FP7- Grant Agreement no. 283393 – *RadioNet3*

Project name: Advanced Radio Astronomy in Europe

Funding scheme: Combination of CP & CSA

Start date: 01 January 2012 Duration: 48 month

Deliverable D10.1

Activity Plan,

architecture and selection of benchmarks

Due date of deliverable: 2012-12

Actual submission date: 2013-05-02

Deliverable Leading Partner: STICHTING ASTRONOMISCH ONDERZOEK IN NEDERLAND (ASTRON), The Netherlands

An European project supported within the 7th framework programme

SEVENTH FRAMEWORK PROGRAMME

1 Document information

Type	Report
WP	10 (Hilado)
Authors	Marco de Vos (ASTRON) & Hilado Team

1.1 Dissemination Level

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

1.2 Content

2 Introduction

2.1 Purpose of this document

The *RadioNet3* JRA Hilado consists of four workstreams that focus on specific targets, but are interrelated in several ways. This document describes the activities within each workstream and their relations. It is intended to guide the activities in the JRA and to lead to a timely release of the other deliverables. Presenting and publishing progress and results will take place at regular conferences and through the Deliverables; this is not described in any detail here.

2.2 Top-level goals

The scientific and technical goal for Hilado is to create optimized prototype software and demonstrator processing pipelines that improve the capabilities of currently planned software packages for existing and emerging radio telescopes. These developments are essential to increase the potential of the *RadioNet* user community in opening up those facilities for the more demanding scientific applications.

Hilado also explores a new development model for astronomical processing software. Expertise from a variety of disciplines needs to be tapped: from mathematics for the foundations of new algorithms, from computer science for optimizations for high-performance computer platforms and from industry to explore the space of novel architectures. Talented people around the world should be enabled and encouraged to contribute. This collaborative project will facilitate the adoption of the new developments, technologies and insights required by huge datasets, and will tap expertise beyond the boundaries of the radio astronomy community. The success of this project depends critically on placing existing as well as novel HPC technologies at the centre of the development of algorithms, software and procedures. The JRA aims at optimized prototype software that can be re-used in a variety of contexts, including both automated pipelines and user-adaptable scripts. This brings maximum value to a broad group of *RadioNet* users by allowing them to balance between highly optimized automated processing and highly flexible interactive application without sacrificing efficiency and performance.

3 Activity plan

3.1 Optimisation of Casacore

Casacore is a C++ library with essential functions for the CASA data reduction package. These functions range from storage managers to solvers, from basic data-types to complex astronomical measures. Although Casacore is thus an essential and necessary component of CASA based data-reduction, as well as for proprietary packages used for e.g. LOFAR and ASKAP processing. Currently the library is maintained in an informal consortium, which severely limited further optimisation. Through Hilado, it becomes possible to make substantial improvements “under the hood”, leading to significant efficiency improvements and making processing of very large datasets possible.

The Hilado plan identified an inventory phase early in the first year of the project, followed by prototype development and verification. In practice, it turned out to be more efficient to three (roughly yearly) cycles of inventory (goal-setting), development and verification.

In the first year the Casacore and CASA system have been improved in several ways. An important step has been the creation of the ASDM storage manager to avoid having to copy the bulk data when importing data in ASDM format into the CASA Table format. The storage manager is quite general and can probably be applied to data stored in UVFITS or FITSIDI format as well. The program converting a FITS file to a MeasurementSet can be changed such that on demand it creates the index file instead of copying the data.

The Measures are an important part of the Casacore package and are expected to be used in SKA software as well. They are thread-safe, but some more optimisation can be done to make them perform better when used in a multi-threaded program.

For the next cycle, two main areas have been identified, relating to datasets and solvers. A third area will relate to the maintenance and release process.

Casacore has the capability of virtually concatenating MeasurementSets, which is exploited in CASA by means of the so-called MMS (multi MS) format. It makes it easily possible to process the MMS parts in parallel, but on the NRAO cluster using the Lustre file systems and on ESO machines using SSDs the gain in performance is less than expected. It has to be investigated where the bottleneck is. Cambridge has quite some expertise in a large cluster with Lustre and will assist in the investigation.

Now StEFCal has established itself well (see 3.3), it can be added to the scimath package of Casacore. This can be done as after having shown it works well in the NDPPP part of the Fast Transient Imaging Pipeline.

On 25-26 April the key developers and users of Casacore (from ASTRON, ESO, NRAO, and ATNF) will meet in Dwingeloo and discuss how Casacore can be improved in terms of maintenance, releasing, functionality and performance. The outcomes of the discussions are expected to drive the developments in Casacore, both within Hilado and by other parties. Current topics and ideas are:

- improved documentation and test coverage
- using STL/Boost classes instead of the home-grown, pre-STL classes like Map, List, and Regex.
- adapt to C++11 (e.g., using the new move semantics can improve performance)
- add StEFcal
- prepare for big data; where applicable use 8-byte instead of 4-byte integers
- keep adapting to new compiler technology (e.g., clang) and keep the code free of warnings.
- The Lattice Expression Language (LEL) is a means to perform expressions on images, for example to add images, find median, etc. Images will increasingly be stored in a distributed way and it might be worthwhile to make it support distributed processing in a Map-Reduce way of processing. Furthermore, on a single node it can be parallelised.

Work in these areas will take Hilado well into its third year. Early in the third year, a final inventory of improvement areas will be made.

3.2 Fast Transient Imager

With the advent of ever-larger data-streams, it becomes unavoidable to move data processing functions that have been traditionally carried out interactively into the domain of streaming processing. Through the development of a Fast Transient Imaging Pipeline (FTIP), Hilado will both build a specific application (boosting LOFAR Transient Processing) and develop insights and components to ease further development in this area.

The FTIP procedure will consist of four parts.

1. Sending the data from the real-time LOFAR control system to the pipeline task processing the data.
2. Run the demixing step of NDPPP on the data to subtract the strong A-team sources from the data while solving for their gains.
3. Do a calibration of the data every 10 seconds.
4. Make an image of the data while applying direction-dependent effects like the beam.

The first step taken in developing the FTIP is writing the requirements and global design document. This work is well underway and the corresponding deliverable will be released in the summer of 2013.

In the coming two years the following steps will be taken to do the processing in an efficient way.

1. Sending the data will be done using the UDP protocol to achieve that the real-time system is not hampered by the FTIP. We expect to use the Pelican framework developed in Oxford for this purpose. Also, a mechanism will be developed to send the data from an existing MeasurementSet, which can be used to test the system. It could also be used to find transients in existing observations.

2. Demixing can take quite some time due to the complexity of the source models and the many parameters to solve for. This will be addressed by using StEFCal developed in WP3 as the solver. Furthermore, it will be checked for each time step, which A-team sources need to be subtracted, because some can be too low at the horizon resulting in a too small apparent flux.

3. The calibration of the data will also make use of StEFCal to make it perform sufficiently well. Furthermore, it has to be parallelised.

4. The existing awimager program will be used to make the images. It has to be enhanced in two ways:

    a. Currently, it makes a single image per run. It must be made possible to make an image per second of data, without having to pay the overhead of calculating the convolution functions too often.

    b. The data of multiple subbands needs to be combined. This is being developed at the moment, but it has to be tested if it is fast enough.

3.3 Large Solvers

In this work stream, we will study the mathematical foundations for large scalable solvers, seeking for new methods to apply multi-level parallelism, and use of accelerators such as GPUs for optimizing e.g. convolutional gridding schemes and faceting.

After the submission of the Hilado proposal, it became quite clear that the Statistically Efficient Calibration algorithm (StEFCal) was to be the main candidate for consideration in this work stream. Activities will thus consist of two main components:

- Work on the existing suggested algorithm
- Preliminary studies on other areas of potential impact.

The StEFCal algorithm has been suggested and a scientific paper outlining it, its application to non-polarized data and its characteristics is currently being completed (S. Salvini, S. Wijnholds, 2013).

A preliminary implementation in C++ has been written and an implementation in CUDA, to allow for effective parallel processing on GPUs, is now underway at Oxford University.

This will be followed in due course by publication and application of the polarization algorithm, whose development is now approaching completion.

From the point of view of using StEFCal (both the unpolarized and the polarised versions) within existing applications and systems, a number of collaborations have been established:

- CASA / CASACORE (G. Van Diepen, ASTRON)
- BBS (G. Van Diepen, ASTRON)
- AARTFAAC (S. Wijnholds et al.)
- MeqTrees + MeerKAT (O. Smirnov et al.)

Future development will include at this stage:

- Inclusion of Direction Dependent effects
- Frequency dependency, for example smoothing over frequencies by polynomial interpolation/approximation over frequencies.

Future efforts within this workstream will aim at

- Imaging and related algorithms
- Generalised Selfcal
- Inclusion of all algorithms developed within standard RA frameworks and tools

3.4 Bringing it to the User

Some of the developments in the other work packages will become available to users quite transparently (e.g. the improvements in Casacore by getting compiled into CASA applications). In many ways though, new algorithms and insights cannot be brought to the user that easily. The script based user-development scheme explored in the past decades is insufficiently suitable for e.g. cluster-based processing. Therefore this workstream will work on a high-level Domain Specific Language (DSL) for data reduction in radio astronomy that can very efficiently bridge the gap between batch/script and interactive/exploratory processing.

In the first instance this language will automatically track dependencies between operations on the data, cache intermediate results so that data reduction is not unnecessarily executed multiple times and allow for multiple branches in the data reduction. It will also be able to interact with cluster job control systems Portable Batch System (PBS). Such a language would make data reduction much more efficient in terms of computational resources and the (very valuable) user time and also reduce the possibility of errors in data reduction which can lead to serious error in scientific interpretation of the data.

In the first instance this language will be targeted at the VLBI users. The initial `back-end', that is the computational components that the language will use in the first instance, will be AIPS/ParselTongue but with a clear path to adding CASA tasks when the user cases demand it.

This will leverage the work already done in Cambridge for local telescopes and some initial investigations for ALMA. It would also build on top of fast developing fields of functional programming, dataflow languages and the computational reasoning about programs.

Although the initial target users are for exploratory data analysis that is essentially always required for VLBI, it is expected that the approach will be useful for at least the more complex (& interesting) ALMA observations. Additionally the ideas developed during the project are likely to find relevance in current project like LOFAR and the forthcoming SKA.

As a first step, JIVE will collect user scripts and analyse them to try to discover common patterns. This analysis will be taken into account in the design of the Domain Specific Language (DSL) that will be developed in this work stream. ParselTongue will be adapted to serve as the first backend for the Domain Specific Language. Focus will be on a specific use case of processing wide-field VLBI data correlated with multiple phase centres. These correlations result in multiple data sets (one for each target within in the field). This should offer some good opportunities to explore data parallelism using HPC facilities.

We will then proceed to investigate the use of the DSL in observatory-based computing by developing a web frontend that allows users to submit scripts in that language to be processed on computing resources with direct access to the EVN data archive. Using a DSL for this purpose will address some of the security concerns that have been raised with direct execution of user-provided Python scripts.

4 Architecture

The detailed work plan as described above does not require a deviation from the architecture originally proposed. Hilado will not aim at the development of an independent package, but will instead contribute to existing libraries (in particular Casacore) and develop modules in existing packages (CASA, MeqTrees, LOFAR OLAP, ParselTongue) where possible. If new software is developed that does not fit naturally in an existing library or package, it will be made available as a new library or light-weight package. Software thus will be mainly developed in publicly available Open Source repositories like the Casacore library and the Python scripting environment. This approach gives good opportunities for dissemination and training.

5 Benchmark Platforms

Several benchmark platforms are available to the Hilado team, in particular:

- A variety of multi-core deskside machines (incl. multi-core and GPU-loaded)
- The LOFAR CEP2 cluster consisting of two Frontend nodes (each 64GB, 16 Cores, 512GB SSD, 5.4TB disk), 100 processing/storage (each 64GB, 24 Cores, 2TB disk) and two “Hoover” nodes (each 64GB, 24 Cores, 2TB disk), all 40Gbps Mellanox InfiniBand interconnect.
- The JIVE HPC cluster (40 node, 384 core CPU cluster which has no job scheduling system)
- The University of Cambridge "Darwin" facility, which is a 600-node 16-way SandyBridge cluster connected with Infiniband FDR 56 GB/s network and backed with a very high performance Lustre store. The HPC cluster is controlled using the Torque variant of PBS.