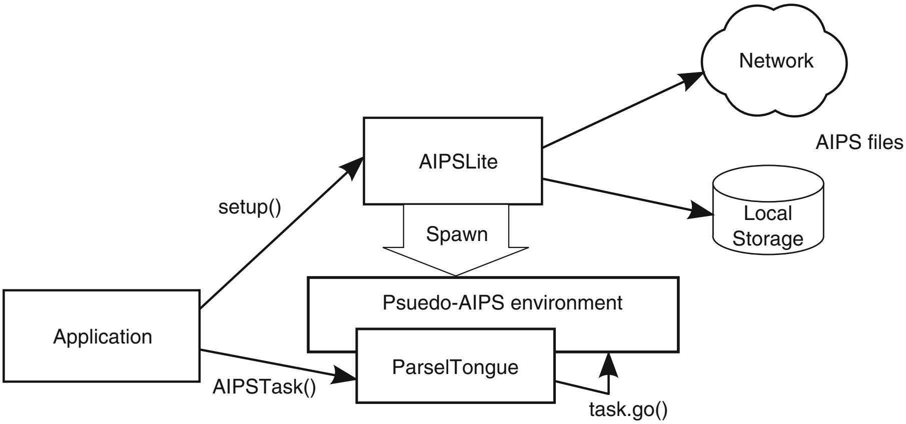

Fig. 1 Overview of AIPSLite architecture. Traditionally, a Python application using ParselTongue requires and existing working AIPS environment. This is not a requirement with AIPSLite. A Python application imports the AIPSLite module which creates the environment (binaries and configuration data) necessary to use the ParselTongue interface to AIPS. This includes downloading a minimal set of AIPS binaries if necessary

16 MB and can be downloaded in seconds. ParselTongue's modules are then loaded within this environment. To ParselTongue, this minimal environment is indistinguishable from a full AIPS distribution, with the exception of the lack of AIPS tasks. AIPS tasks are implemented as self contained executables meaning only those tasks which are to be used are required. These tasks are acquired with AIPSLite in one of two ways. AIPSLite contains a subclass of ParselTongue's AIPSTask class. The AIPSLite version checks if the task is present and if not fetches it along with its metadata[1] from the network. Alternatively, AIPSLite provides a getTask method which takes a task name or list of task names as its argument and fetches them via the network. The process of establishing the minimal AIPS environment is as follows:

1. Determine architecture and required libraries
2. Define environment variables
3. Determine AIPS version
4. Construct list of required files: libs, binaries, and metadata
5. Establish rsync connection to AIPS server and transfer files
6. Create AIPS run-time resources: Template, memory, data areas
7. Run initialization code to populate run-time areas
8. Create private configuration area, populating from template
9. Create data area(s)

At this point the ParselTongue modules may be loaded and used as normal. While this approach may be used in any situation, it is most beneficial when used on clusters. A typical use case of this infrastructure is depicted in Fig. 2 in which the minimal environment is set up by a head node and is placed on a shared disk. When the job is run by the resource manager, one of the nodes assumes the role of managing the job and is responsible for dividing and distributing the data. Division of the data by spectral channels or time interval are possible in interferometric data. The actual optimal decomposition will depend on the specific application. For this paradigm to be efficient, the time required to process the data must be greater than the that required to distribute the data among the worker nodes. In the case discussed in Section 3 the data is distributed via a shared Network File System (NFS) disk system on a gigabit Ethernet network.

In the AIPSLite system, the worker nodes set up their own configuration area (`DA00`) and AIPS data areas. They process data in an isolated environment and save the results in a shared area. This is contrary to AIPS' normal mode of operation in which all AIPS instances running on a single host share configuration and data areas. AIPSLite provides a *per-process* as apposed to a *per-host* private AIPS distribution which is set up at run time with minimal overhead. This allows for an unlimited number of AIPS environments to be

[1] AIPS Tasks are accompanied by a HLP file which contains information on the data structures used by the task which AIPS or ParselTongue requires as well as documentation on its functionality.

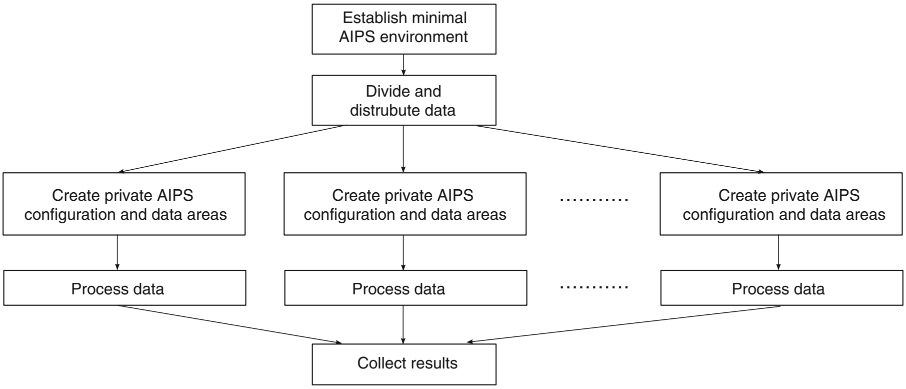
Establish minimal AIPS environment
Divide and distrubute data
Create private AIPS configuration and data areas
Create private AIPS configuration and data areas
Create private AIPS configuration and data areas
Process data
Process data
Process data
Collect results

Fig. 2 Typical usage of AIPSLite on a compute cluster. The head node is responsible for setting up the minimal AIPS environment. A master node manages the worker nodes (three in this case), distributes data, and collects output. Worker nodes will use AIPSLite methods to set up their data storage and configuration areas. The AIPS disks and DA00 in AIPS terminology

established and driven independently, and does away with the requirement to configure AIPS for each host it will run on.

Process isolation is largely facilitated in two ways. Firstly, prior to loading the ParselTongue modules, AIPSLite is used by each process to create a private DA00 area and used for the life time of that process. is initialized from the TEMPLATE directory which forms part of the minimal AIPS distribution. Each process also creates its own AIPS disk for use during its lifetime. This architecture was developed to overcome a failure mode common under high load, in which ParselTongue would fail to successfully allocate AIPS POPS numbers during early testing. If the cluster architecture supports it, both the DA00 and the AIPS disks are created in a temporary partition that is allocate by the job scheduler running on the cluster. The contents of this partition may be deleted on completion of the job by the batch system.

3 Observations

In this section an observational campaign will be described in which the EVN Mark4 correlator was utilized at 100 % capacity, and with sufficiently low time and spectral averaging to allow for full primary beam imaging. The goal of the observation (EL032) was the study of massive star formation regions via maser kinematics. The scientific objectives were two-fold, to image in high spatial and frequency resolution known sites of methanol maser emission and thereby constrain the kinematics of the systems to allow for improvement of the models of massive star formation. The second objective was to search the surrounding area for additional maser activity. Due to the high spatial and frequency resolution and wide field of view, this represents a significant data processing challenge. The software described in the previous section was applied to the problem and the specific approach is discussed in this section. For astronomical results from this campaign refer to [20].

Several sites of known maser emission were observed with the EVN. A single baseband of 2 MHz bandwidth was used. Left and right hand polarizations were recorded with 2-bit sampling at a rate of 4 MSamples/s producing an aggregate bit rate of 16 Mb/s. Stations included were Cambridge, Darnhall, Effelsberg, Medicina, Noto, Onsala, Torun and Westerbork. The experiment was correlated on the EVN MkIV VLBI Data Processor at JIVE with 1024 spectral channels and an integration time of 0.25 s (see [15]). Dual circular polarizations were produced. Each of the maser sites listed in Table 1 were observed for two periods of approximately 1 h, separated by approximately eleven hours.

3.1 Data calibration

The calibration of the data did not present computational challenges and was performed by traditional means. For the purpose of determining calibration solutions the data could be averaged by a factor of eight in time and thirty two in frequency which were then applied to the full, non-averaged data. Calibration of the data was performed in traditional AIPS. A priori amplitude calibration was carried out using the system temperature and antenna gain curves with the task `APCAL`. Phase calibration was performed by the AIPS task `FRING`. A two stage approach to fringe fitting was performed by solving for the fringe delay over two minutes of data on the field's principal bright calibrator. These solutions were then applied to the data and then full fringe delay, rate, and phase solutions were determined on the entire dataset. Self calibration was implemented using several maser sources in the field of view. Some of the maser sources were significantly brighter than the calibrator sources. In these cases the maser was used to calibrate the data with the calibrator source being

Table 1 Parameters of the target sources

Source	RA	DEC	Flux (Jy)	Velocity (LSR)	D (kpc)
G40.62-0.14	19:06:00.8	06:46:37	17	31	2.2
G78.12+3.63	20:14:26.1	41:13:31	38	−6	1.6
G73.06+1.80	20:08:09.8	35:59:20	10	−2	0.7
W75S(3)	20:39:03.5	42:25:53	39	0	3.0
L1206	22:28:52.1	64:13:43	109	−11	1.2
Cep A	22:56:18.1	62:01:49	1420	−3	0.7
W3(OH)	02:27:03.8	61:52:25	3880	−44	2.2
S231	05:39:12.9	35:45:54	208	−13	2.0
AFGL5142	05:30:45.6	33:47:52	90	2	1.4
S252	06:08:53.7	21:38:30	495	11	2.2
S255	06:12:54.5	17:59:20	72	5	1.4
S269	06:14:37.3	13:49:36	61	15	1.0

Accurate positions were previously obtained with a single baseline MERLIN observation. The Cep A field is considered in this paper used solely for the purpose of astrometric reference. The AIPS task CVEL was used to correct the frequency spectrum to account for movement of the instrument relative to the local standard of rest.

3.2 Wide field search

The search for sources can be either performed in the visibility domain or the image domain. Due to the low number of stations, and the short observation time, the visibility data discussed in this section are reasonably modest (≈20 GB per source). The high resolution and wide field of view result in extremely large image domain maps (≈30 TB per source). In this case, operating in the visibility domain would be advantageous as the volume of data is significantly lower. The visibility domain, fringe rate mapping technique [22] implemented in AIPS as FRMAP exists, but requires sources to be detectable on many individual baselines within a a short period of time to locate the source, and therefore incur far worse detection thresholds. Computational requirements aside, searching in the image domain is a simpler problem. The imaging routines in AIPS offer a high degree of flexibility and when the data is in the image domain, simple algorithms can be used for source detection. As we had access to HPC resources the image domain was determined to be the more suitable domain to work in, providing a simpler problem and saving implementation time. The software described in Section 2 was used to accomplish the wide field search.

3.3 Imageable field of view

To determine the scale of the problem the imageable field of view must be calculated. The field of view is primarily limited by three factors; (1) the physical optics of the antennas involved, (2) the time and, (3) frequency averaging dictated during correlation. An additional issue, the w-term effect originates from the assumption that the array is 2 dimensional and can result in distortions that increase with distance form the phase center. Solutions to this last problem now exist in imaging algorithms [6, 11, 13] with the polyhedral imaging technique been provided in AIPS.

3.3.1 Time & frequency averaging limitation

The imageable area within which the amplitude will have decreased to no less than 90 % due to time and bandwidth smearing is given in arcseconds by the following formulas [3]:

$$FOV_{BW} \leq 49.5'' \frac{1}{B} \frac{N_\nu}{BW_{SB}} \tag{2}$$

$$FOV_{Time} \leq 18.56'' \frac{\lambda}{B} \frac{1}{t_{int}} \tag{3}$$

where B is the baseline length in units of thousands of km (2.5), N_ν is the number of frequencies per subband (1024), BW_{SB} is the subband bandwidth in Megahertz (2), λ is the observing wavelength in centimeters (5) and t_{int} the typical integration time in seconds (0.25). Yielding 90 % amplitude field of view radii of 168′ and 2.5′ respectively. The mismatch between these values is indicative of the fact that this is a spectral line observation. The high spectral resolution was driven by the need to accurately determine the velocities of the maser sources. Had we only been interested in continuum imaging we could have chosen a much lower number of channels (e.g. 64) without consequences for wide field mapping. The nature of this observation (narrow band, spectral line) also allowed the side stepping of AIPSs limitations, e.g. wide band imaging.

3.3.2 The w-term limitation

The (u, v, w) vector associated with a visibility is calculated for a specific direction. When we image wide fields this vector will vary considerably over the field and if not corrected for, will cause a distortion of sources which increases in severity with distance from the field center. Algorithms such as *w-projection* [6] have been developed to correct for this. This algorithm is not available in AIPS however, which handles the problem via polyhedral imaging. The field is divided into many sub-fields, each with recalculated visibility phases and (u, v, w) vectors.

3.3.3 Primary beam limitations

The baselines with the largest dishes (and therefore the narrowest primary beam envelope) in this experiment are these consisting of the Effelsberg 100 m dish coupled with a 32 m class dish (those at Cambridge, Medicina, Noto, Torun). Taking the calculations from [17] and scaling to 6.7 GHz, we get a half-power beamwidth (HPBW) of approximately 2.5 min^{-1} which effectively limits our high sensitivity field of view. Disregarding Effelsberg data extends the HPBW to at least 5.8 min^{-1}, albeit at lower sensitivity. It should be noted that these are the half power limits i.e. 50 % amplitude field of view, while the above limits are to 90 % amplitude levels, meaning that the primary beam limitation is much more severe than the time and frequency averaging effects.

If we limit the area we image to the 100 m–32 m baseline HPBW, we cover an area of approximately half that radius and one can average up to at least 0.5 s of arc without adverse time averages effects.

The most computationally intense portion of the data analysis is the imaging of the data; the transformation of the calibrated interferometric data into sky images is accomplished by first interpolating the data onto a regular grid. This is usually accomplished by means of a convolution. The data is then be transformed to the image domain by means of an FFT.

Based on the inherent resolution of data and allowing for Nyquist sampling, a cellsize of 1.5 mas was used for imaging. As we can decompose the field into sub-fields or facets, we need not image a rectangular area encompassing the imageable portion, rather we will approximate the circular field with small rectangular facets. Using the most limiting half power beam width, that involving Efflesberg and a 32 m antennas, of 2.5 $\min^{-1}$, we image an area of $\pi \times 75^2 \sim$ 17500 s^{-2}. With the cell size of 1.5 mas and ~1000 spectral channels this yields an data cube of order 7×10^{12} cells. As these cells are stored internally as 32 bit floating point numbers the storage required to accommodate this cube is of the order of 30 TB. The gridded UV data produced as an intermediate step in the imaging process requires a similar amount of capacity. It is the processing of these quantities of data that necessitated the use of a distributed computing solution. The AIPSLite system described in Section 2 was used for the imaging and detection process.

3.4 Processing the wide field

At the highest level, the workflow for performing the imaging and analysis follows from the AIPSLite processing model shown in Fig. 2 and is as follows:

- Determine execution environment
- Decompose data based on computing resources present
- Distribute datasets to nodes. On each compute node:
    - Dynamically configure node for AIPS usage
    - Load data
    - Decompose field into optimized facets
    - Sequentially image facets
    - Run detection routines and collect statistics
- Aggregate statistics and store centrally
- Identify sources of emission and remove from dataset
- Re-analyze portions of the dataset affected by the previous step

3.5 Data decomposition

As the target sources in these observations are masers and have a finite frequency (although subject to Doppler broadening) the frequency channels are to first order independent with respect to the imaging process. Furthermore, on the cluster used for this project, the number of compute nodes allocated was 100, therefore the number of spectral channels exceeds the number of computing resources by a factor of 10. These factors allow for convenient data decomposition along the frequency axis, with each compute node dealing with a subset of frequency channels.

It is not possible to image the entire field in one pass. Apart from the fact that the AIPS IMAGR task has a maximum field size of 8192 × 8192 cells, the computational requirements would be prohibitive. Furthermore, the tangent shifting approach to handling the w-term problem, necessitates the use of subfacet imaging. In this approach, the field is divided into sub-fields or 'facets' which are imaged separately. The observations described here are not sensitive to structures large with respect to the facet size. The shortest baseline present is approximately one tenth of the longest. Highly resolved sources are not expected in these observations, but were they to exist, they would be imaged by this faceted imaging approach. Experiments were performed to determine if there was an optimal choice for the dimensions of the facets in terms of cells and frequency channels which would minimize the computational time required. A field of dimensions 8192 cells squared with 16 frequency channels was imaged with various decompositions. From this it was determined that the optimal facet is of size 2048 cells squared. The number of channels imaged in each run is less significant. Channel sizes of 2, 4 and 8 yielded comparable performances, with 4 being marginally more optimal.

AIPS provides the SETFC program to automate the generation of facet parameters. It was found to lack some flexibility required for this project such as configurable geometric layout. Similar functionality to that provided by SETFC but with extra configuration options was developed in Python as part of the analysis.

3.6 Multiple CPU usage

Processing nodes in modern high performance clusters typically consist of single or multiple processing cores. With multi-CPU machines it can be quite a challenge to keep the CPUs busy as they have to compete for I/O. AIPS is in general quite I/O expensive; most tasks do not allow for in-place editing of data, instead a new output dataset is created by the task being executed. If a task's I/O dominates performance then this additional computational capability is of little use. The process of interferometric imaging is computationally intense. Construction of a regular grid suitable for Fourier inversion from interferometer data is accomplished via convolutional gridding. This step, in which each visibility is convolved by a kernel and evaluated at the grid points, dominates the imaging process [5]. The gridded data is then inverted yielding the image data.

An experiment was run on the Joint Institute for VLBI in Europe's ALBUS cluster to determine if multi processing of AIPS imaging can indeed improve performance or whether the system is I/O bound. The ALBUS cluster consists of four nodes, each with four CPUs. The multi-core tests were run on this system to determine how performance scales to four CPUs. It should be noted here that the multi CPU AIPS usage discussed is via separate executions of AIPS tasks via ParselTongue. The tasks themselves are not parallelized, rather separate instances of them are run in parallel acting on different data.

The analysis, summarized in Fig. 3, shows that using multiple CPUs can be quite beneficial, with a speedup of 1.5 when using two processors and 2.5 when using 4 processors. This demonstrates that the AIPS imaging process is computationally bound—while significant I/O is required it is not a bottleneck. The memory used also scales with the number of processes. For the data

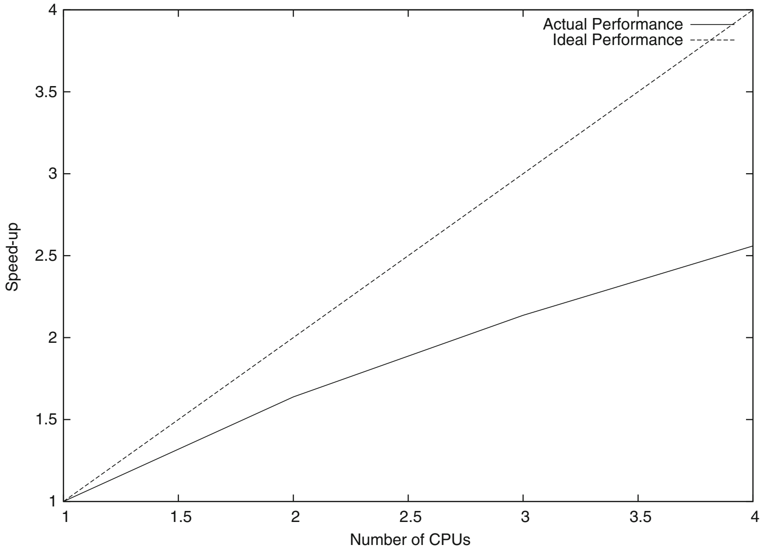

Fig. 3 Actual speedup (*solid line*) and ideal speedup (*dashed line*). The *lines* represent the reciprocal of the execution time normalized by the single CPU execution time. In the ideal case, using *N* processors results in a speedup of *N*. A speedup of >1.5 was achieved on two processors and a factor of >2.5 on 4 processors

described in this paper, AIPS utilized approximately 450 MB of RAM per process.

3.7 Detection mechanism

While imaging such a large field is a challenging problem, so is the detection of sources of emission in the field. Due to the size of the imaged data previously discussed the detection procedure must also be automated. The sparse UV coverage of the observations described causes the point spread function (PSF) or *dirty beam* to be relatively flat (see Fig. 4) and makes analysis of non-deconvolved images difficult, a CLEAN deconvolution was performed during the imaging stage. To collect information form the imaged data, Python routines were developed to extract information from AIPS tables and the output of AIPS tasks. Data about the cleaned flux is extracted from the CC tables.[2] While an AIPS task can return output by setting task variables, some tasks such as IMEAN present some of their information via textual output only. To account for this a parser was implemented to extract desired information from the output of the task. These gathered parameters were collected for subsequent analysis.

[2] AIPS CC tables contain per-pixel flux levels extracted from the image by the deconvolution process, and act as source models within AIPS.

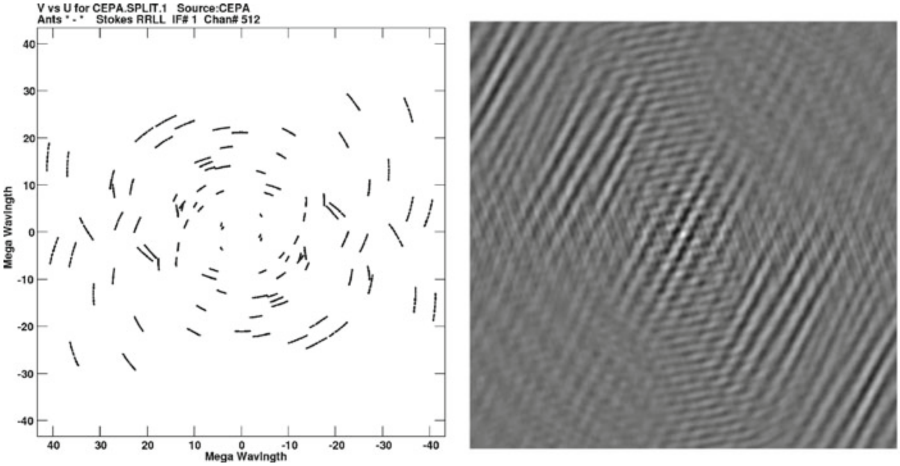

Fig. 4 *Left*: UV-Coverage of the observation of Cep A. Coverage is sparse due to the low number of antennas (eight) and the short observation time of 2 h. *Right*: Resulting PSF. The sparse UV-Coverage results in a flat PSF with significant sidelobes. This makes analysis of non-deconvolved images difficult as the sidelobes add to the noise in the image

The AIPS task IMAGR implements a Cotton–Schwab deconvolution algorithm [16]. This provides superior image fidelity than other algorithms such as the Clark method [4] but is more processor intensive. A Clark CLEAN was also tested for performance. In this mode, IMAGR was used to produce non-deconvolved (‘dirty’) images which were then deconvolved by the Clark method with APCLN. This deconvolution method allows only for one quarter of the image to be deconvolved meaning the images size has to be doubled in both dimensions. This extra imaging load more than offsets any speedup gained. For this reason IMAGRs Cotton-Schwab CLEAN was used.

The information collected by the imaging and analysis software includes: the location and flux of possible areas of emission as well as noise statistics, upper and lower limits in the vicinity of the candidates. These data are recorded for all facets in the field. Post-processing software developed for this project then analyzes these data and identifies possible detections based on the following criteria:

- The source should have a signal to noise of greater than five to avoid an abundance of false positives, given the large search area.
- Sources separated by a distance comparable to the resolution of the instrument are taken to be the same source.
- A source should be present in at least two neighboring channels.

A map is then generated by the software of potential detections for the users inspection. The software can then remove verified sources from the UV dataset and these channels can then be re-analyzed. This is desirable due to the flat beam (PSF) of these observations. If a source is not subtracted from the data, other weaker sources may be hidden by the sidelobes of the strong source.

4 Results

4.1 Performance

For each of the following maser sites Cep A, W3(OH) and AFGL5142 a circular field of radius 1.25 min^{-1} was processed. For each of these objects a job was prepared which and submitted to the Irish Centre for High End Computing (ICHEC) cluster via the PBS system. A resource allocation of 64 dual CPU nodes for 45 h was requested for each job totaling 5760 CPU hours per job. The RAM requirement were relatively low with each process requiring approximately 450 MB. Two process were running per node (as they each have two CPUs) for a total of 900 MB. This is well below the 4 GB available.

The output of these jobs are large data files containing information about regions of flux, including right ascension, declination, frequency and noise statistics. The AIPS logs are also recorded and saved. For a typical run, approximately 700 MB of statistics about emission in the field are produced and 1.5 GB of AIPS logs. In general, the AIPS logs are not of further use and can be deleted although they can be useful for problem diagnosis when and testing.

The emission statistics are then further processed as outlined above to identify masers in the field. The output of this processing is graphical

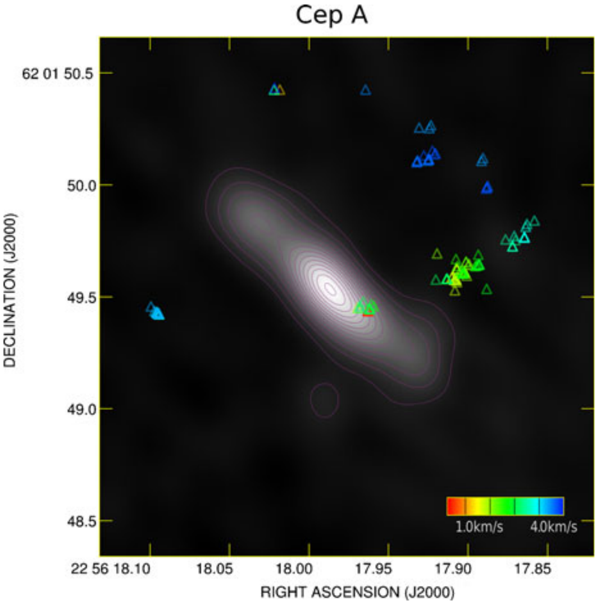

Fig. 5 Initial automatic detections in the Cep A field. The *triangles* show potential maser detections. The observation bandwidth is 2 MHz with 1024 channels giving a velocity resolution of 0.2 km/s. A subset of this range in which detections were made is plotted here. Background continuum image courtesy of S. Curiel

Table 2 Sample of masers detected in the Cep A field

RA (J2000)	Dec (J2000)	Freq (MHz)
22:56:17.965	62:01:49.42	6668.673
22:56:17.910	62:01:49.56	6668.701
22:56:17.905	62:01:49.58	6668.714
22:56:17.962	62:01:49.44	6668.726
22:56:17.916	62:01:49.56	6668.734
22:56:17.867	62:01:49.74	6668.748

The frequency information provides the velocity information in the line of sight allowing the kinematics of the system to be probed

representations of maser candidates, which are inspected and confirmed manually. When sources are confirmed they can be fed back into the system and a followup analysis is performed whereby the confirmed emission is subtracted from the data with the AIPS task UVSUB and the affected channels are reanalyzed. Due to the relatively flat dirty beam, strong sources of emission will leak emission over a wide area and can mask weaker sources. The subtraction and re-analysis stage circumvents this effect.

Running ParselTongue in the normal manner with standard POPS allocation, common configuration and data areas proved unreliable. The process isolation features discussed above were a requirement for stability. When these features were implemented a typical failure rate of less than one process per job was attained. Typically a job would spawn in the order of 10000 processes. Processes that failed were automatically re-run. A situation in which a job failure reoccurred was not observed.

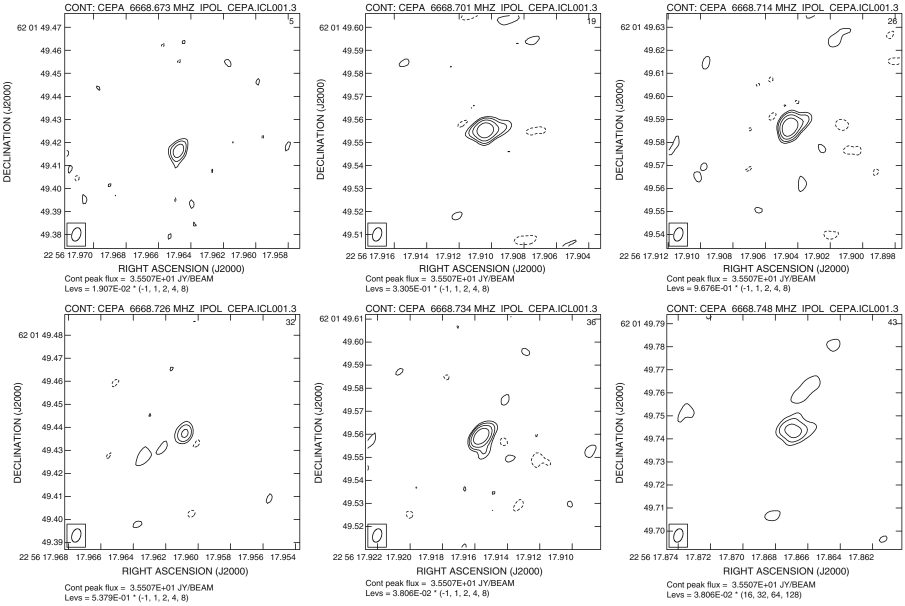

Fig. 6 A selection of the Masers detected in the Cep A field. The automated detections were followed up manually and imaged in AIPS

4.2 Detections

An example of the detections made by the software are shown in Fig. 5. Some of some of the most significant detections are listed in Table 2. As the number of detections was in general low, typically tens, they could be followed up manually. The automatically detected sources were re-imaged and plotted manually in AIPS. The results are shown in Fig. 6. The inner 8 s^{-1} each field was thoroughly imaged manually as it was known to contain sources of emission [1, 12, 21] These portions of the data were used to verify that the automated routines were performing satisfactorily. Our software successfully identified independently all known sources of emission previously reported.

In the analyzed maser sites, outlying sources of emission were not found. While this is disappointing, this result in itself provides useful information on the locality of the star formation regions. The scientific implications of these results are presented in [20].

5 Conclusion

We have developed a lightweight infrastructure, AIPSLite, to allow the deployment of AIPS routines on distributed systems. Using this infrastructure we developed a pipeline in Python with the ParselTongue interface which implements a truly distributed AIPS based analysis of wide field VLBI data. The resulting software has been shown to be highly robust, and is easily deployed on a heterogeneous multi-processor cluster environment running in this case PBS, and breaks the processing bottlenecks which have limited the use of AIPS in this and many other large scale datasets. This pipeline will be used to search the remaining maser sites in this observational campaign for unknown sources. Many masers in the field would be readily detectable without Effelsberg's contribution. In the future, when computation performance dictates, an analysis of a wider field, up to diameter 5.8 $\min^{-1}$, may be desirable. AIPSLite has been incorporated into ParselTongue as of the 2.0 release.

AIPSLite provides methods to set up AIPS on compute nodes with minimal effort providing infrastructure on which to run large AIPS based jobs. The task of data decomposition is left to the programmer as it is highly specific to the task at hand. This approach is useful when a job is highly batch parallizable i.e. the data can be easily be split into smaller chunks which can be handled independently. Interactive use, via either the AIPS TV or via user input is not easily facilitated in this mode of operation. The resources required by the worker nodes are the same as those for a traditional AIPS approach. The main requirement being that enough disk space and RAM be available to allow for the intermediate data products generated must be available to by AIPS for the portion of data to be processed.

Acknowledgements George Heald is thanked for his useful comments on the manuscript. Salvador Curiel is thanked for his continuum image of Cep A. S.B. acknowledges support by Enterprise Ireland, Science Foundation Ireland, and the Higher Education Authority. K.T. acknowledges support by the EU Framework 6 Marie Curie Early Stage Training programme under contract number MEST-CT-2005-19669 "ESTRELA". This effort is supported by the European Community Framework Programme 7, Advanced Radio Astronomy in Europe, grant agreement no.: 227290. ParselTongue was developed in the context of the ALBUS project, which has benefited from research funding from the European Community's sixth Framework Programme under RadioNet R113CT 2003 5058187. The authors wish to acknowledge the SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support.

References

1. Bartkiewicz, A., Szymczak, M., Cohen, R.J., Richards, A.M.S.: Magnetic field in Cepheus A as deduced from OH maser polarimetric observations. Mon. Not. RAS **361**, 623–632 (2005). doi:10.1111/j.1365-2966.2005.09203.x
2. Beswick, R., Garrington, S.T., Gonzalez-Solarez, E., Harrison, P.A., Kettenis, M., Muxlow, T.W.B., Richards, A.M.S., van Langevelde, H.J., Walton, N.A., Winstanley, N.: On-demand radio imaging and access to calibrated data for all astronomers. In: Proceedings of the 8th European VLBI Network Symposium (2006)
3. Campbell, B.: Field of view calculations for the EVN correlator at JIVE. URL: http://www.evlbi.org/user_guide/fov/fov06.html (2006)
4. Clark, B.G.: An efficient implementation of the algorithm 'CLEAN'. Astron. Astrophys. **89**, 377 (1980)
5. Cornwell, T.J.: Ska and Evla computing costs for wide field imaging. Exp. Astron. **17**, 329–343 (2004). doi:10.1007/s10686-005-7449-z
6. Cornwell, T.J., Golap, K., Bhatnagar, S.: W Projection: a new algorithm for wide field imaging with radio synthesis arrays. In: Shopbell, P., Britton, M., Ebert, R. (eds.) Astronomical Data Analysis Software and Systems XIV. Astronomical Society of the Pacific Conference Series, vol. 347, p. 86 (2005)
7. Deller, A.T., Tingay, S.J., , Bailes, M., West, C.: DiFX: a software correlator for very long baseline interferometry using multiprocessor computing environments. Publ. ASP **119**, 318–336 (2007). doi:10.1086/513572
8. Deller, A.T., Tingay, S.J., Brisken, W.: Precision southern hemisphere pulsar VLBI astrometry: techniques and results for PSR J1559-4438. Astrophys. J. **690**, 198–209 (2009). doi:10.1088/0004-637X/690/1/198
9. Guzman, J.C., Humphreys, B.: The Australian SKA Pathfinder (ASKAP) software architecture. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7740 (2010). doi:10.1117/12.856962
10. Kettenis, M., van Langevelde, H.J., Reynolds, C., Cotton, B.: ParselTongue: AIPS Talking Python. In: Gabriel, C., Arviset, C., Ponz, D., Enrique, S. (eds.) Astronomical Data Analysis Software and Systems XV. Astronomical Society of the Pacific Conference Series, vol. 351, p. 497 (2006)
11. Kogan, L., Greisen, E.: Faceted imaging in AIPS. In: AIPS Memo 113. AIPS Memo Series, vol. 113 (2009)
12. Minier, V., Conway, J.E., Booth, R.S.: VLBI observations of 6.7 and 12.2 GHz methanol masers toward high mass star-forming regions. II. Tracing massive protostars. Astron. Astrophys. **369**, 278–290 (2001). doi:10.1051/0004-6361:20010124
13. Perley, R.A.: Wide Field Imaging II: imaging with non-coplanar arrays. In: Perley, R.A., Schwab, F.R., Bridle, A.H. (eds.) Synthesis Imaging in Radio Astronomy. Astronomical Society of the Pacific Conference Series, vol. 6, p. 259 (1989)
14. Roy, J., Gupta, Y., Pen, U.-L., Peterson, J.B., Kudale, S., Kodilkar, J.: A real-time software backend for the GMRT. Exp. Astron. **28**, 25–60 (2010). doi:10.1007/s10686-010-9187-0

15. Schilizzi, R.T., Aldrich, W., Anderson, B., Bos, A., Campbell, R.M., Canaris, J., Cappallo, R., Casse, J.L., Cattani, A., Goodman, J., van Langevelde, H.J., Maccafferri, A., Millenaar, R., Noble, R.G., Olnon, F., Parsley, S.M., Phillips, C., Pogrebenko, S.V., Smythe, D., Szomoru, A., Verkouter, H., Whitney, A.R.: The EVN-MarkIV VLBI data processor. Exp. Astron. **12**, 49–67 (2001)
16. Schwab, F.R.: Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry. Astron. J. **89**, 1076–1081 (1984). doi:10.1086/113605
17. Strom, R.: What is the primary beam response of an interferometer with unequal elements? In: Bachiller, R., Colomer, F., Desmurs, J.-F., de Vicente, P. (eds.) European VLBI Network on New Developments in VLBI Science and Technology, pp. 273–274 (2004)
18. Taylor, G.B., Carilli, C.L., Perley, R.A. (eds.): Synthesis Imaging in Radio Astronomy II. Astronomical Society of the Pacific Conference Series, vol. 180 (1999)
19. Thompson, A.R., Moran, J.M., Swenson, G.W.: Interferometry and Synthesis in Radio Astronomy. Wiley (1986)
20. Torstensson, K.J.E., van Langevelde, H.J., Vlemmings, W.H.T., Bourke, S.: Dynamics of the 6.7 and 12.2 GHz methanol masers around Cepheus A HW2. Astron. Astrophys. **526**, A38 (2011). doi:10.1051/0004-6361/201015583
21. Vlemmings, W.H.T., Diamond, P.J., van Langevelde, H.J., Torrelles, J.M.: The magnetic field in the star-forming region Cepheus A. From H_2O maser polarization observations. Astron. Astrophys. **448**, 597–611 (2006). doi:10.1051/0004-6361:20054275
22. Walker, R.C.: The multiple-point fringe-rate method of mapping spectral-line VLBI sources with application to H2O masers in W3-IRS5 and W3/OH/. Astrophys. J. **86**, 1323–1331 (1981). doi:10.1086/113013