JOINT INSTITUTE FOR VLBI IN EUROPE

Package comparison for BHC pipeline

BlackHoleCam software pipeline WP

Revision	1.0
Author	Ilse van Bemmel
Date	19 December 2014

Scope

The processing of BlackHoleCam (BHC) / Event Horizon Telescope (EHT) observations requires development of a pipeline. The purpose of this pipeline is to automate data processing such that user involvement is minimized. It should produce science quality images and non-imaging data products.
The pipeline and its components are under development at JIVE. This document contains an analysis of available software packages at the onset of the BHC project, and the functionality in the most recent version.
The goal of this document is to compare existing packages, and determine which package will be most suitable to serve as a basis for the pipeline development. In addition there are several existing pipelines, as well as individual scripts that provide parts of the pipeline. These are assessed for portability and suitability for mm-VLBI.

Introduction

The purpose of the BHC pipeline is to process correlated observations from global mm-VLBI telescopes into science quality images. The ultimate science goal is to image the emission from the scales closest to the black hole in our Galaxy and in M87. The main output of the pipeline for BHC scientific purposes is images. It is therefore crucial that the package of choice has imaging capabilities. Since there is useful information in the observations even when the data quality is not sufficient for imaging, the pipeline should also produce non-imaging products such as phase-closure relations.
The pipeline will be optimized for the purpose of BHC science, operating in the 1mm wavelength regime. The components of the pipeline, such as the fringe finder, should be generic enough to enable additional VLBI use cases to develop their own pipeline.

Boundary conditions

This section discusses a range of minimal requirements for the data processing pipeline. We assume that the correlator produces standard FITS-IDI output, or that converter routines exist to generate this. Each processing step that takes place after correlation will be considered an element of the pipeline.
At the moment, the complexity of mm-VLBI data processing requires user interaction in all steps. In addition, user experience is often essential to achieve good quality and reliable data products. A pipeline to capture this process should have the possibility for a user to adjust and test the settings of all required tasks.
With the data quality being variable, and not all data being suitable for imaging, the pipeline should be modular. This enables a user to employ only the parts of the pipeline that are required for a certain result, thus optimizing the processing speed.
To optimize the data-flow through the pipeline, it is desirable to have only one package with a single data format, rather than having to convert data back and forth between packages. This has been shown to be very ineffective (Bourke 2012, reports from ALBiUS project).

Current software packages

In this section the existing radio astronomy processing packages are described in order of their age. For all packages we have analyzed the most recent version.

1. AIPS: Astronomical Imaging Processing System

AIPS is the classical radio astronomy data processing software, which has been in use for decades. It has continued to expand to include modern platforms. AIPS was designed by NRAO to work with the VLA from its first years, and is currently maintained by NRAO and AUI.
AIPS has a Python wrapper script that enables easy scripting of complex tasks: ParselTongue. As ParselTongue relies on the Obit package, installation has long been a bottleneck. This is solved with the availability of Homebrew packages via the JIVE wiki. These packages require active maintenance to ensure access for a broad range of operating systems and platforms.
AIPS is by far the best tested package with full VLBI capabilities. Several pipelines have been developed using ParselTongue, and are currently in operation. However, AIPS is no longer actively developed by NRAO. Though it still has active and rapid user support, it is expected that reliability will decline in the future.
The fringe finder in AIPS is based on the Schwab-Cotton algorithm, which is more sensitive to weak sources, but does not always enable the user to determine a true signal-to-noise ratio of the fringe, since the noise in delay and rate space is not well defined.

2. Miriad

The Miriad package was initiated by BIMA in the late 1980's, in a response to the then existing processing packages. It was aimed to be more flexible and more programmer friendly (see Sault+ 1995, ASPC 77 433).
Like AIPS, Miriad uses tasks with parameters that can be specified by the user. It can read UV-FITS files, but has its own internal file format. There is a small, but very stable user base for Miriad, especially in the HI community, which is actively involved in further development.
Scripting in Miriad requires shell scripts or Python. There is a large and up to date documentation set, including example data reductions, maintained at ATNF.
This package has no fringe finder, and to the best of our knowledge has never been used for processing VLBI observations.

3. HOPS: Haystack Observatory Postprocessing System

This package is used both by a specialized group of mm-VLBI astronomers, and geodetics researchers (Earth Rotation Services, or ERS). It was designed in the mid 1990's to process MkIV correlator data. The internal data format is still based on this. The package has no imaging capabilities, but provides a significant part of the steps required for the pipeline. It is under active development at MIT Haystack and in use both there and at MPIfR Bonn. It is currently the data reduction package of choice for Event Horizon Telescope observations.
HOPS is used for ERS data processing, where it is a critical component of the data analysis. This implies that development of this package requires close contact with the ERS to ensure the continuity of their use cases. There is a risk that their requirements are not compatible with BHC requirements.
HOPS is a command line package, which can be scripted using any shell command language. In principle Python can call the tasks through the Python.os interface, but this has not been tested.

The fringe finder in HOPS is based on the Alef-Porcas method, with added incoherent capabilities as described by Rogers, Doeleman & Moran (1995). It requires an iterative approach and hands-on experience to achieve a good fit.
Future development for HOPS is mainly ensured through the use by geodetic research. Additional development comes from users who contribute their own scripts to a larger database. There is no dedicated user support, though the team in Haystack is very approachable and helpful.

4. Difmap

About the same age as HOPS, Difmap provides a fast and flexible editing and mapping programme. It cannot perform any calibration steps, but does self-calibration in a highly optimized manner. It was developed by CalTech. Since Difmap only does the imaging step, it will not be considered as an option for the full pipeline.

5. CASA: Common Astronomy Software Applications

The CASA package is being developed for the purpose of supporting the post-processing needs of next generation radio telescopes, such as the JVLA, VLBA and ALMA. Under the guidance of NRAO the consortium also includes ESO, NOAJ, CSIRO/ATNF and ASTRON. CASA applications for ALMA are actively being developed, and the prospect of continuing this development on long term is good. There are no dedicated VLBI tasks within CASA, most notably there is no fringe finding algorithm and no dedicated imaging task for observations with sparse uv-coverage. A detailed analysis of the requirements to make CASA VLBI-ready has been written up by NRAO and MIT authors (REF).
Future development of CASA at NRAO and within the ALMA consortium does not include any VLBI related tasks, but the team welcomes contributions from other institutes as part of the CASA package.
The user base of CASA is growing, and there is a platform to report bugs and obtain support. CASA has a Python interface, making it easy for users to develop and exchange scripts. There is an extensive database with examples for data reduction.

6. LOFAR software suite

The LOFAR software is the only package that has an innate pipeline framework where the user is entirely removed from the data processing. The framework consists of a collection of tasks with predefined settings. The LOFAR definition of a pipeline is a series of such tasks, called in a certain order. This results in a highly reliable pipeline from a software point of view, but it lacks the flexibility required for mm-VLBI data processing. The pipeline framework is still under development, but expected to be finished within the project timeline for BHC.
The calibration algorithms are highly optimized for speed using the StefCal algorithm (Salvini & Wijnholds, REF), which makes this a very time efficient package. The software has no fringe finder, but this is under development for the calibration of the international baselines. The package is relatively young, and requires dedicated software developers for implementing the VLBI data processing in the pipeline framework. A Cookbook with examples exists, and there is dedicated user support and issue tracking.

7. PIMA

This is a comprehensive, VLBI specific package for data processing. Like HOPS, it does not have imaging capabilities. It focuses strongly on batch processing of astrometry and geodetics observations. The user base of PIMA is small in astronomy, and the package lacks a proper documentation and example infrastructure. Installation requires assistance from the developers, which is not desirable for a package aimed at a global user base.

Comparison basis

In this section we compare the above packages on their current abilities to serve as a fully capable VLBI data processing package. The presence of a fringe finder is not considered crucial, as development of this component will be part of the ongoing work in JIVE. An imaging task is considered a key element, since developing new imaging routines is not part of the work-package. The sparse uv-coverage may necessitate alternative imaging methods, as exploited in optical interferometry.
Listed below are five general aspects which are considered, each with several requirements. To each of these requirements a weight is assigned. The weight definition is: 1=optional, 2=desired, and 3=required. The scoring is done on a scale from 1 to 5, in which 1 is the lowest (poorest quality or non-existing), and 5 is the best possible.
The requirements are scored per package. A weighted mean is calculated for each aspect, and for the total. We also calculate a standard deviation of the scores. This number is an indication of the risks involved. A high standard deviation implies that some requirements are well met, but others require work to be done. A low value indicates that all requirements are met to a similar standard. The package of choice therefore has a high weighted mean, combined with a low standard deviation.

The list of requirements, including their weight in brackets:

1. Reliability
   - 1.1. Existence of coding standards and version control (2)
   - 1.2. Software tests exist to test installation (1)
   - 1.3. Size of the user community to find issues quickly (2)
   - 1.4. Presence of user support to solve issues quickly (3)
2. Flexibility
   - 2.1. Maintenance possible by central IT facility in an institute(2)
   - 2.2. Implementation additional VLBI use cases is possible(2)
   - 2.3. Data format can accommodate VLBI data (1)
   - 2.4. Routines to import data from FITS-IDI exist and include meta-data (3)
3. Future
   - 3.1. Package will be actively developed in future (3)
   - 3.2. Compatibility with multiple OS versions (2)
   - 3.3. Upgrades and releases with improved capability are expected (2)
4. User access
   - 4.1. Learning curve is not too steep (2)
   - 4.2. Graphical User Interface is present and of good quality (1)
   - 4.3. User documentation is present and up to date (3)
   - 4.4. Example reductions exist and are up to date (1)
   - 4.5. Package maintenance and installation by user is easy (3)
5. Pipeline readiness
   - 5.1. Package can accommodate the full pipeline (3)
   - 5.2. Framework to build pipeline exists (3)

The list reflects the needs for a package which is highly future proof and easily accessible to a large variety of users. The grades reflect the *current* state of the package, except for pipeline readiness, which indicates the *potential* to accommodate the full pipeline in this package. This includes a rough assessment of the likelihood that missing components can and will be added, and the amount of work involved in that.
For AIPS requirements 5.1 and 5.2 depend on the presence of ParselTongue and Obit. These are independent packages, adding to the complexity when software needs to be maintained or upgraded. Though AIPS and ParselTongue are at the basis of many current pipelines, preference is given to a single package.

With ‘framework’ we refer to the presence of a uniform scripting language that is used within the package. It enables users to develop their own scripts and exchange them. The LOFAR software has a genuine pipeline framework, which adds an additional level of abstraction, but limits user interaction.

The scores are listed below in Table 1. It is evident that CASA scores best overall for implementing the VLBI pipeline, having both the highest mean and the lowest deviation. This implies the package meets most of the requirements, and has the least amount of work to be done. The package has excellent future prospects thanks to JVLA and ALMA, and the existence of a large user base, including all required documentation and support. Another very strong point of CASA is the exchange of user scripts in a uniform language (Python). CASA does require the development of VLBI related tasks, but this also is an opportunity to develop an optimized fringe finder algorithm for mm-VLBI applications.

As expected, AIPS is currently the most flexible and reliable VLBI package, but it lacks commitment for future development. Dedicated VLBI processing software such as HOPS and PIMA score lower, due to the limited user access and reliability. Unfortunately, this is not expected to improve in the near future.

An argument could be made that instead of developing a fringe finder for CASA, one could develop an imaging task in HOPS or PIMA. Though this is certainly possible, and may even be a task of comparable complexity and workload, these packages would still fall short in the additional requirements we have considered here.

Conclusions

This document aims to answer the question which package would be most suitable to host a pipeline for the BHC project, but with additional focus on future generic VLBI capabilities. We have performed a requirements analysis for a range of existing packages used in radio astronomy. We assessed the reliability, flexibility, future prospects, user access and the existing options to develop and implement a software pipeline. CASA is the best suitable package for pipeline development in BHC, and for making VLBI more accessible for other users.

Future work

Several aspects of the VLBI data processing are currently not present in CASA. The next step in the project will be to analyze the missing tasks, and start a design process to develop algorithms and tasks to implement in the CASA package. The purpose is to make these tasks fully native in CASA, and flexible enough to accommodate additional use cases for VLBI observations in CASA.

References

TBA

Components	AIPS	Miriad	HOPS	CASA	LOFAR	PIMA
1. Reliability	**4.5**	**2.0**	**2.6**	**4.3**	**3.6**	**1.4**
1.1 coding standards	4	2	2	4	4	2
1.2 software tests	3	2	2	3	4	2
1.3 user community size	5	2	3	4	4	1
1.4 user support	5	2	3	5	3	1
2. Flexibility	**5.0**	**1.3**	**3.3**	**3.7**	**4.0**	**4.7**
2.1 maintenance	5	3	3	5	5	3
2.2 upgrade	5	1	3	3	4	5
2.3 data format	5	1	5	3	3	5
2.4 conversion	5	1	3	4	4	5
3. Future	**2.1**	**2.4**	**2.4**	**5.0**	**3.0**	**3.0**
3.1 active development	1	3	3	5	5	5
3.2 compatibility	4	1	1	5	1	1
3.3 upgradeability	2	3	3	5	2	2
4. User access	**4.0**	**3.2**	**1.5**	**4.6**	**2.0**	**1.0**
4.1 learning curve	4	2	2	4	2	1
4.2 GUI	2	3	1	3	1	1
4.3 documentation	5	3	1	5	3	1
4.4 example reductions	3	4	1	5	3	1
4.5 installation by user	4	4	2	5	1	1
5. Pipeline readiness	**4.0**	**2.5**	**2.0**	**5.0**	**4.0**	**2.0**
5.1 full pipeline	4	2	1	5	5	1
5.2 framework	4	3	3	5	3	3
Total score	**3.9**	**2.3**	**2.3**	**4.5**	**3.2**	**2.3**
Standard deviation	1.2	1.0	1.1	0.8	1.3	1.6

Table 1. Scoring overview of the packages. For AIPS a point was subtracted for requirements 5.1 and 5.2, since they rely on the presence of ParselTongue and Obit, which are separate packages.