# Performance of the LOTAAS v.1 pipeline on cartesius

### Time taken by the individual pipeline components per beam (24-core node)

fits2fil: 6min
rfifind: 15min
mpiprepsubband (253 trials): 3min
single pulse search: 1min
realfft: 10sec
rednoise: 10sec
accelsearch (-zmax=0 ; –numharm=16): 1min20sec
accelsearch (-zmax=50 ; –numharm=16): 12min
accelsearch (-zmax=50 ; –numharm=8): 5min
accelsearch (-zmax=50 ; –numharm=8): 26min
plots: 20sec
python sifting and folding: 21min
pfd scrunching: 5sec
data copying: a few secs
candidate scoring: a few secs

### Total time spent for the first large set of DM trials (0-4000)

mpiprepsubband: 40min
sp: 16min
realfft: 3.5min
rednoise: 3.5min
accelsearch (zmax=0;numharm=16): 21min
accelsearch (zmax=50;numharm=16): 192min
accelsearch (zmax=50;numharm=8): 80min
accelsearch (zmax=200;numharm=8): 416min

### Total time spent for the second large set of DM trials (4000-10000)

mpiprepsubband: 24min
sp: 8min
realfft: 2min
rednoise: 2min
accelsearch (zmax=0;numharm=16): 11min
accelsearch (zmax=50;numharm=16): 96min
accelsearch (zmax=50;numharm=8): 40min
accelsearch (zmax=200;numharm=8): 208min

| % time alloc. | zmax=0;numharm=16 | zmax=50;numharm=16 | zmax=50;numharm=8 | zmax=200;numharm=8 |
|---|---|---|---|---|
| **fil conversion** | **3** | **1** | **2** | **<1** |
| **rfifind** | **9** | **3** | **6** | **2** |
| **dedispersion** | **37** | **16** | **25** | **8** |
| **sp search** | **14** | **5** | **9** | **3** |
| **realfft** | **3** | **1** | **2** | **<1** |
| **rednoise** | **3** | **1** | **2** | **<1** |
| **accelsearch** | **18** | **67** | **46** | **81** |
| **folding** | **12** | **5** | **8** | **3** |
| **data copying/etc** | **1** | **1** | **1** | **<1** |

Total processing time per beam (zmax=0;numharm=16): ~3hours
Total processing time per beam (zmax=50;numharm=16): ~7hours
Total processing time per beam (zmax=50;numharm=8): ~5hours
Total processing time per beam (zmax=200;numharm=8): ~13h40m

## Performance of the LOTAAS v.1 GPU pipeline on cartesius

mpiprepsubband (253 trials): 38sec

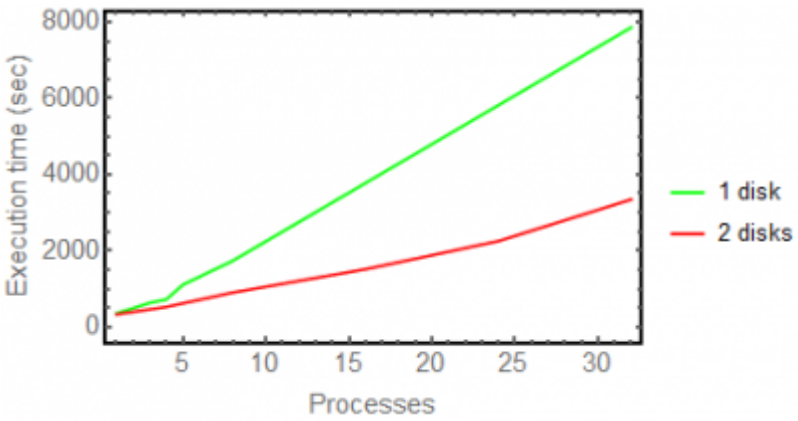
## Data transferring (CEP$/LTA)

32-bit to 8-bit downsampling on CEP2 (per observation): 6-8 hours
Transferring from CEP2 to LTA (per observation): 2-3 hours
Observation downloading on cartesius (1-core): ~8hours
Observation downloading on cartesius (home area, 8jobs in parallel.sh):<2hours

## Benchmarks for filterbank creation with psrfits2fil

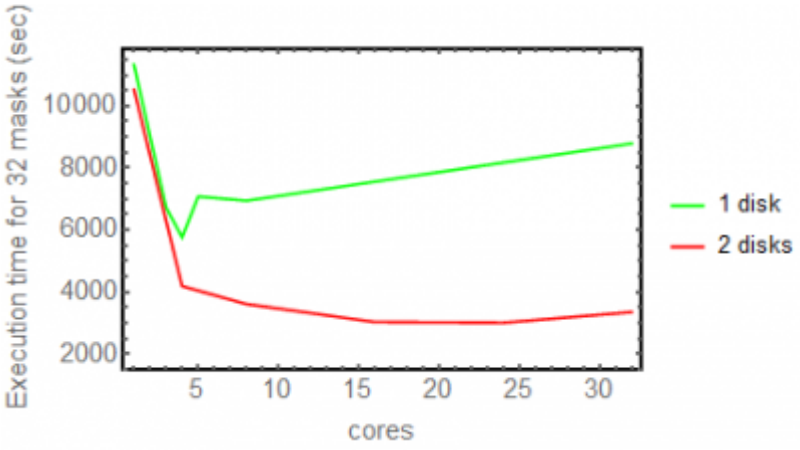
psrfits2fil was executed with different numbers of parallel processes. The following plot shows the amount of time needed in order to create the fil files for various cases of parallel psrfits2fil instances.

Using the same disk the following cases were tried: 1,3,4,5,8,12,16. Anything above 16 is just an extrapolation
for 2 disks: 1,4,8,12,16,20,24,28,32




Using multithreading with 2 disks, gives a smooth linear performance up to 24 cores, and then it turns slightly worse, probably due to I/O.

Using the above results, I extrapolated the time needed with each work strategy in order to compute 32 filtebanks.



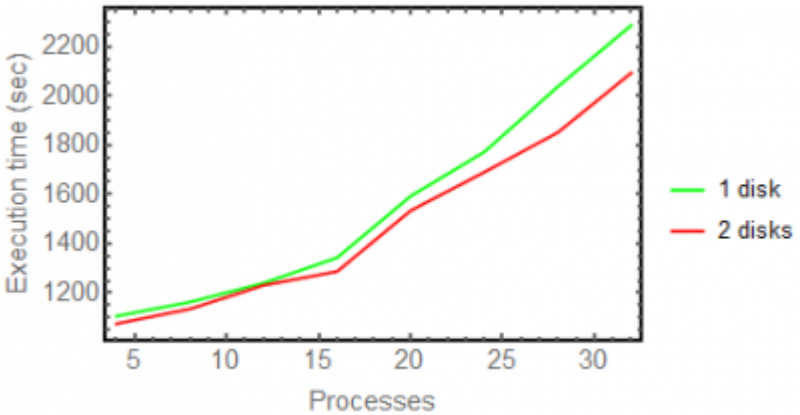

When using the same disk, the fastest execution time is achieved having 4 psrfits2fil instances running in parallel. Above that, probably disk I/O normalises all the results and the performance decreases gradually, probably due to the increased I/O calls, since the throughput must already be saturated.

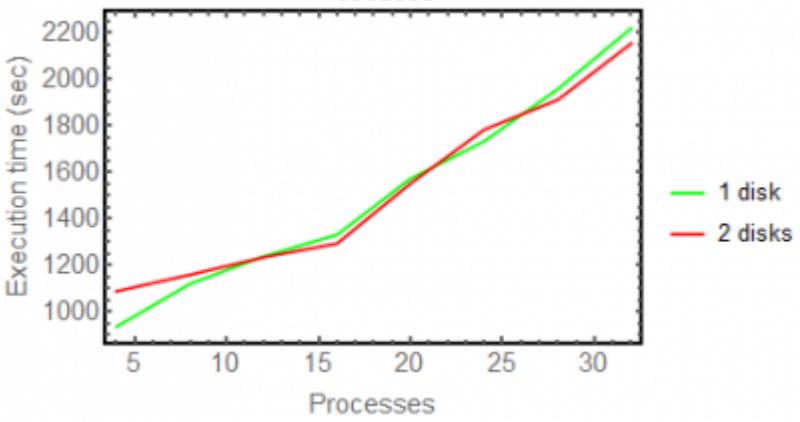
Using 2 disks, the performance is significantly better, and the best results are achieved using 24 psrfits2fil instances in parallel, although the difference remains small.

## rfifind benchmarks

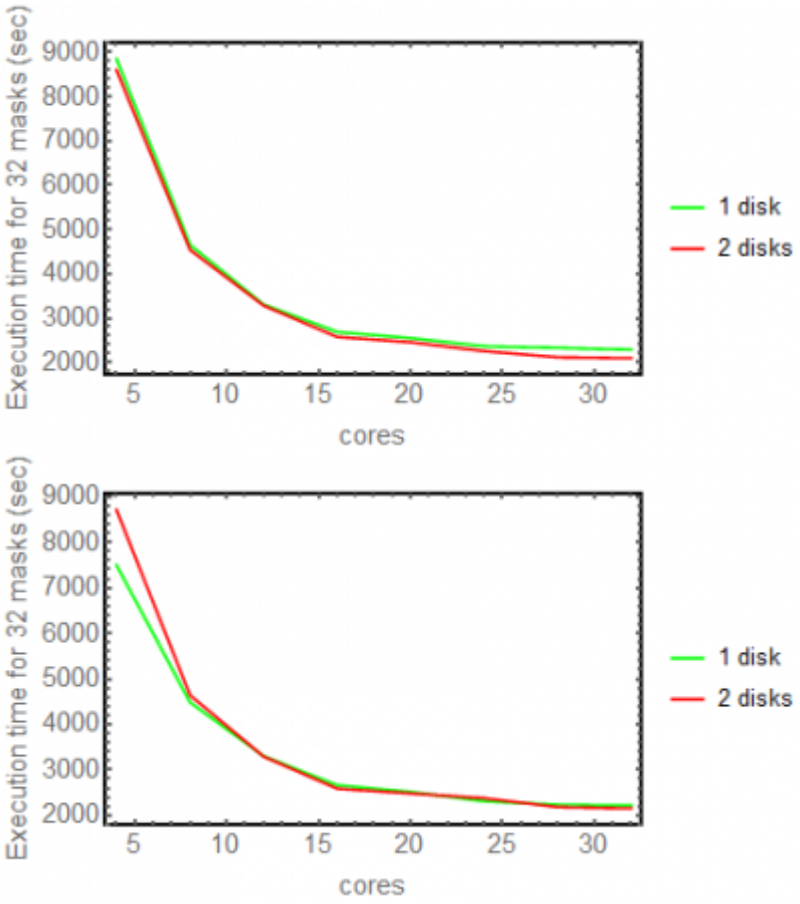
I ran the same tests twice.

I created rfi masks running rfifind in parallel for 4,8,12,16,20,24,28 and 32 cores (>16 hyperthreaded).
In the following plots I plot the number of parallel instances of rfifind executed (x-axis) and the time taken for these to be completed (y-axis).



In the following plots, I extrapolated the above results in order to find the optimal number of parallel jobs in order to compute 32 rfi masks




From the above, we can conclude that using 1 or 2 disks does not make a big difference. Also, hyperthreading works smoothly, and indeed the best strategy is to have the maximum possible number of rfifind instances running in parallel.

## Dedispersion strategies for Cartesius

The max number of timeseries (.dat files) that can be stored in 32GB of /dev/shm.
The dedispersion plan is split at two points, the first at DM~45 without downsampling, and the second at DM~135 with x2 downsampling.

Part1: 480<file number<504
Part2: 984<file number<1008
Part3: 1944<file number<1968