

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=5): 80min
 accelsearch (zmax=200;numharm=5): 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=5): 40min
 accelsearch (zmax=200;numharm=5): 208

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

Total processing time per beam (zmax=0;numharm=16): ~3hours

Total processing time per beam (zmax=50;numharm=16): ~7hours

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

From:
<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=dragnet:benchmarks_of_the_lotaas_pipelines&rev=1440115588

Last update: **2015-08-21 00:06**

