CMS multicore jobs at PIC

PIC
port d'informació científica

Carles Acosta Silva
Bruno Rodríguez Rodríguez

introduction

backfilling

mcfloat

backfilling vs mcfloat results

conclusions

introduction

backfilling

mcfloat

backfilling vs mcfloat results

conclusions

introduction

PIC is a multi-VO site. Atlas, CMS, LHCB, magic, etc.

CMS and ATLAS (T1 and T2) are submitting mcore jobs in production

Torque-2.5.13. Maui-3.3.4.

mcore_sl6

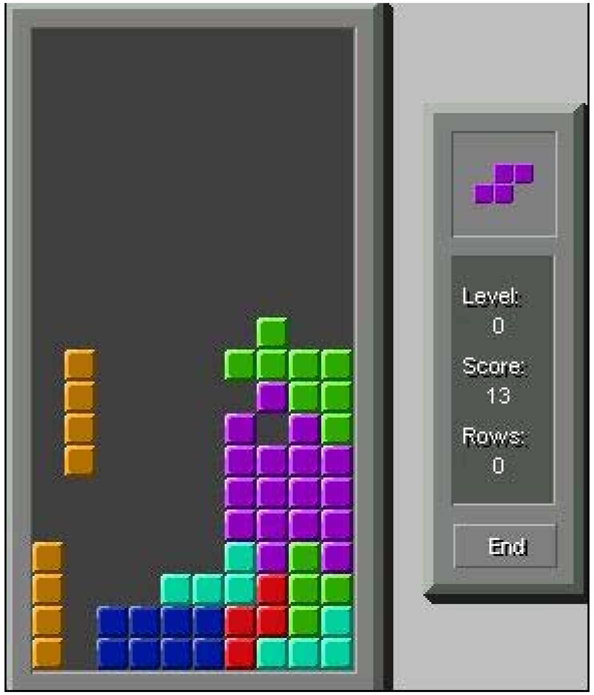
mcore_sl6_atlas

mcore_sl6_at2

3 identical queues for each VO
(historical reasons, monitoring, etc.)

mcore_sl6: right now, the mcore_sl6 queue is used only by CMS

```
# qstat -Q -f mcore_sl6
Queue: mcore_sl6
    queue_type = Execution
    max_user_queuable = 200
    total_jobs = 134
    state_count = Transit:0 Queued:67 Held:0 Waiting:0 
Running:66 Exiting:0
    acl_host_enable = True
    acl_hosts = 
ce09.pic.es,ce08.pic.es,ce07.pic.es,pbs04.pic.es,ce11.pic
.es,ce10.pic.es
    resources_max.walltime = 107:00:00
    resources_default.neednodes = mcore
    resources_default.nodes = 1:ppn=8
    resources_default.walltime = 107:00:00
    acl_group_enable = True
    acl_groups = cmprd,dteam
    mtime = 1402911369
    resources_assigned.nodect = 66
    enabled = True
    started = True
```



Scheduling mcore jobs

2 approaches tried at PIC

- Backfilling with Maui config
- mcfloat script (Jeff Templon, NIKHEF)

introduction

backfilling

mcfloat

backfilling vs mcfloat results

conclusions

Backfilling

Backfill allows to run jobs out of order from the priorization to maximize the use of our resources

In general,

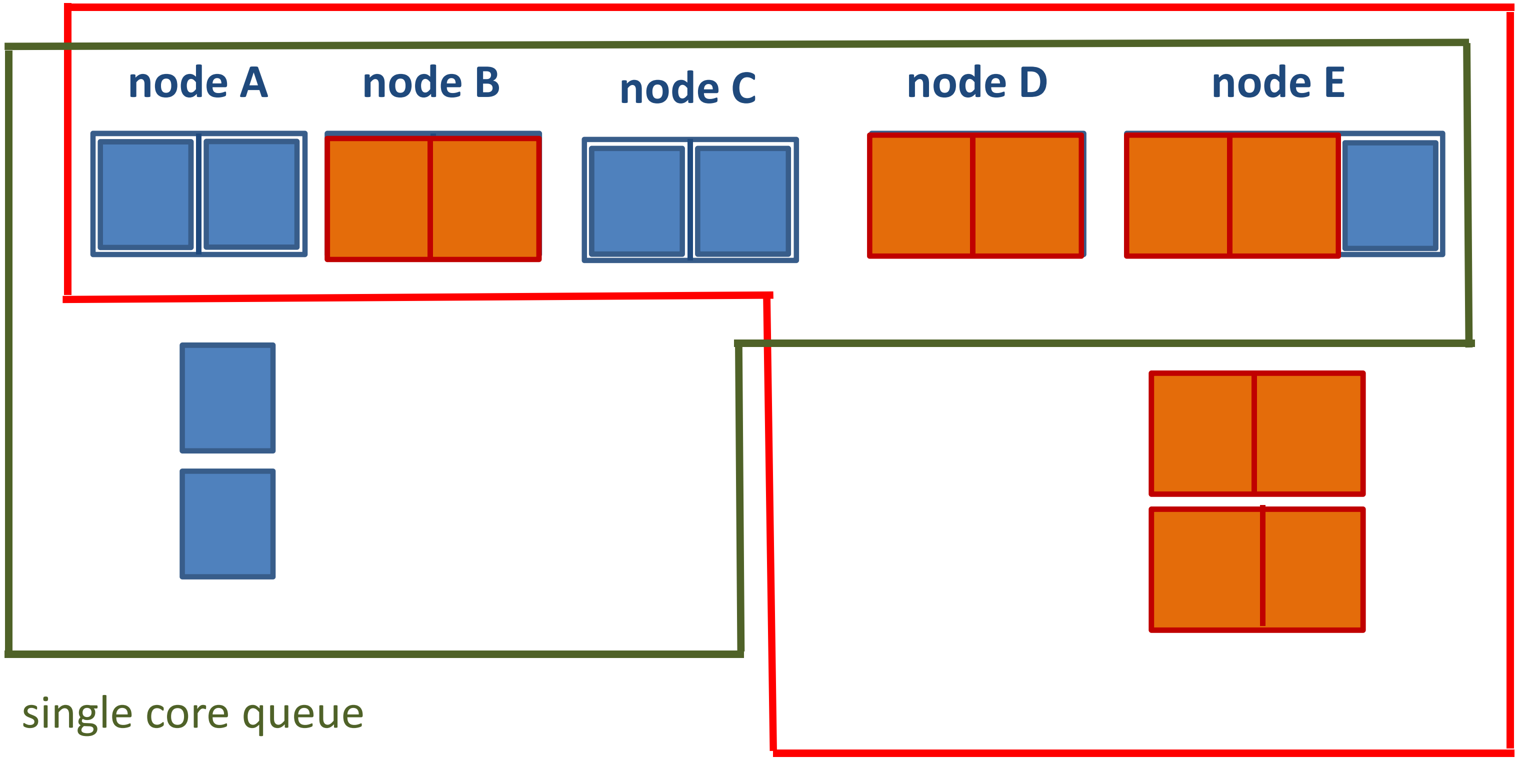
1) favor smaller and shorter running jobs
2) the influence of the job priorization is reduced
3) strong dependence with job wallclock prediction

Backfill policy in Maui: one tunning for each site requeriments

A lot of Maui backfilling options to tune!

```
BACKFILLPOLICY -> FIRSTFIT
RESERVATIONPOLICY -> CURRENTHIGHEST
RESERVATIONDEPTH -> 64
BFCHUNKSIZE -> 8
BFCHUNKDURATION -> 01:30:00
```

Backfilling



single core queue

mcore queue

introduction

backfilling

mcfloat

backfilling vs mcfloat results

conclusions

mcfloat

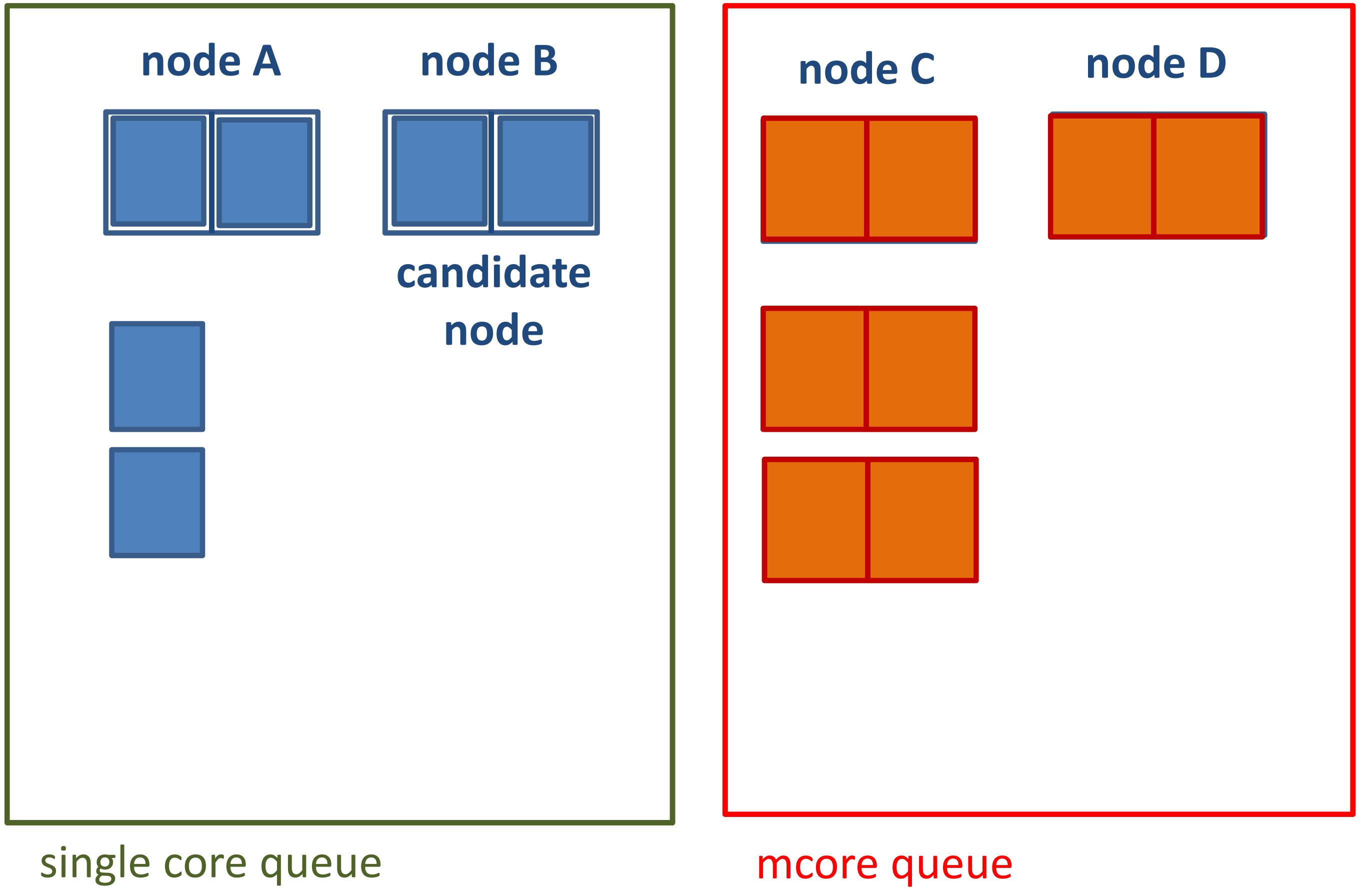
Python script developed at NIKHEF. Please refer to Jeff Templon talks to obtain further information (https://indico.cern.ch/event/305625/)

Basically, moves the WNs in and out of the single core and mcore core queues

- adjusts WN properties to drain the nodes and free slots
- keeps the mcore slots open
- Tune the system to minimize draining impact based on these parameters :

CANDIDATE NODES: 95 nodes (968 slots)
MAXDRAIN: 16 (nodes)
MAXFREE: 73 (slots)

mcfloat



node A

node B

candidate node

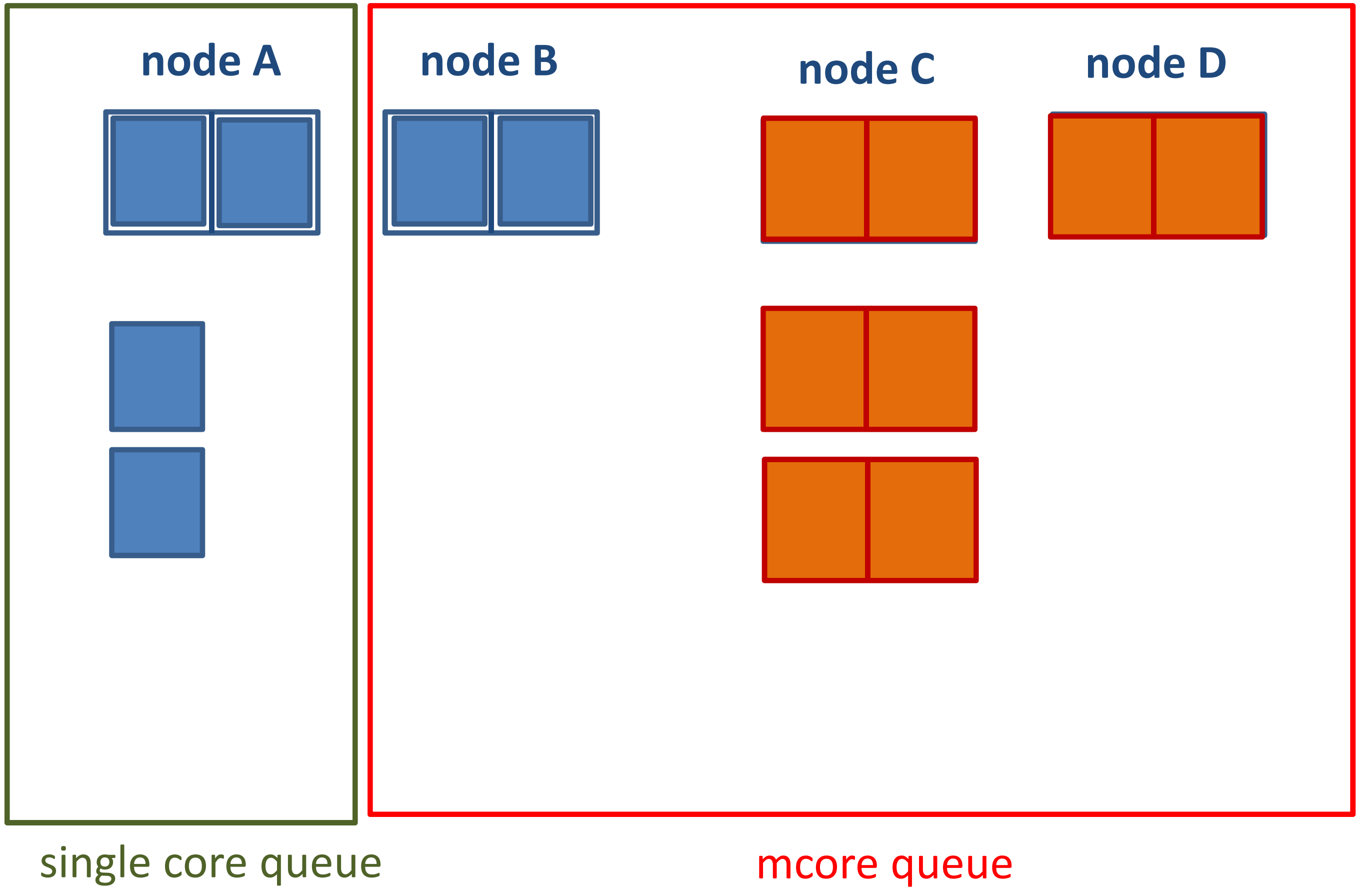
node C

node D

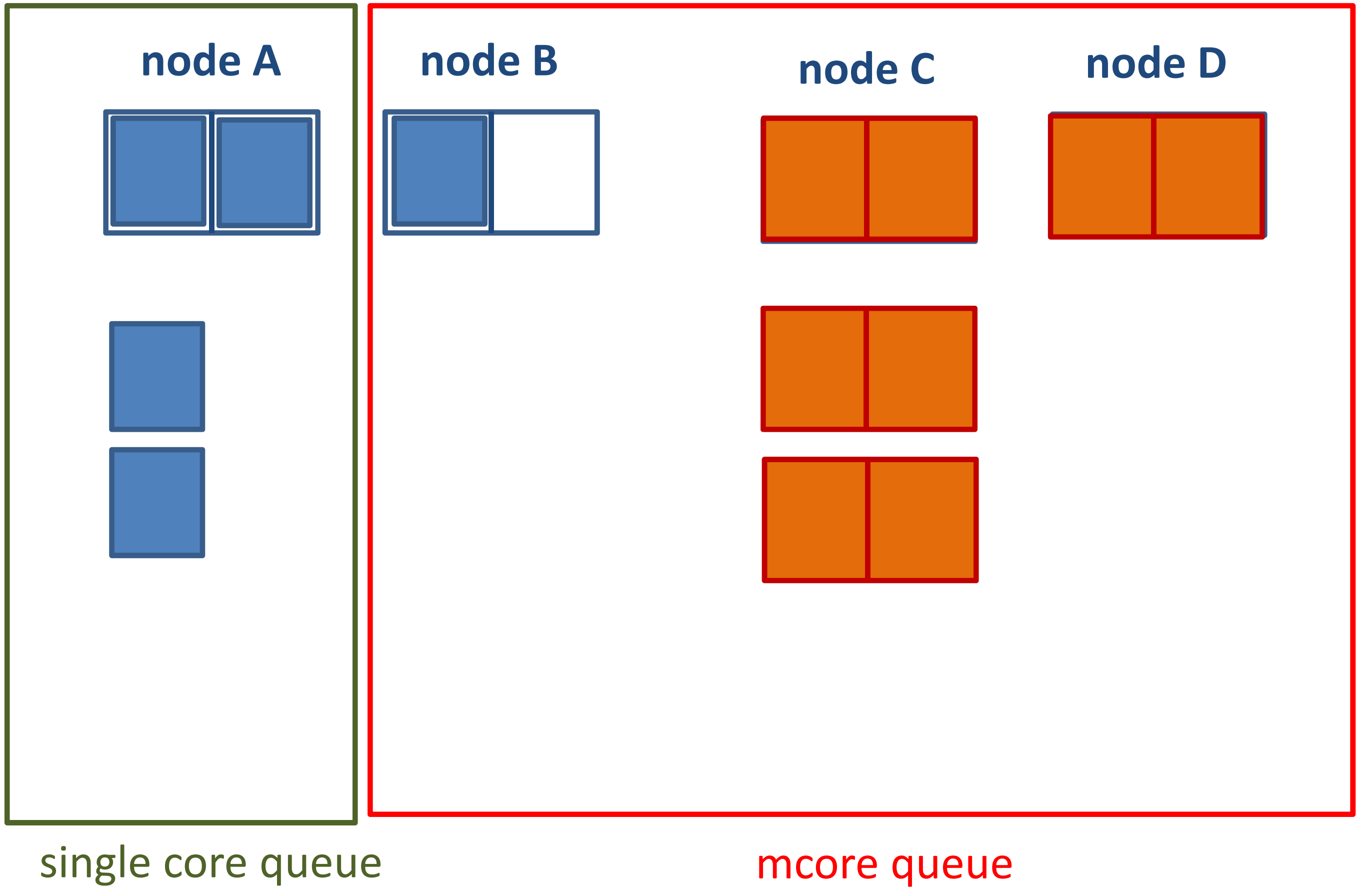
single core queue

mcore queue

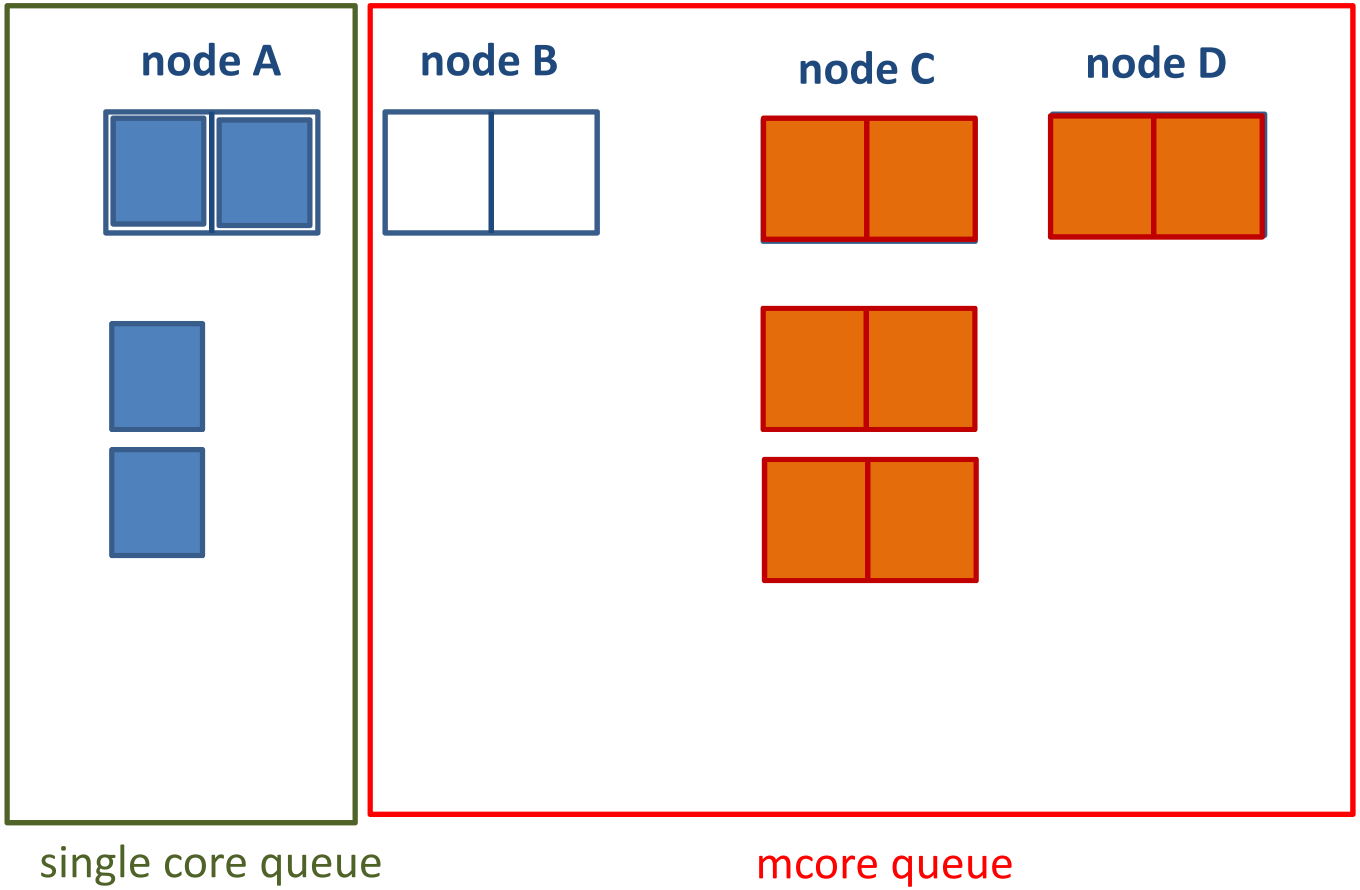
mcfloat



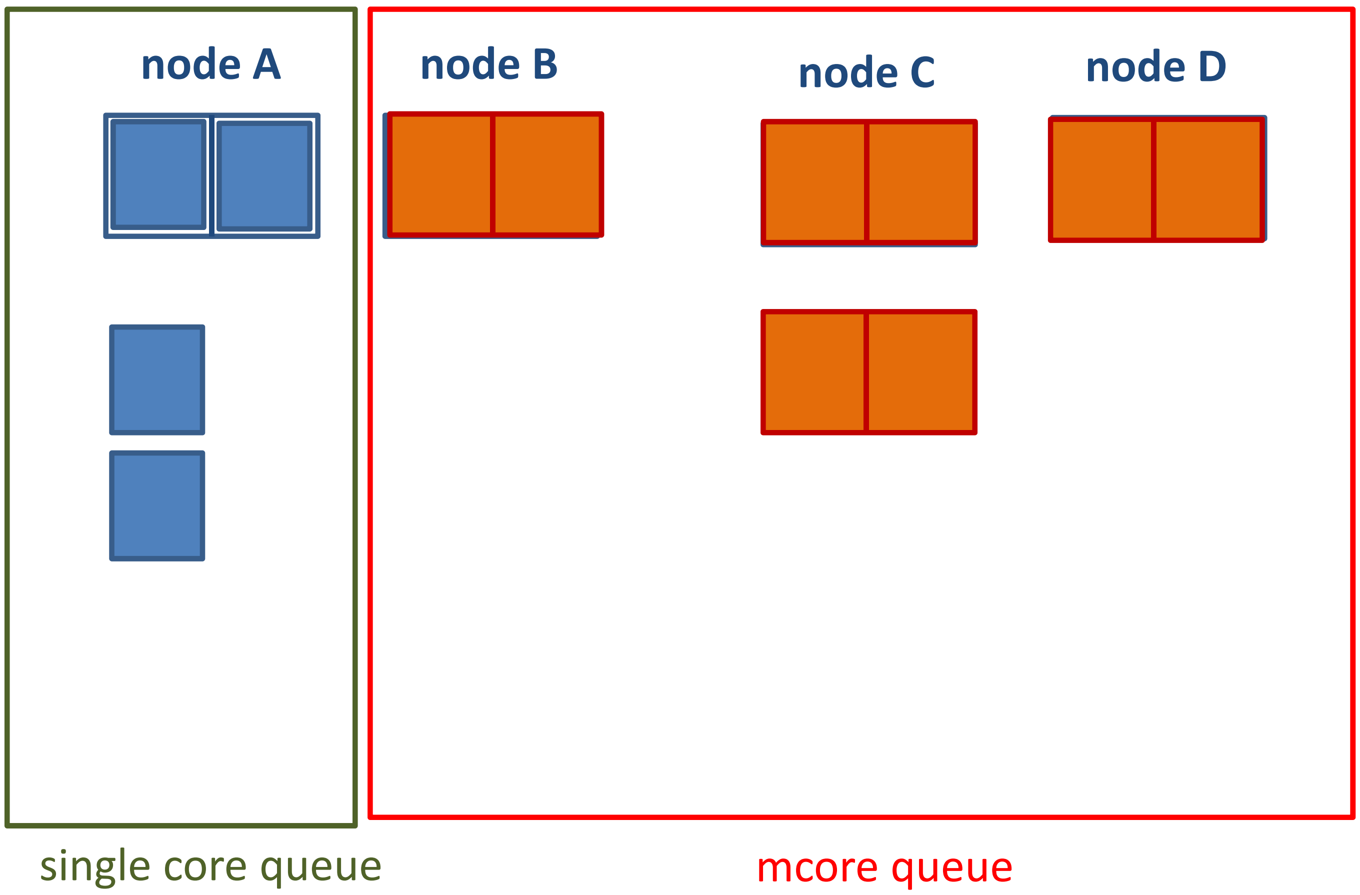
mcfloat



mcfloat



mcfloat



introduction

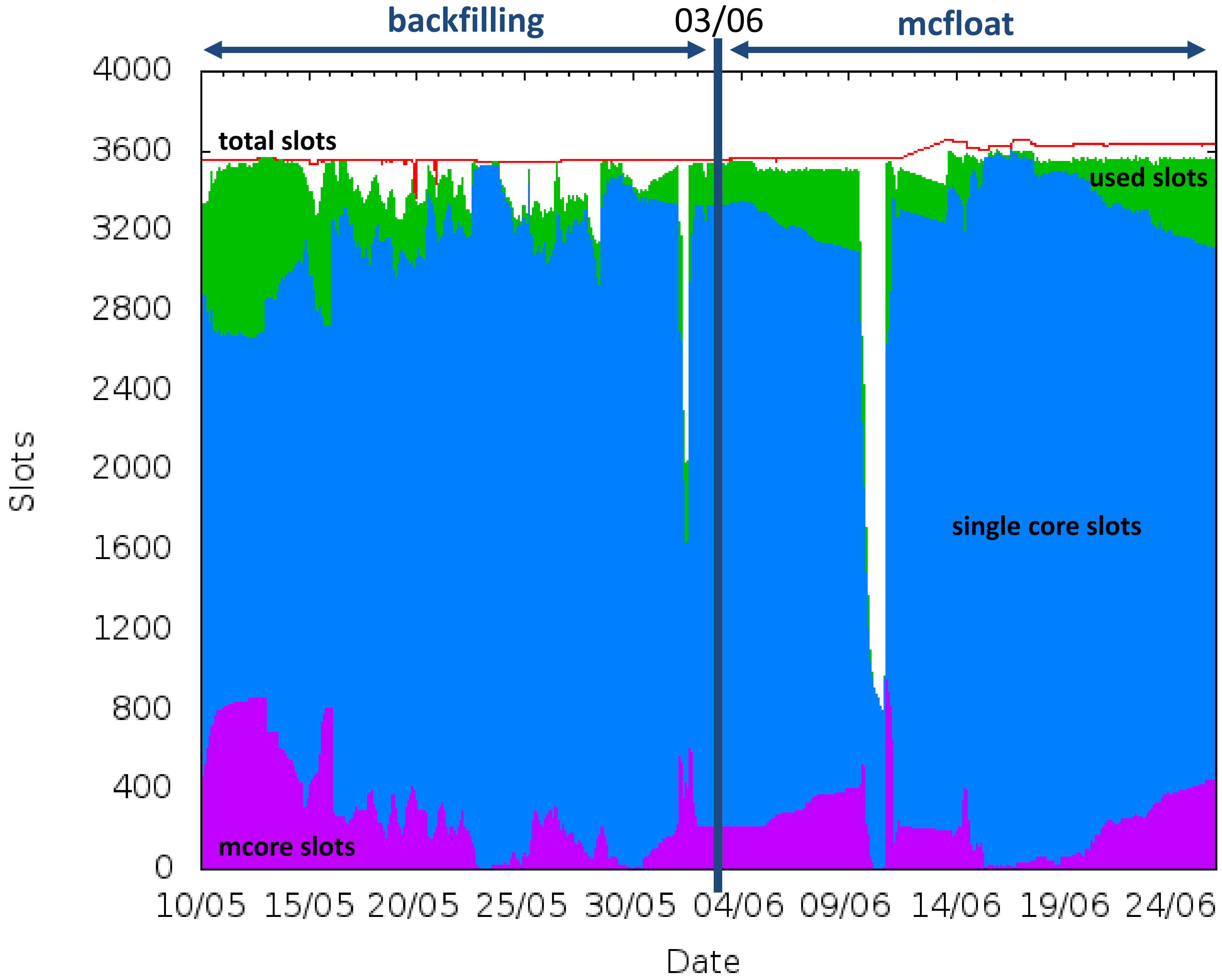
backfilling

mcfloat

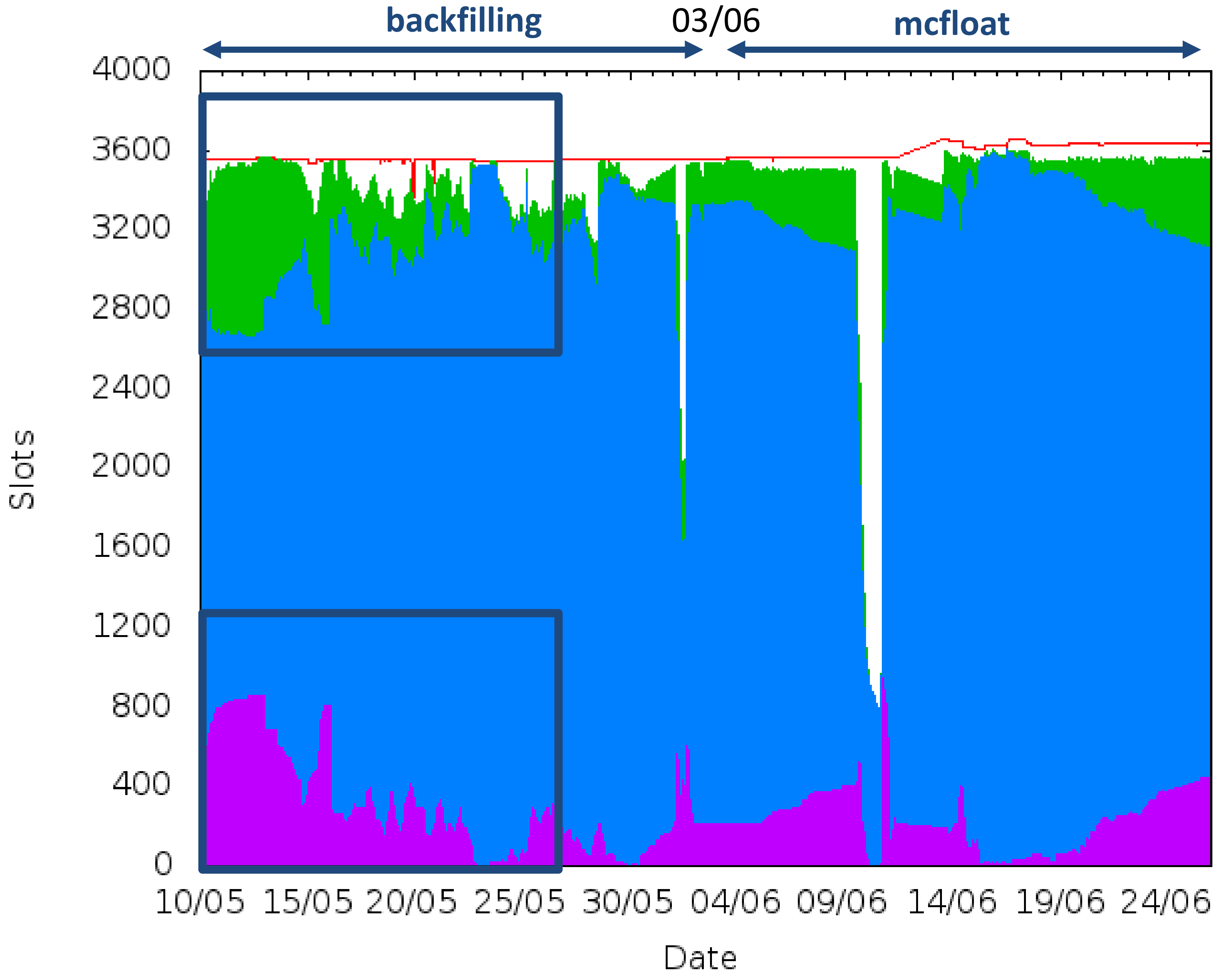
backfilling vs mcfloat results

conclusions

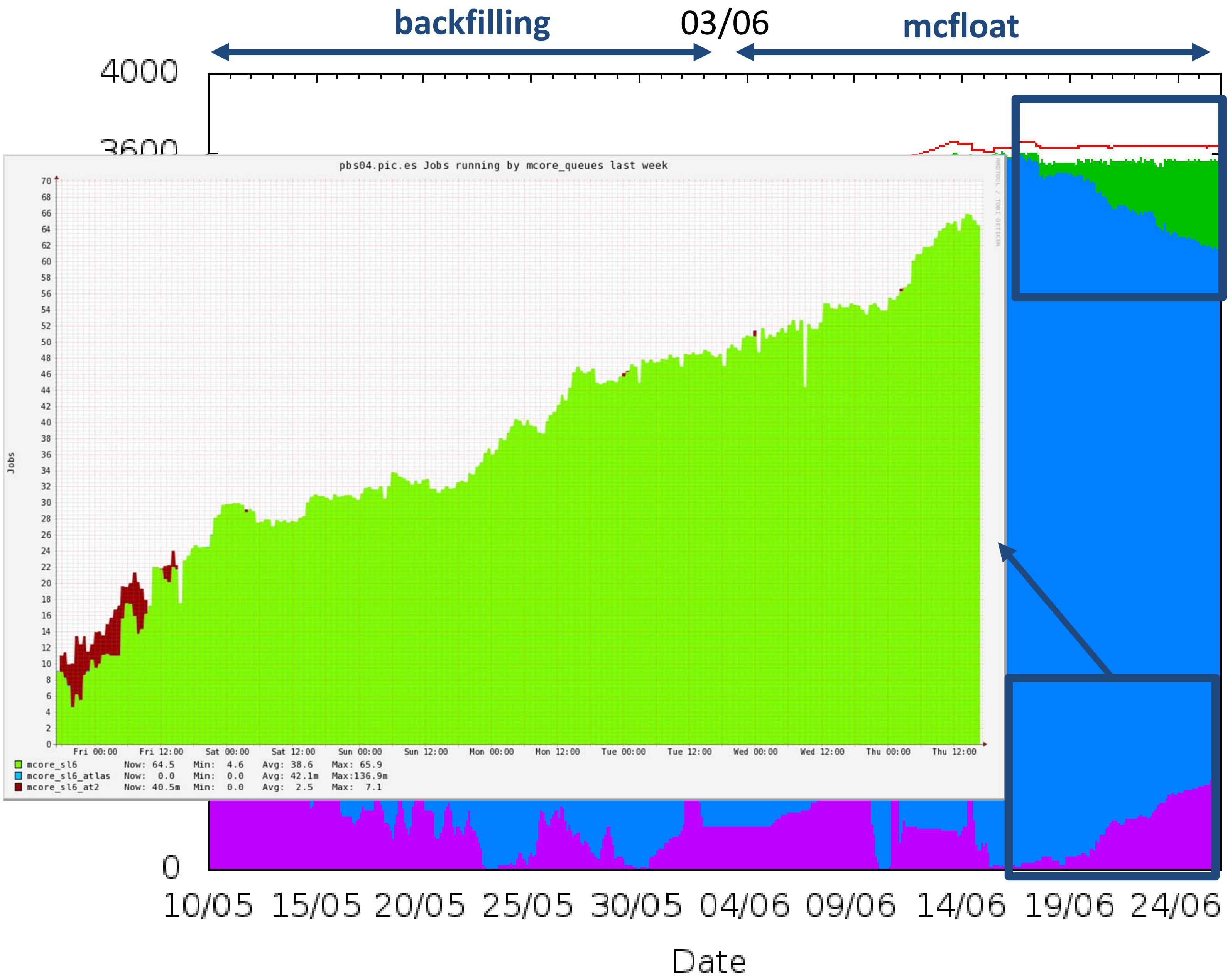
backfilling vs mcfloat results



backfilling vs mcfloat results

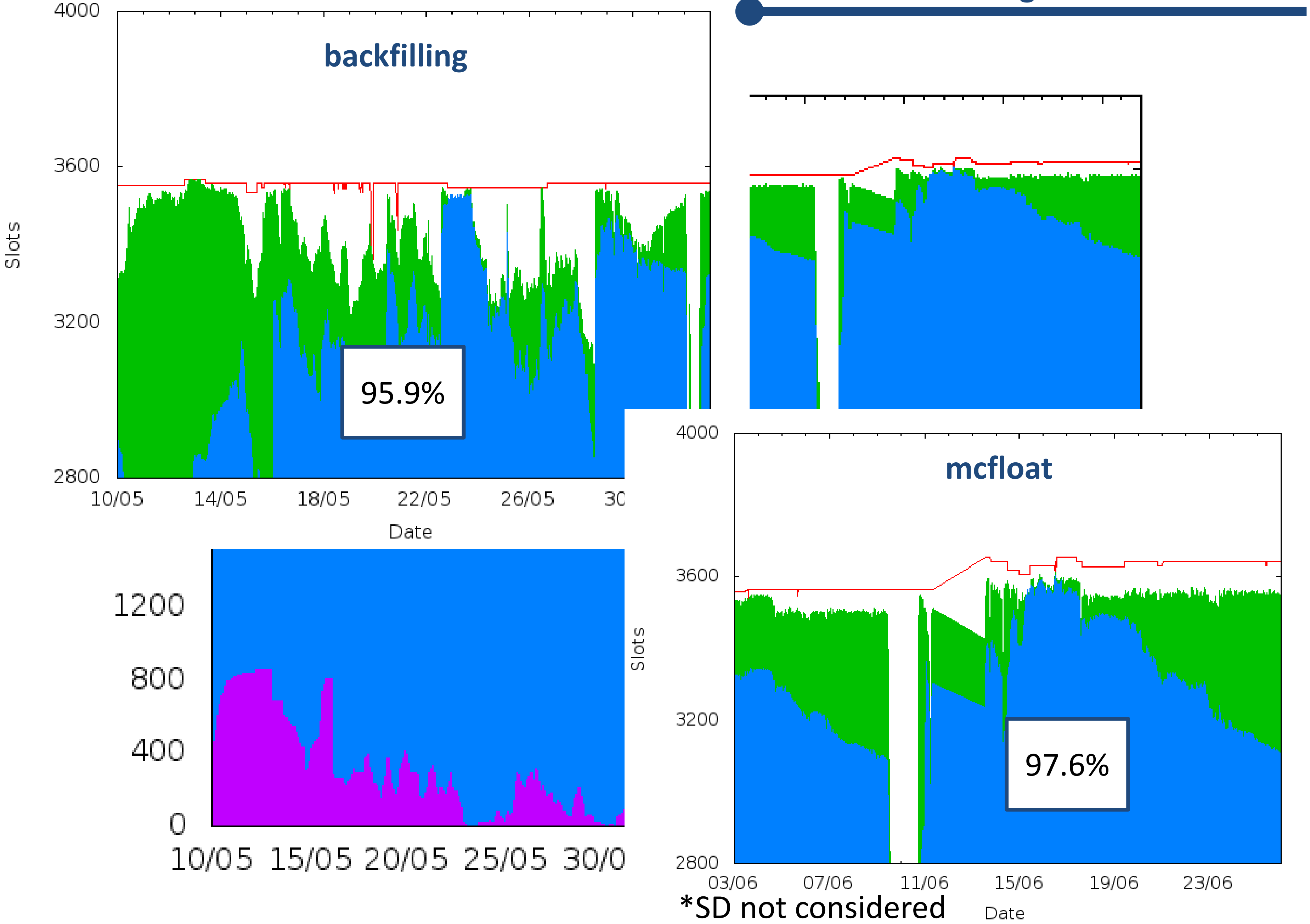


backfilling vs mcfloat results



```
🟩 mcore_sl6        Now: 64.5   Min:  4.6   Avg: 38.6   Max: 65.9
🟦 mcore_sl6_atlas  Now:  0.0   Min:  0.0   Avg: 42.1m  Max:136.9m
🟥 mcore_sl6_at2    Now: 40.5m  Min:  0.0   Avg:  2.5   Max:  7.1
```

backfilling vs mcfloat results

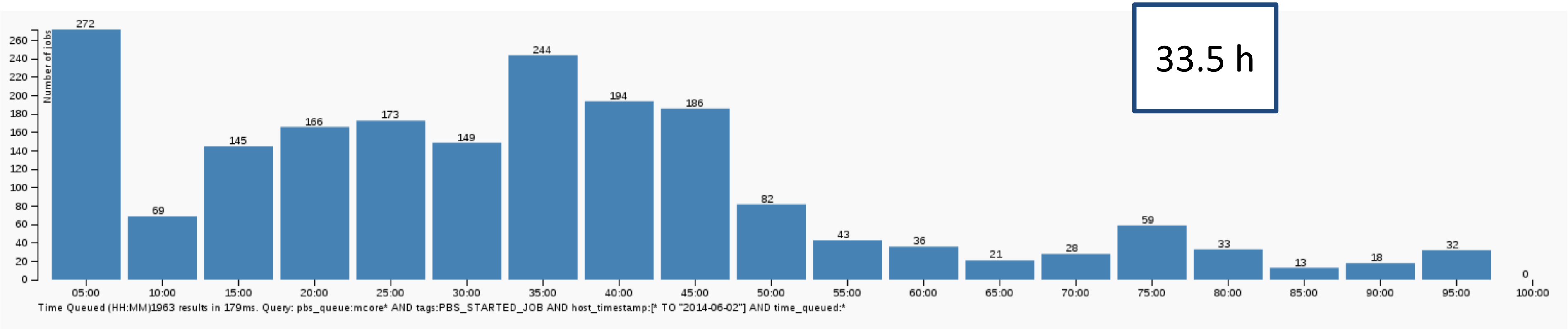


*SD not considered

Job queued time (backfilling) 10/05 - 02/06

78.4% CMS
21.6% Atlas T1

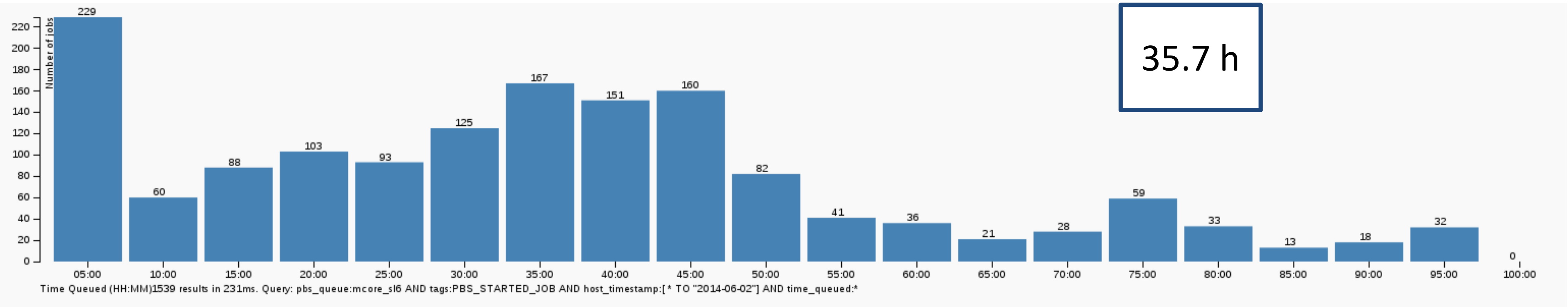
total mcore



1963 results in 179ms. Query: pbs_queue:mcore* AND tags:PBS_STARTED_JOB AND host_timestamp:[* TO "2014-06-02"] AND time_queued:*

33.5 h

CMS mcore



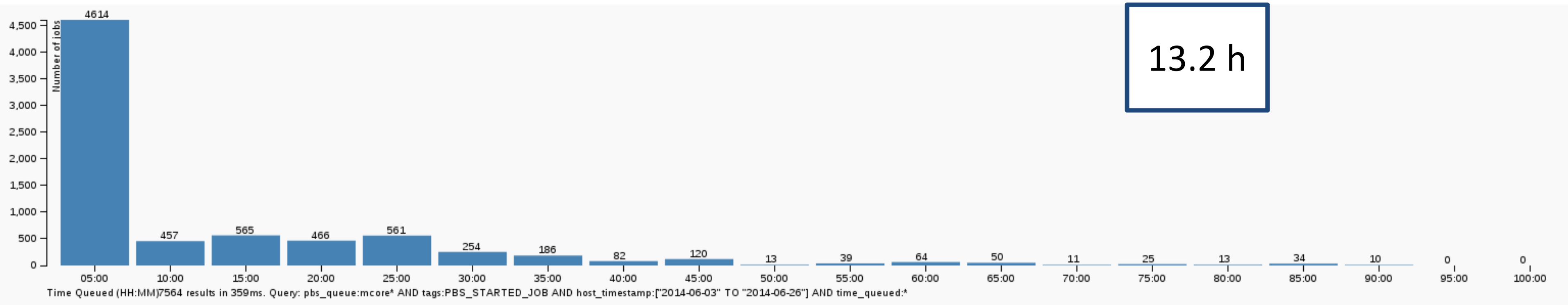
1539 results in 231ms. Query: pbs_queue:mcore_sl6 AND tags:PBS_STARTED_JOB AND host_timestamp:[* TO "2014-06-02"] AND time_queued:*

35.7 h

Job queued time (mcfloat) 03/06 - 26/06

41.4% CMS
58.6% Atlas T1+T2

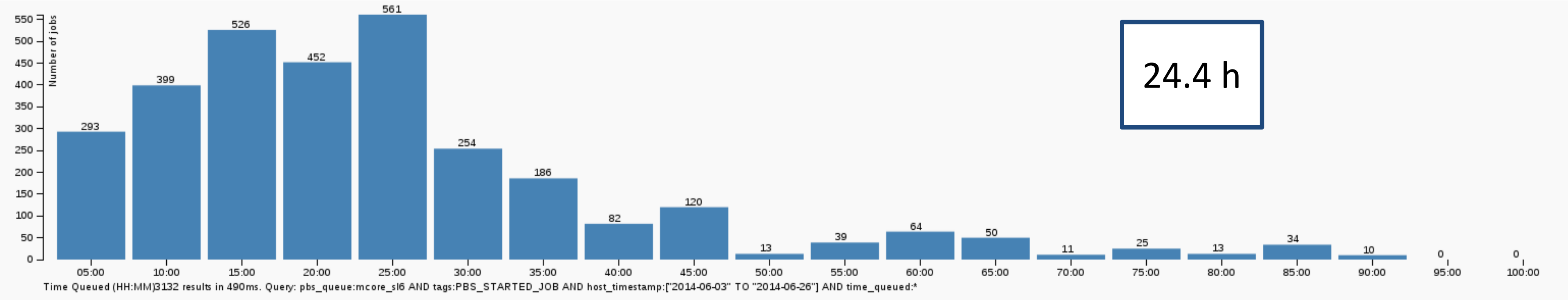
total mcore



Time Queued (HH:MM)7564 results in 359ms. Query: pbs_queue:mcore* AND tags:PBS_STARTED_JOB AND host_timestamp:["2014-06-03" TO "2014-06-26"] AND time_queued:*

13.2 h

CMS mcore



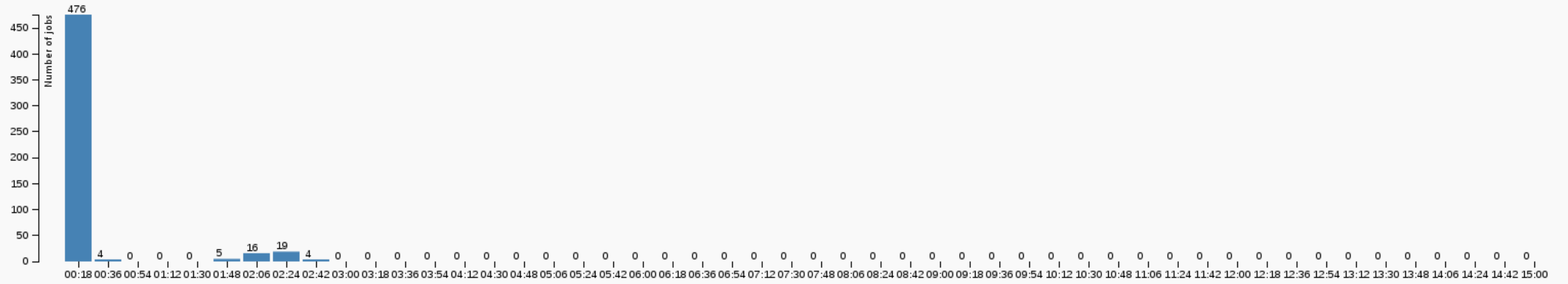
Time Queued (HH:MM)3132 results in 490ms. Query: pbs_queue:mcore_sl6 AND tags:PBS_STARTED_JOB AND host_timestamp:["2014-06-03" TO "2014-06-26"] AND time_queued:*

24.4 h

backfilling vs mcfloat results

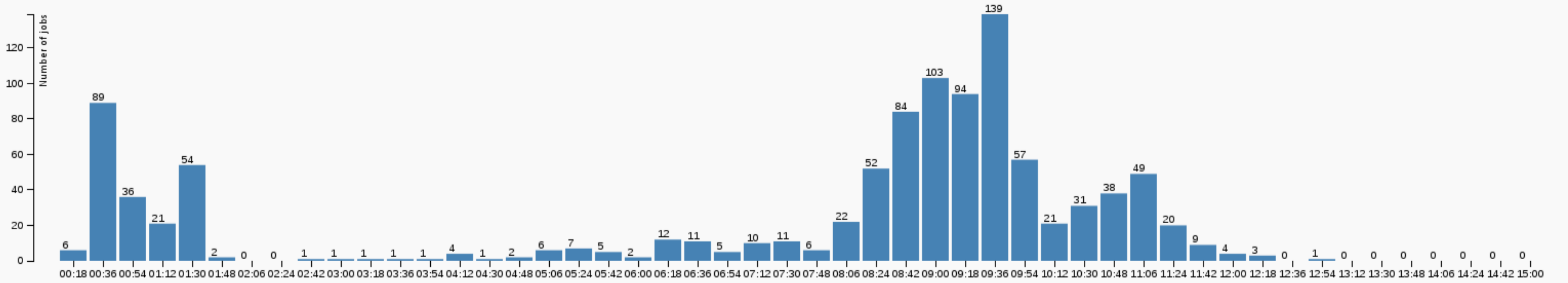
Job running time (walltime)

Atlas T1+T2 mcore



524 results in 315ms. Query: pbs_queue:mcore_sl6_at* AND tags:PBS_FINISHED_JOB AND host_timestamp:["2014-05-10" TO "2014-06-26"] AND time_queued:* AND pbs_used_walltime_seconds:*

CMS mcore



1025 results in 413ms. Query: pbs_queue:mcore_sl6 AND tags:PBS_FINISHED_JOB AND host_timestamp:["2014-05-10" TO "2014-06-26"] AND time_queued:* AND pbs_used_walltime_seconds:*

introduction

backfilling

mcfloat

backfilling vs mcfloat results

conclusions

conclusions

- Temporary PIC configuration in 3 queues for the different experiments but considering to join Atlas T1 and CMS in the same queue in the future

- After testing the Torque+Maui backfilling configuration and the custom mcfloat script:

mcfloat solution is clearly better

- better use of the whole farm
- lower job queued time

- Difference queued time between CMS and Atlas due to the different Fair-Share, job running time and submission patterns