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 D10.2

Report on optimisation studies, indicating resulting improvement and guidelines for prototyping and benchmarking

Due date of deliverable: 2013-06-30

Actual submission date: 2013-06-25

Deliverable Leading Partner: EUROPEAN SOUTHERN OBSERVATORY – ESO
EUROPEAN ORGANISATION FOR ASTRONOMICAL RESEARCH IN THE SOUTHERN HEMISPHERE (ESO)

An European project supported within the 7th framework programme

SEVENTH FRAMEWORK PROGRAMME

1 Document information

Type	Report
WP	10
Authors	Dirk Petry (ESO) & Ger van Diepen (ASTRON)

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 goal of workstream 1 in JRA Hilado is to optimise existing core libraries, in particular CASACore, in order to increase the efficiency of processing of large interferometric radio-astronomical datasets, and to enable new science to be done that is now prohibited by the performance limitations of existing calibration and analysis packages. The purpose of this document is to set out the areas where such optimisations have most potential impact and to define an implementation plan. It will serve as a reference and guideline for the remainder of the workstream.

2.2 Background

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 cycle, access to native ALMA and LOFAR dataset was implemented (see D10.3). The analysis for the second cycle was completed in a meeting on April 25-26th, 2013. Present at this meeting were Malte Marquarding (CSIRO), Jim Jacobs (NRAO), Dirk Petry (ESO), Wim Brouw (ASTRON), Ger van Diepen (ASTRON), Arno Schoenmakers (ASTRON). Final plans for the third cycle will be made mid-2014.

2.3 Overview

This document is ordered along the various areas where optimisations will be made. Since these vary from high-level conceptual to in-depth technical issues, the language varies somewhat throughout the document.

3 Access to native data-formats

The basic data-format in Casacore and the CASA system is the Measurement Set, which sets out both the data-organisation and meta-data. Most radio telescopes generate some native data-format that can be converted to Measurement Sets. This operation, however, is data and compute intensive. Major improvements can be achieved by giving direct access to the native data-formats.

This has been implemented for ALMA through 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 Measurement Set can be changed such that on demand it creates the index file instead of copying the data.

Support for really big images requires that the data are stored in different resolutions to have faster access in e.g. the viewer when showing the entire image. This requires an extra ‘resolution’ argument in functions accessing the data.

For big images, the current tiling approach is not sufficiently efficient. The approach will be optimised for large image datasets.

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.

4 Measures classes/tables

The Measures are an important part of the Casacore package and are expected to be used in SKA software as well. They contain definitions and conversions for a wide range of coordinates and units.

Some of these units require data tables (with e.g. Observatory coordinates and leap seconds). Currently these have to be maintained locally at each site. It is proposed to have a data directory in the code. It will contain a measures directory containing ephemerides and geodetic directories. Requests for additions will be handled through the Issue Tracker system.

Currently the Measures classes are thread-safe. However, more optimisation has to be done to make them perform better when used in a multi-threaded program.

5 Parallelization

Several basic Casacore algorithms can be parallelized relatively easily, e.g. the Sort and GenSortIndirect classes.

Some code in the scimath module will be extended (or changed) with code from packages like GSL.

Parallelisation of higher-level algorithms will take place at CASA system level. Now that StEFCal has established itself well, 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.

6 Standard classes

Standard classes have to be used as much as possible to increase robustness and performance. This requires several major changes:

- replace Map, HashMap, List, Link by their STL-counterparts.
- look if OrderedMap and SimpleOrderedMap can be replaced too.
- remove cregex and base Regex on std::regex.
- replace PtrHolder by unique_ptr (note: auto_ptr is deprecated).
- always use shared_ptr in CountedPtr.
- use std::sort in GenSort.

- look into Jenkins master and server.

Some of these STL classes are only available for newer compilers, so where needed (and if possible) the code has to be #ifdef-ed on C++11.

Some code can be adapted to new C++11 features. However, that should only be done as a background job since its priority is quite low. Using the && (move) semantics in e.g. Array operators +, *, etc should be supported as it speeds up Array expressions because fewer temporaries are needed. Template pre-instantiation for specific types can speed up the build and reduce library sizes.

Currently all statics in Casacore are thread-safe. CountedPtr is thread-safe once it uses shared_ptr.

Some classes like Table and some Measures classes use their own reference counting. This should be replaced by CountedPtr, but a problem is that they have the feature to not increment the count to avoid leaks when mutual references are used.

Classes themselves are not thread-safe and it is not intended to make them thread-safe because the lock/unlock penalty can be too high. It means that when using Casacore objects in different threads, care should be taken that different objects are used if their state can be changed.

7 Casacore Repositories and Releases

In order to boost efficiency in further development, a single code repository is required. It has been decided that the googlecode repository should be used. In principe github could be used, but this will require too much changes. .

The current state of the nrao-nov12 branch of the googlecode repository is that all NRAO changes till 22-Apr-2013 have been merged in and that ImageAnalysis code has been removed. It is the question if file AntennaResponses should be in casacore. All warnings by gcc, clang, and doxygen have already been fixed. The multi-beam support in the Image classes (which is different from the CASA implementation) has been added.

A Steering Committee will be setup to coordinate the central repository and will meet quarterly through skype.

The following steps need to be taken to unify the repositories.

- An svn external has to be created in the CASA repository to link to the casacore repository.
- NRAO has to use the standard Casacore CMakeLists.txt files. In particular, their way to build a release has to be changed because it is very different from normal builds.
- NRAO has to make the change at a good point in time. It is best to do it right after the upcoming CASA-4.1 release. A detailed plan has to be made to ensure that changes in people's workspace are committed in time or copied over.
- ASTRON has to merge the latest NRAO changes into the nrao-nov12 branch. When fully done, the branch has to be merged into the trunk.
- Before changing to googlecode, NRAO should build CASA against the googlecode nrao-nov12 branch to see if the CASA code builds and runs well.
- After this NRAO will tag and release casacore.

Googlecode has its own Issues mechanism for bugs and enhancement request. NRAO uses JIRA. It must be possible to use both mechanisms. Commit messages should contain the issue or JIRA number unless the change is very small.

Some rules and guidelines have to be obeyed when changing Casacore code. The old AIPS++ guidelines will be used as a starting point. Furthermore it is important that test programs run well with valgrind and that sufficient test coverage is achieved. The build system already supports tests with valgrind. ASTRON will look into adding support for test coverage.

Each module in the Casacore package will be assigned a coordinator who understands the code, coordinates changes, maintains the code, makes sure it builds and tests well (also with valgrind), and has sufficient test coverage.

In principle each site can make its a release, but it is expected that the NRAO release schedule will be leading and used by the other partners. In principle the Casacore release version does not need to be the same as the CASA version.