Estimate of the Number of CPU Nodes to Execute the Vircam Pipeline in Paranal

Sandra Castro, Stefano Zampieri - ESO/DFS

July 28, 2006

1 Introduction

References used to write this document:

- VISTA Infra Red Camera DFS System Impact 1.1 from 2005-02-08.
- DFS Vircam pipeline interim version 0.3.0.

This document shows the steps taken to estimate the number of processors necessary to execute the Vircam pipeline in Paranal. Our tests were performed based on the current available pipeline recipes and data. The hardware used to execute the recipes is a PC running Scientific Linux with the following configuration:

```
2 processors
model name :  AMD Opteron(tm) Processor 250
cpu MHz :  2389.414
cache size :  1024 KB
MemTotal:  4090124 kB
SwapTotal:  4192824 kB
```

2 Time and CPU Estimates

Our tests were performed on 3 computers with a total of 6 processors such as the ones described above. The data used for tests are Vircam simulated data. The recipes read the Multi Extension FITS (MEF) files, but do the processing on individual extensions, which are passed to the recipes as input parameters. We know from the VISTA documentation that the following numbers are expected to be observed:

Data:

- Average of 200 Gb of science data per night (about 14 hours long).
- Peak of 1.2 Tb per night.
- Each raw MEF file is 0.27 Gb of size.
- Average night: 200 Gb/0.27 Gb = 740 images per night.

Processing time. The pipeline takes about:

- 120 seconds to reduce 12 science images, which is 10 seconds per image extension.
- 10x16=160 seconds to process 1 full MEF image (16 extensions).

3 Conclusion

In an average night, a typical exposure time is 30 seconds, being able to reach 10 seconds in a peak night. Our requirement is to reduce an image as fast as it is observed. If the pipeline takes 160 seconds to reduce 1 single exposure, then we need 160/30 = 5.3 processors to reduce this image before the next one starts to be observed. Further consideration needs to be taken in order to have a more realistic estimate.

1. The current version of the main science recipe (vircam_jitter_microstep_process), does not yet include all the foreseen functionalities such as crosstalk, persistence, defringing and background correction, which may increase the data processing time by approximately 10 to 20%.

2. There is an overhead that comes from the Condor system, which submits the jobs to be processed in parallel. In addition to the time taken by Condor, the main overhead comes from the I/O concurrency, which will depend on the number of processes accessing the same disk at the same time.

3. There is another step at the end of the cascade which joins all the products into 16 extensions. This time is proportional to the number of products created, which in itself depends on the number of input raw frames.

We have used 6 processors to run this test on the main science recipe, using 12 input raw frames and creating 21 products. The detailed processing time of this test is:

```
recipe time = 320 seconds (+ 20% for missing functionalities)
Condor + I/O time = 138 seconds
Join step time = 147 seconds
Total time = 605 seconds (669 sec)
```

If we consider that the time to observe these 12 frames was 360 seconds, then our system with 6 processors was not sufficient to finish the job before the next template started. In order to accomplish the job we would need (669*6)/360 = 11 processors. Our recommended number of processors is **16**, which means one processor per FITS extension.

4 Appendix

```
EXECUTION TIME PER RECIPE
--------------------------------------------------------------------------
JITTER MICROSTEP PROCESS

Recipe:  vircam_jitter_microstep_process
Input:  12 raw science
Output:  12 simple_images, 3 super_frames, 3 super_frames_confidence_maps,
1 stacked_jittered_image, 1 confidence_map, 1 object_catalogue

Time:  120 sec
--------------------------------------------------------------------------

--------------------------------------------------------------------------
LINEARITY ANALYSE

Recipe:  vircam_linearity_analyse
Input:  6 raw darks, 6 raw dome flats
Output:  1 channel_table, 1 bad_pixels_map
```

```
Time:  21 sec
--------------------------------------------------------------------------

--------------------------------------------------------------------------
DARK COMBINE

Recipe:  vircam_dark_combine
Input:  5 raw darks, 1 master_bias
Output:  1 master_dark

Time:  12 sec
--------------------------------------------------------------------------

--------------------------------------------------------------------------
DARK CURRENT

Recipe:  vircam_dark_current
Input:  4 raw darks
Output:  1 master_dark_current

Time:  4 sec
--------------------------------------------------------------------------

--------------------------------------------------------------------------
DETECTOR NOISE

Recipe:  vircam_detector_noise
Input:  2 raw darks, 2 raw dome flats
Output:  1 readgain_file

Time:  3 sec
--------------------------------------------------------------------------

--------------------------------------------------------------------------
MASTER DOME FLAT

Recipe:  vircam_dome_flat_combine
Input:  5 raw dome flats
Output:  1 master_dome_flat

Time:  26 sec
--------------------------------------------------------------------------

--------------------------------------------------------------------------
MASTER TWILIGHT FLAT

Recipe:  vircam_twilight_flat_combine
Input:  3 raw twilight flats
Output:  1 master_twilight_flat, 1 master_conf_map

Time:  21 sec
--------------------------------------------------------------------------
```