CASA Parallelization Meeting

7 August 2014

Sandra Castro

Current Status

- Multi-MS processing

The underlying code of partition and mstransform has been re-factored in the past few months. A new class has been created in *gcwrap/parallel/parallel_data_helper.py,* which is the helper class for tasks that need to handle Multi-MS (MMS).

There is still work to do on the heuristics side of the MMS cases in mstransform. Mstransform is the general task that can do everything the other tasks do so it is more complex and only to be used by advanced users. Simplified versions of mstransform are in partition, split2 and hanningsmooth2. It is also foreseen that we create a cvel2 to replace cvel.

- Automatic Input/Output relation in mstransform

```
input MS  --  output MS
input MMS --  output MMS
```

The user can set the parameter *createmms=True* to have:

```
input MS  --  output MMS
```

- Heuristics of transformations and MMS in mstransform:

Input MMS case

MStransform will process an input MMS in parallel whenever possible. Each SubMS of the MMS will be processed in a separate engine and the results will be post-processed at the end to create an output MMS.

Naturally, some transformations available in mstransform require more care when the user first partition the MS. If one wants to do a combination of spws by setting *combinespws=True* in mstransform, the input MMS needs to contain all the selected spws in each of the SubMSs or the processing will fail. For this, one may set the initial *separationaxis* to scan or use the default auto with a proper *numsubms* set so that each SubMS in the MMS is self-contained with all the necessary spws for the combination.

The task will check if the subMSs contain all the selected spws when *combinespws=True* and if not, it will issue a warning and process the input MMS as a monolithic MS. In this case, the separation axis of the output MMS needs to

be different than that of the input and will be re-calculated by the task. **This is still under development/testing in mstransform!!!**

A similar case happens when the separation axis of the input MMS is per scan and the user asks to do time averaging with time spanning across scans. If the individual SubMSs are not self-contained of the necessary scans, the spanning will not be possible.

It is important that the user sets the separation axis correctly when first partitioning the MS, therefore we will need to educate them with proper documentation, workshops, CASA cookbook. I will put all this in a document as soon as the last work in the heuristics is finished.

Output MMS case

In a similar way to the input MMS, some transformations require that either selected spws or scans are contained in each SubMS of the MMS in order to work. The combination of some transformations and *createmms=True* is not possible in some cases, depending on the choice of *separationaxis* and *numsubms*.

Because of all these constraints it will not be possible to have the initial MS partitioned in a very granular way as many people think. One needs to group together in a SubMS, all the necessary data that needs to be processed in each engine. One engine cannot rely on data of another engine. Unless we also use message passing here.

Known work remaining

- Heuristics of input/output MMS and transformations such as the combination of spws and time averaging across scans.
- Some sub-tables still need to be reviewed and consolidated for the MMS output creation.
- Discuss an idea to introduce a time *separationaxis* for the MMS creation in partition. Would this be useful?
- Improve error handling for MMS processing. Deal with NULL MS selections, failed SubMSs, etc.

Testing

- We need to test on real ALMA and EVLA cases using full data reduction scripts to reveal potential bugs.
- We need automated tests running on MMSs in Jenkins. Currently there is only test_mstransform_mms, which needs to be extended.
- We need science user testing.
- We need heuristics tests specifically trying the different transformations with input and output MMS. I am creating separate tests (more unitary) for parallel_data_helper.

Task list

- For 4.3
    - Finish the heuristics work.
    - Fix the sub-tables consolidation.
    - Improve the error handling.
    - Create more tests.
    - Initial documentation.
- For 4.4
    - Finish the documentation.
    - Fix bugs on code based on first user testing.
    - Prepare parallel workshop.

Regarding our visit to NRAO this fall. I will be available only in November and start of December, not in October.