testing multicore jobs at PIC
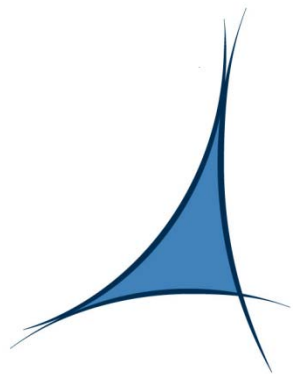
PIC is a multi-VO site. Atlas, CMS, LHCB, magic, etc.

CMS and ATLAS are using the test environment of PIC to test multicore jobs

1 VM cream-CE (EMI-3 // SL6.4)

1 VM PBS server (EMI-3 // SL6.4). Torque-2.5.13. Maui-3.3.4.

WNs

```
# qstat -q

server: pbs02.pic.es

Queue            Memory CPU Time Walltime Node  Run Que Lm  State
---------------- ------ -------- -------- ----  --- --- --  -----
cms_mcore_test     --   800:00:0 107:00:0  --    0   0 --   E R
pic_test_atlas     --   800:00:0 107:00:0  --    0   0 --   E R
test               --   80:00:00 87:00:00  --    0   0 --   E R
                                                ----- -----
                                                  0     0
```

## cms_mcore_test queue: 2 nodes of 8 Torque slots for CMS

```
# qstat -Q -f cms_mcore_test
Queue: cms_mcore_test
    queue_type = Execution
    total_jobs = 0
    state_count = Transit:0 Queued:0 Held:0 Waiting:0
Running:0 Exiting:0
    acl_host_enable = True
    acl_hosts = ce02-test.pic.es,ui01-test.pic.es
    resources_max.cput = 800:00:00
    resources_max.walltime = 107:00:00
    resources_default.cput = 800:00:00
    resources_default.neednodes = cms_mcore
    resources_default.walltime = 107:00:00
    acl_group_enable = True
    acl_groups =
cmpilot,cmprd,cmt1p,cmt1a,cmhiprod,cmhit1prod,dteam,cms,p
ic
    mtime = 1384874932
    resources_assigned.nodect = 0
    enabled = True
    started = True
```

```
# pbsnodes td457.pic.es
td457.pic.es
    state = free          → 1 slot=1core
    np = 8
    properties = cms_mcore,test
    ntype = cluster
    status =
rectime=1391164749,varattr=,jobs=,state=free,
size=430077936kb:459048096kb,netload=24047749
0428,gres=cpu_factor:=2.7709,loadave=0.00,ncp
us=8,physmem=24592292kb,availmem=47849076kb,t
otmem=49183644kb,idletime=1540753,nusers=0,ns
essions=? 0,sessions=? 0,uname=Linux
td457.pic.es 2.6.32-431.1.2.el6.x86_64 #1 SMP
Thu Dec 12 13:59:19 CST 2013
x86_64,opsys=linux
    gpus = 0
```
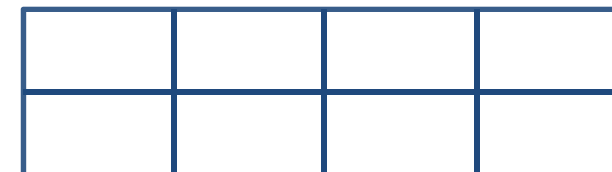
1 slot=1core

Node A
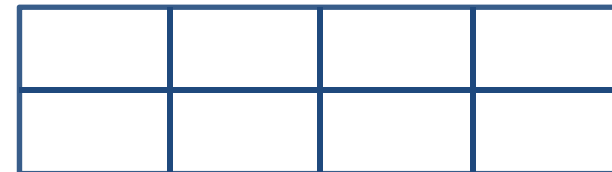
Node B

SMP qsub
```
$ echo "/usr/bin/stress –cpu X" | qsub -q cms_mcore_test –l nodes=1:ppn=X
24611209.pbs02.pic.es
```

SMP glite-ce-job-submit
```
$ glite-ce-job-submit -o id-job -a -r ce02-test.pic.es:8443/cream-pbs-cms_mcore_test job-smp.jdl
https://ce02-test.pic.es:8443/CREAM749661960
```

In CREAM JDL syntax:
        WholeNode= false;
        HostNumber = 1;
        CPUNumber = X;

This is "translated" to different
Batch systems (PBS, SGE, Condor, LSF or
SLURM in EMI-3)

SMP qsub

```
$ echo "/usr/bin/stress –cpu 8" | qsub -q cms_mcore_test –l nodes=1:ppn=8
24611209.pbs02.pic.es
```

Sites are publishing their SMP capabilities:

```
$ ldapsearch -x -h bdii.pic.es -p 2170 -LLL -b mds-vo-name=pic,mds-vo-name=local,o=grid | grep
"GlueHostArchitectureSMPSize:"
GlueHostArchitectureSMPSize: 8
GlueHostArchitectureSMPSize: 8
GlueHostArchitectureSMPSize: 8
GlueHostArchitectureSMPSize: 8
GlueHostArchitectureSMPSize: 8
GlueHostArchitectureSMPSize: 8
```

What does this mean with different kind of WNs?
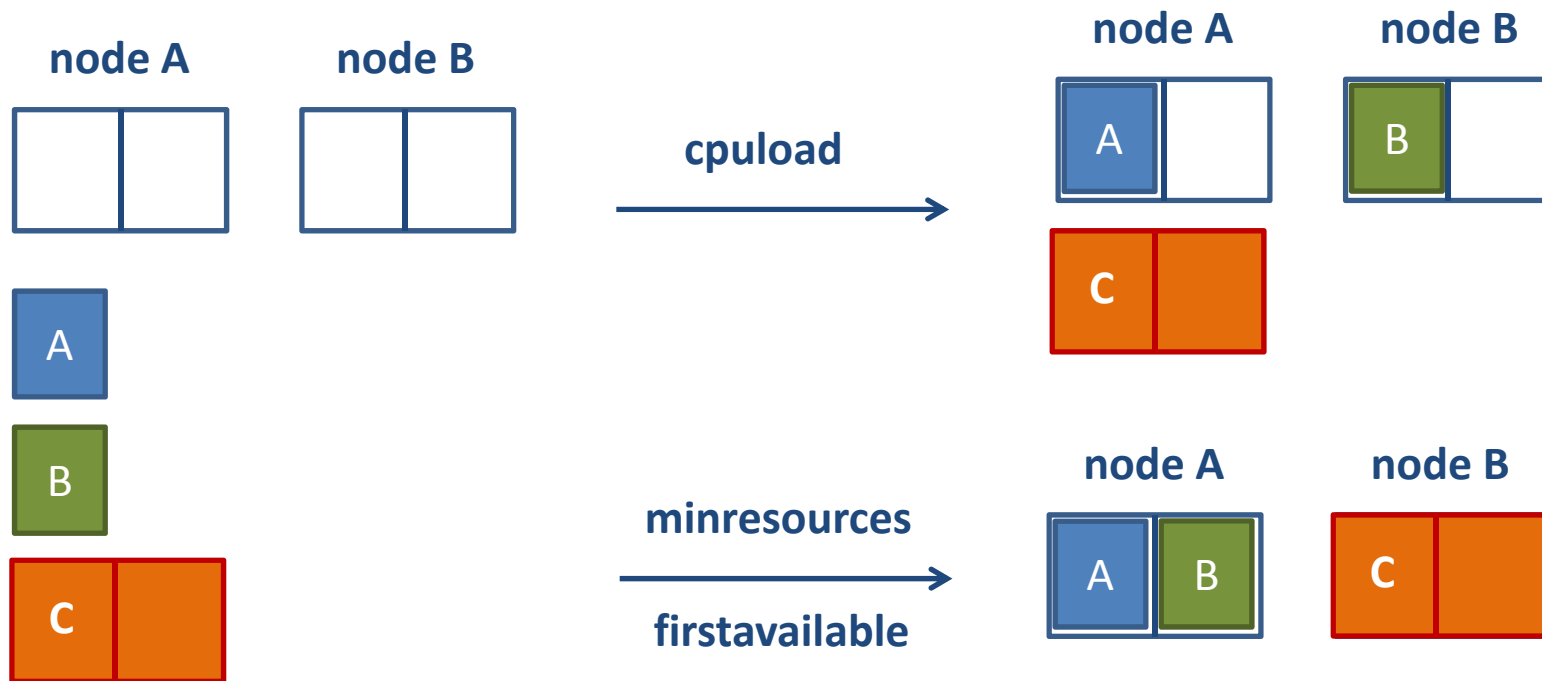The minimum number of slots per node?

Just testing with a queue with very limited resources, but there are no special differences between this queue and the ones in production (cputime)

However, we do not want to allow multicore in production because we want to be sure that the Accounting is correct.

Torque accounting logs correct
APEL can handle this information
WLCG accounting report ready?

Just testing with a queue with very limited resources, but there are no special differences between this queue and the ones in production (cputime)

No special Maui options changed (Nodeallocationpolicy from cpuload to minresources or firstavailable)

node A **node B**

A

B

C

cpuload →

node A **node B**

A B

C

minresources →

firstavailable →

node A **node B**

A B C

Scheduling with backfilling options is the challenge
for sites and experiments-> needed much more testing

We offer the same kind of queue to ATLAS and CMS and it works

When we start to receive multicore jobs in production, we will check the
efficiency, waiting time, etc. in a shared system

These are just preliminary tests, much more testing is needed!

If internal scheduling is doing inside the CMS pilot, this could potentially helps
but again more testing is required