TORQUE and MAUI Tutorial WLCG Workshop January 2007

Steve Traylen, CERN, steve.traylen@cern.ch

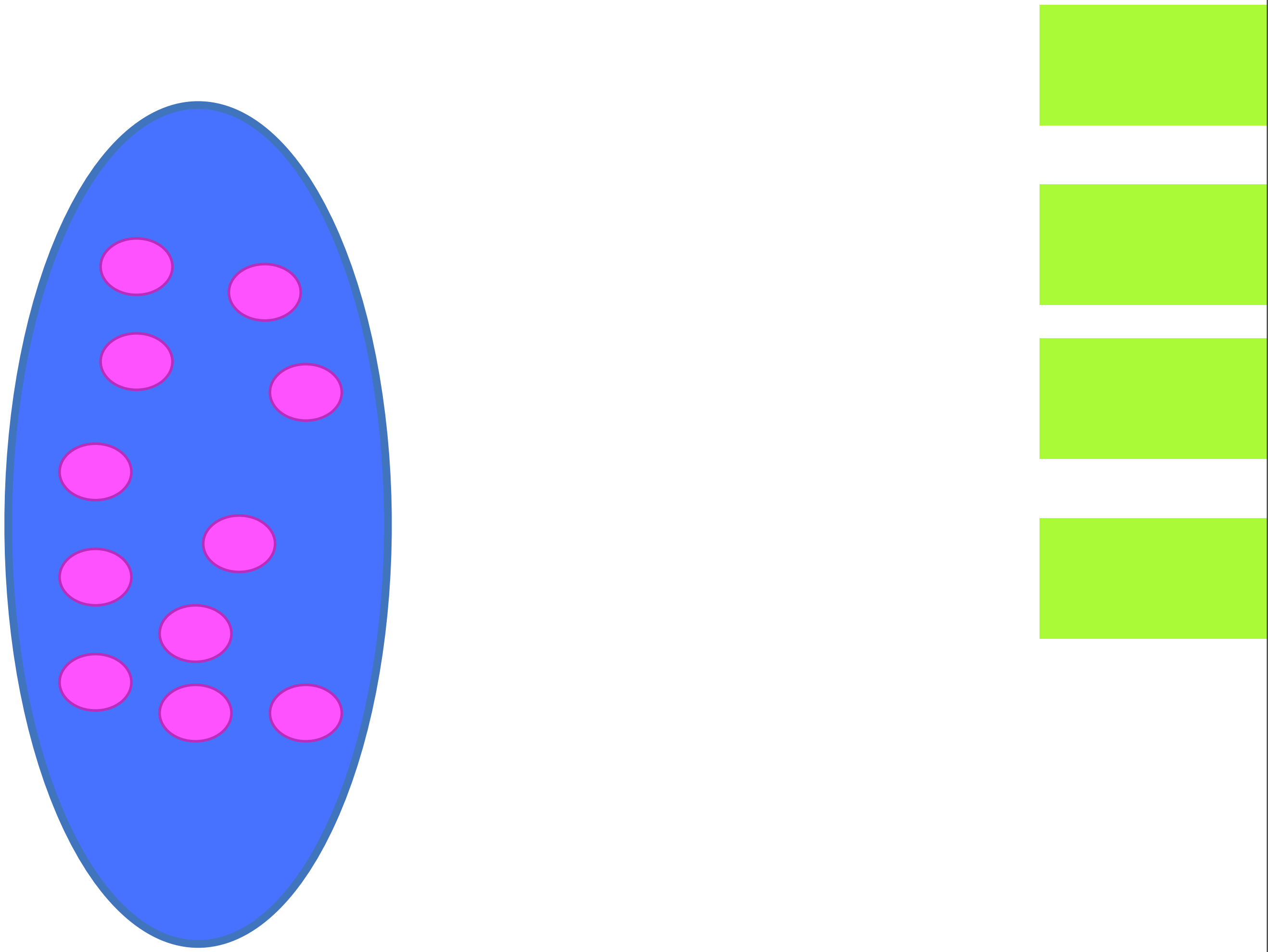
Introduction

- **Torque and MAUI easily the most prominent in EGEE.**
  - MAUI can be used with SGE and LSF as well.
- **Covers,**
  - Maui Priorities, Hard and Soft Limits
  - Maui Reservations
  - Diagnosis.
- **Many new features here, new versions required.**
  - torque > 2.1.6
  - maui > 3.2.6p17
  - This really is about to be released soon to the production grid!

TORQUE and MAUI

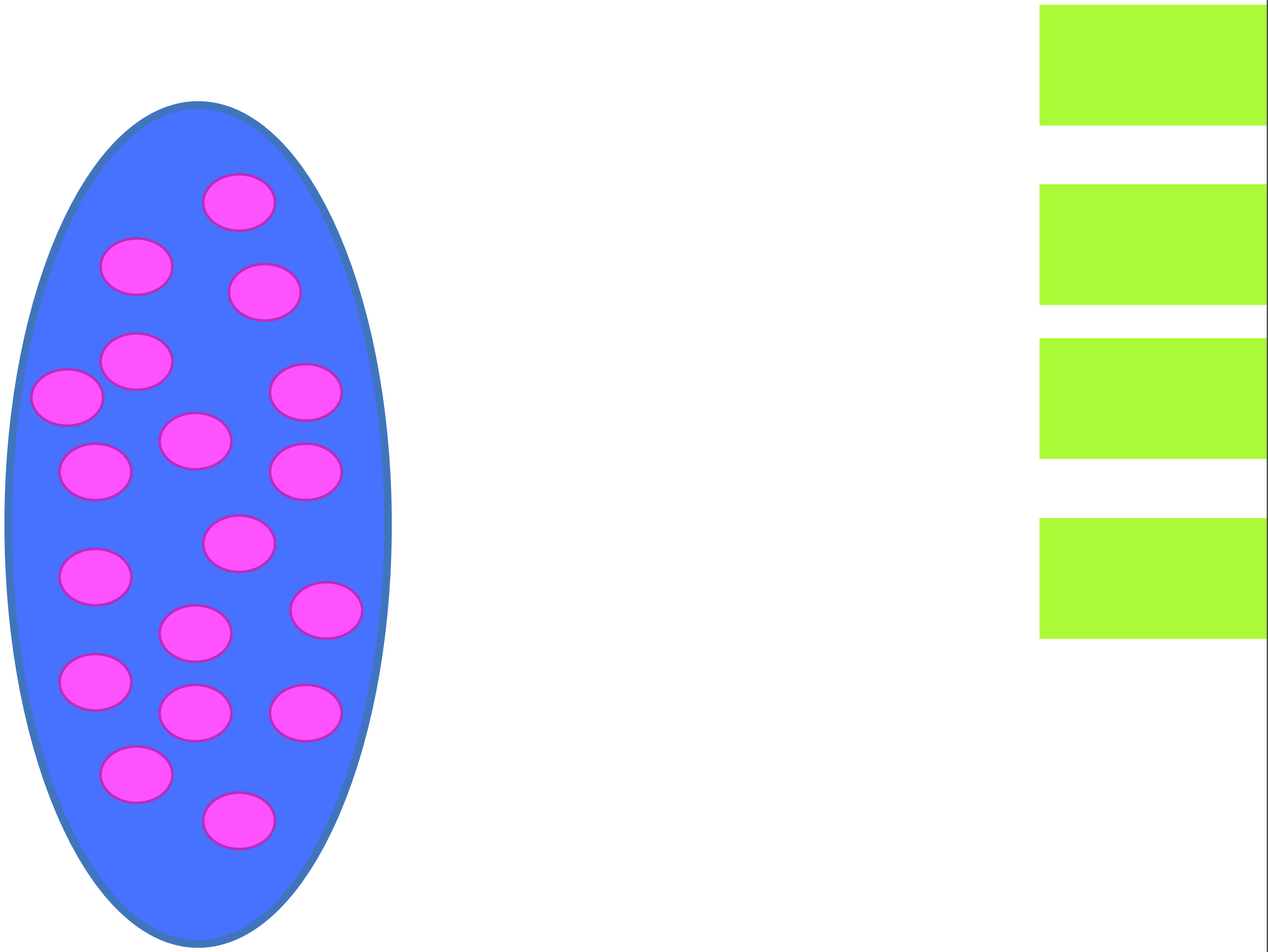
- **What is TORQUE's job as the resource manager.**
  - Accepting and starting jobs across a batch farm.
  - Cancelling jobs.
  - Monitoring the state of jobs.
  - Collecting return codes.
- **What is MAUI's Job?**
  - MAUI makes all the decisions.
  - Should a job be started asking questions like:
  - Is there enough resource to start the job?
  - Given all the jobs I could start which one should I start?
- **MAUI runs a scheduling iteration:**
  - When a job is submitted.
  - When a job ends.
  - At regular configurable intervals.

Scheduling a Job in MAUI



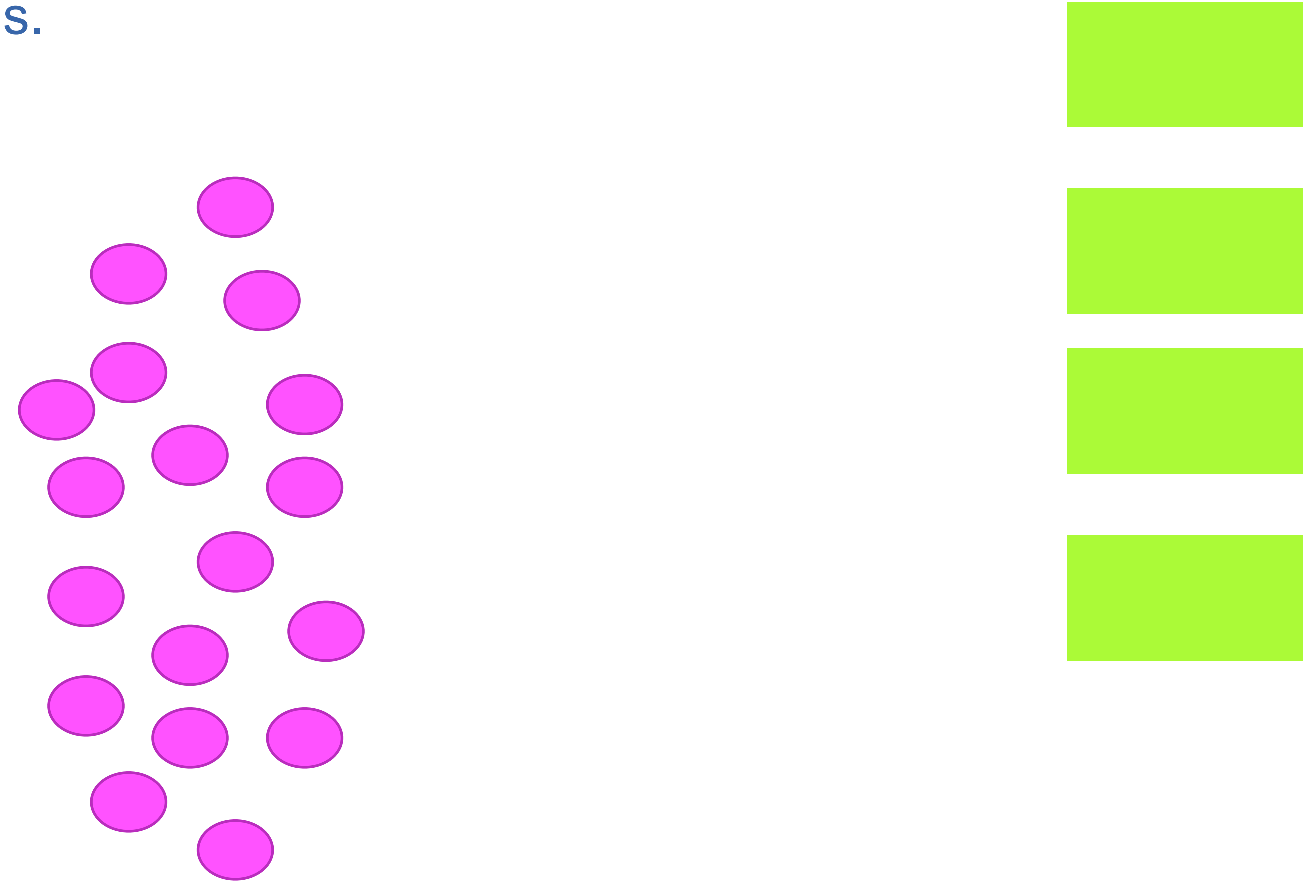
Scheduling a Job in MAUI

- Jobs are submitted into a pool of jobs.
- Forget about queues, MAUI considers all jobs.



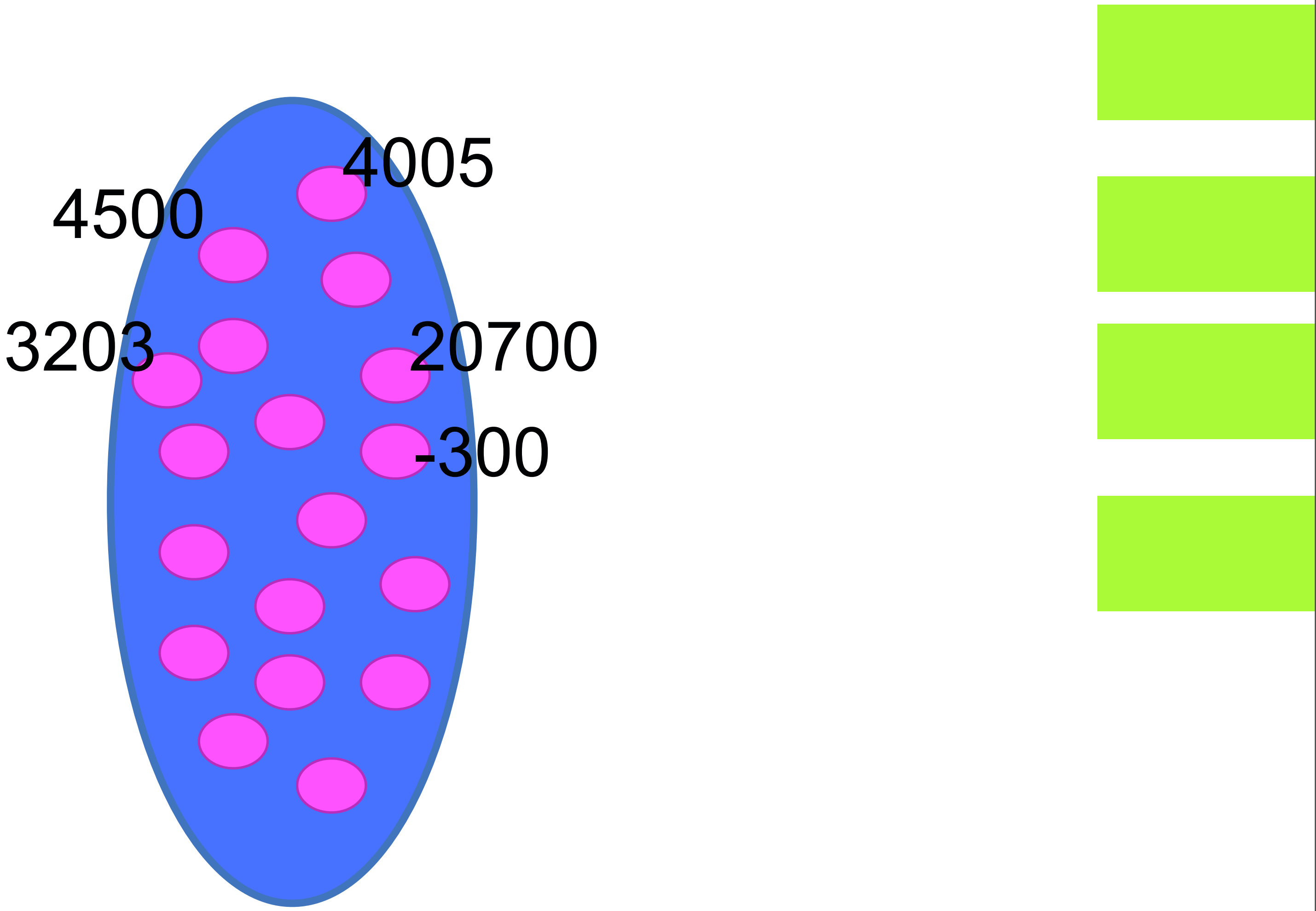
Scheduling a Job in MAUI

- Maui scans through all the jobs and nodes:
  - When a job is submitted.
  - When a job completes.
  - And at periodic intervals.



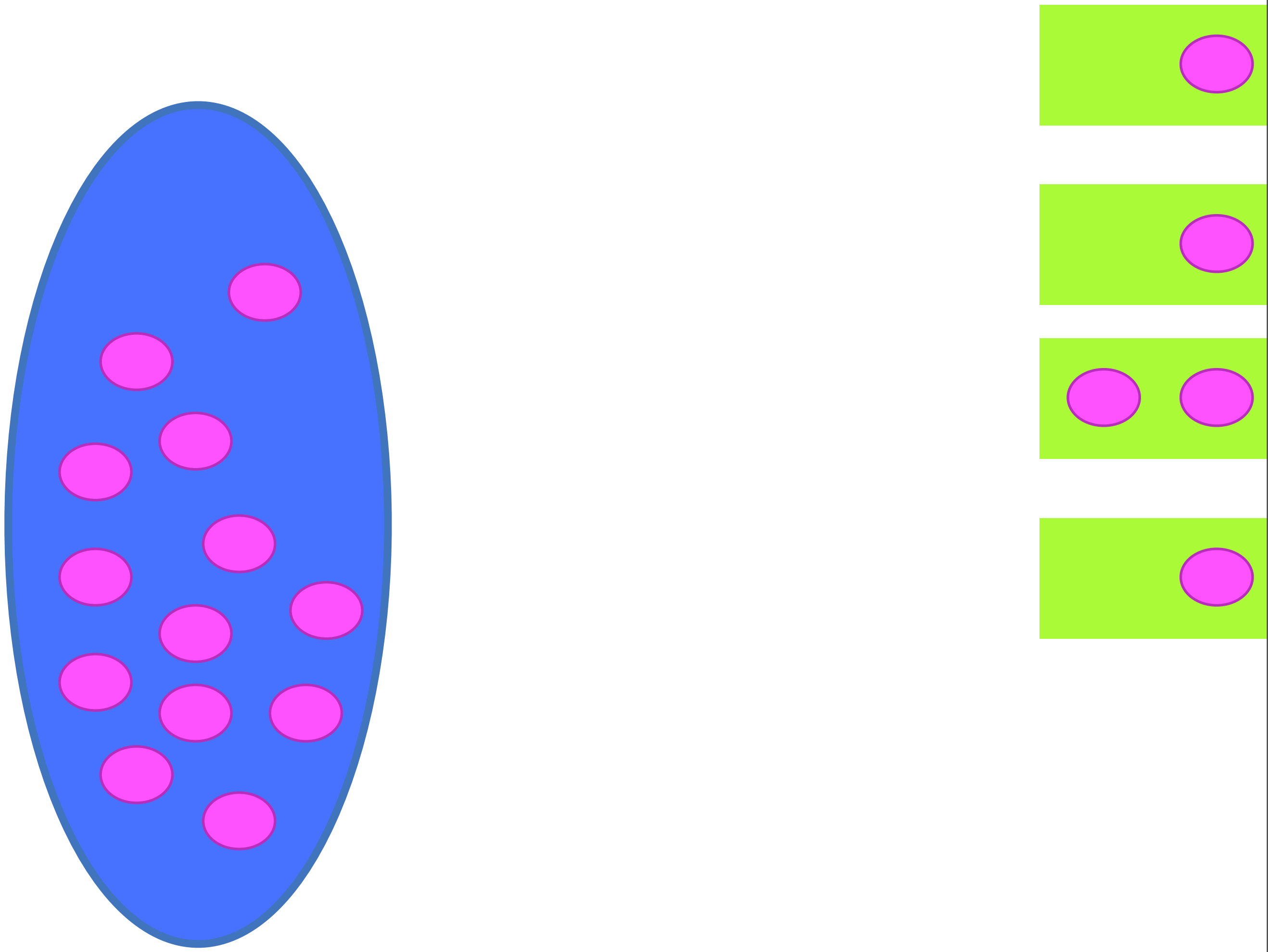
Scheduling a Job in MAUI

- Each job has a priority number calculated.



Scheduling a Job in MAUI

- Each job has a priority number calculated.
- The highest priority is executed first.



Components of a Job's Priority

- A job's priority is made up from components:
  - CRED* = Credentials, e.g user or group name, submission queue, ...
  - FS* = Fairshair, e.g considers historical usage of user, group, ...
  - RES = Resources, e.g. Number of nodes requested, length of job, ..
  - SERV* = Service, e.g Time job has been queued,
  - TARGET = Target, e.g Jobs must run within two days.
  - USAGE = Usage e.g Time consumed by jobs running now.

Components of a Job's Priority

- A job's priority is made up from components:
  - CRED* = Credentials, e.g user or group name, submission queue, ...
  - FS* = Fairshair, e.g considers historical usage of user, group, ...
  - RES = Resources, e.g. Number of nodes requested, length of job, ..
  - SERV* = Service, e.g Time job has been queued,
  - TARGET = Target, e.g Jobs must run within two days.
  - USAGE = Usage e.g Time consumed by jobs running now.
- Each component is weighted and summed to form the priority,

PRIORITY = CREDWEIGHT * (CREDComp) + FSWEIGHT * (FSComp) + ...

- A common mistake is to leave say FSWEIGHT at 0 having configured FS.
- Components, e.g. CREDComp are made up of SubComponents Will only look at *s today.

Subcomponents of CREDComp

- CRED components are static contributions to the overall priority number. e.g username, groupname, submission queue.

Config Attribute	Value	Summary
CREDWEIGHT	10	Component Weight
USERWEIGHT	20	SubComp’ Weight
USERCFG[straylen]	PRIORITY=1000	Static Priority for Me.
CLASSWEIGHT	5	SubComp’ Weight
CLASSCFG[short]	PRIORITY=10000	Static Priority for short Queue

Subcomponents of CREDComp

- CRED components are static contributions to the overall priority number. e.g username, groupname, submission queue.

Config Attribute	Value	Summary
CREDWEIGHT	10	Component Weight
USERWEIGHT	20	SubComp' Weight
USERCFG[straylen]	PRIORITY=1000	Static Priority for Me.
CLASSWEIGHT	5	SubComp' Weight
CLASSCFG[short]	PRIORITY=10000	Static Priority for short Queue

PRIORITY = CREDWEIGHT * (CREDComp) + FSWEIGHT * (FSComp) + ...

Subcomponents of CREDComp

- CRED components are static contributions to the overall priority number. e.g username, groupname, submission queue.

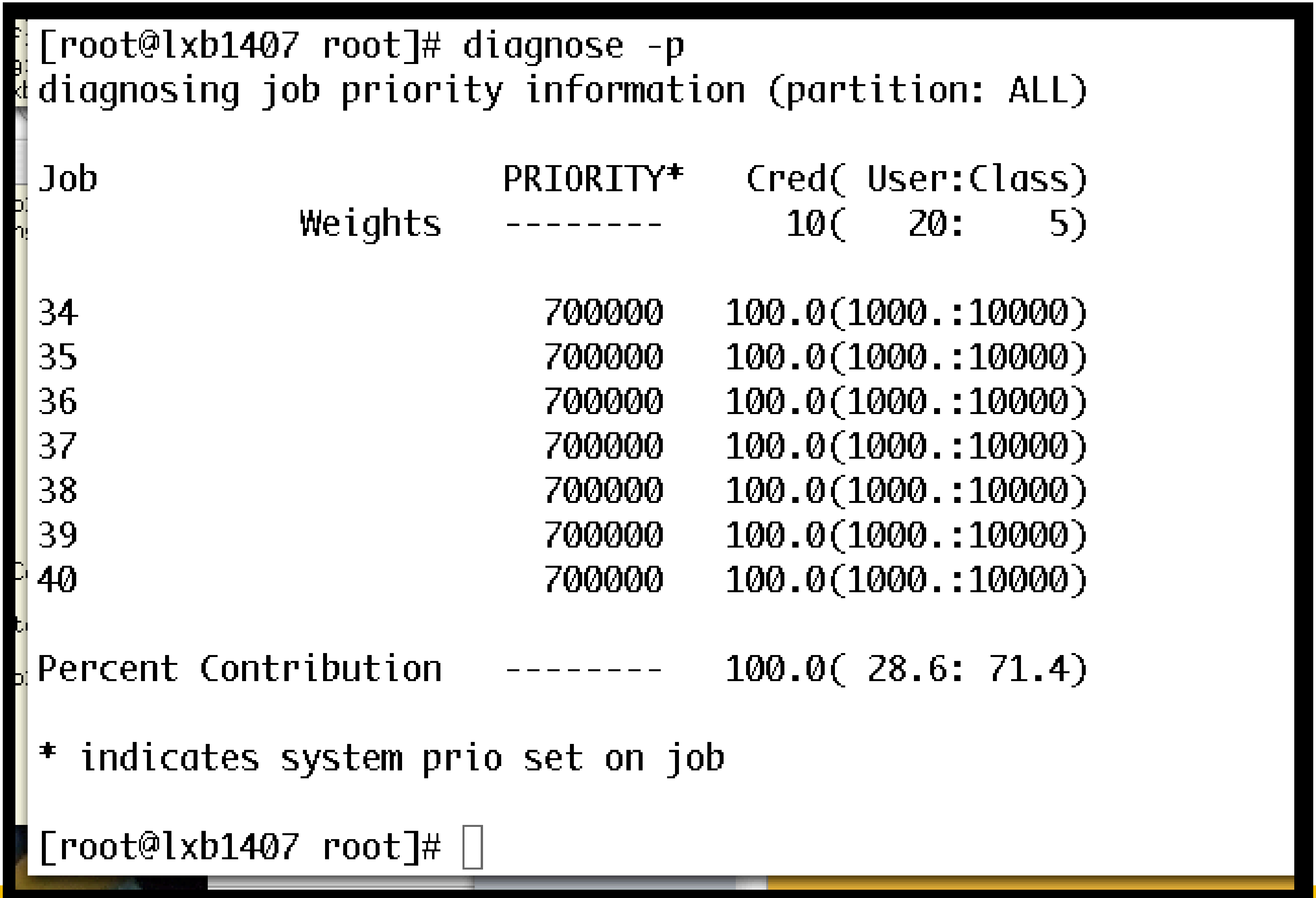
Config Attribute	Value	Summary
CREDWEIGHT	10	Component Weight
USERWEIGHT	20	SubComp' Weight
USERCFG[straylen]	PRIORITY=1000	Static Priority for Me.
CLASSWEIGHT	5	SubComp' Weight
CLASSCFG[short]	PRIORITY=10000	Static Priority for short Queue

PRIORITY = CREDWEIGHT * (CREDComp) + FSWEIGHT * (FSComp) + ...

CREDComp = USERWEIGHT * (USERCFG[straylen] priority)
+ CLASSWEIGHT * (CLASSCFG[short] priority) + ...

Inspecting Job Priorities

- The the “diagnose -p” command is used for this.



```
[root@lxb1407 root]# diagnose -p
diagnosing job priority information (partition: ALL)

Job                     PRIORITY*   Cred( User:Class)
             Weights    --------      10(   20:    5)

34                        700000   100.0(1000.:10000)
35                        700000   100.0(1000.:10000)
36                        700000   100.0(1000.:10000)
37                        700000   100.0(1000.:10000)
38                        700000   100.0(1000.:10000)
39                        700000   100.0(1000.:10000)
40                        700000   100.0(1000.:10000)

Percent Contribution    --------   100.0( 28.6: 71.4)

* indicates system prio set on job

[root@lxb1407 root]# 
```

Subcomponets of FSComponent

- **FS subcomponents consider historical usage of the batch service.**
- **USAGE:**
  - MAUI calculates usage is for each USER, GROUP, CLASS, QOS and ACCOUNT.
- **TARGET**
  - SysAdmin can specify in a TARGET for every USER, GROUP, CLASS, QOS or ACCOUNT.
- **Comparison of USAGE and TARGET.**
  - So for each FSSubComponent e.g. username the used and target values are compared to give a contribution to a queued jobs priority value.

Calculating Usage For A User

- **FSPOLICY=DEDICATEDPS, uses walltime as the metric.**

$$Usage = \frac{\sum_{i=0}^{DEPTH-1}(U_i * DECAY^i)}{\sum_{i=0}^{DEPTH-1}(T_i * DECAY^i)}$$

Calculating Usage For A User

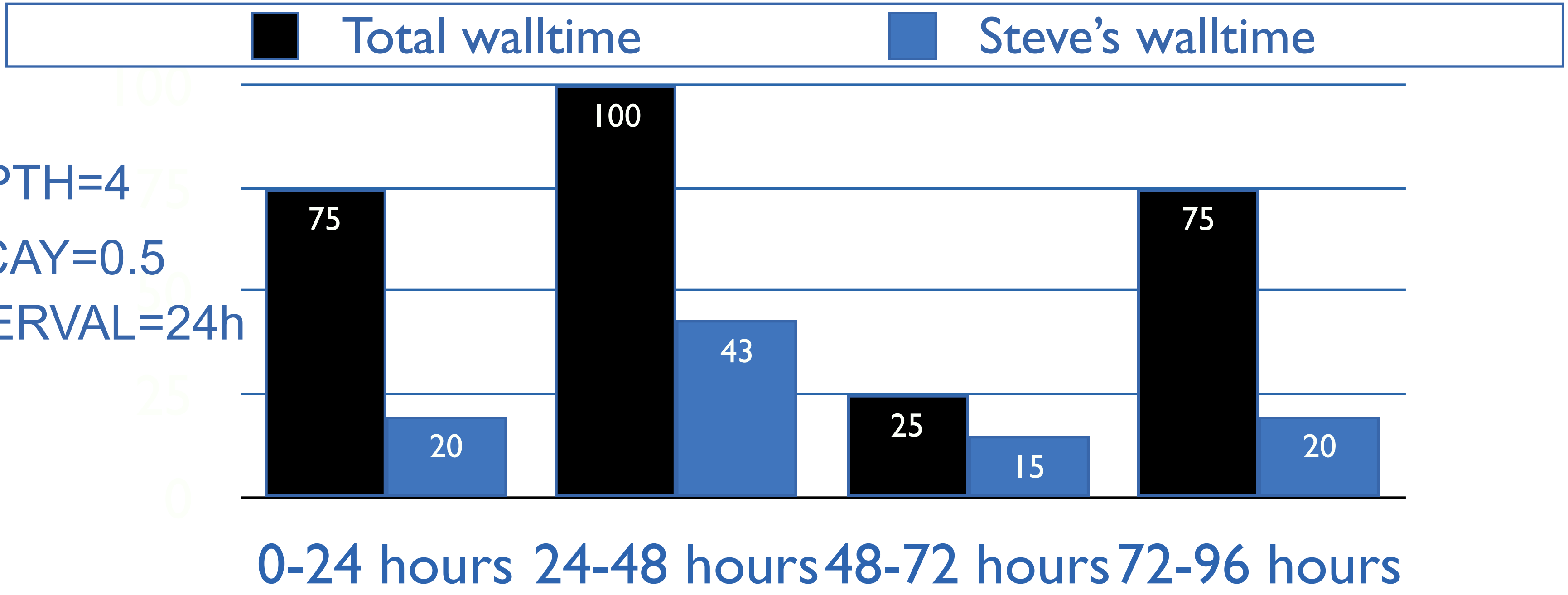
- FSPOLICY=DEDICATEDPS, uses walltime as the metric.

$$Usage = \frac{\sum_{i=0}^{DEPTH-1}(U_i * DECAY^i)}{\sum_{i=0}^{DEPTH-1}(T_i * DECAY^i)}$$

FSDEPTH=4

FSDECAY=0.5

FSINTERVAL=24h



$$USAGE = \frac{20 * 0.5^0 + 43 * 0.5^1 + 15 * 0.5^2 + 20 * 0.5^3}{75 * 0.5^0 + 100 * 0.5^1 + 25 * 0.5^2 + 20 * 0.5^3}$$

Specifying a Target for a User

- For each user, group, class a target can be specified in the configuration.

Config Attribute	Value	Summary
FSWEIGHT	10	Component Weight
FSUSERWEIGHT	20	SubComp’ Weight
USERCFG[straylen]	FSTARGET=1000	FS target for me.
USERCFG[fred]	FSTARGET=500	FS target for Fred.
USERCFG[DEFAULT]	FSTARGET=20	FS target for everyone else.

- Note: The share will be 1000:500:20:20:20:.....
  - Number of users can make a large difference.
  - Solution: Avoid [DEFAULT] ,easy for groups, ...
- Have your FSTARGETS add to 100 if possible.
  - USAGE is reported as a % so diagnosis easier.

Comparing Target and Usage

- A comparison of the target and usage for the user, group or class then gives the contribution to the jobs overall priority.
- There are two configurations for this calculation:
  - Difference - FSPOLICY=DEDICATEDPS is rubbish.
  - Ratio - FSPOLICY=DEDICATEDPS% is much better.

Comparing Target and Usage

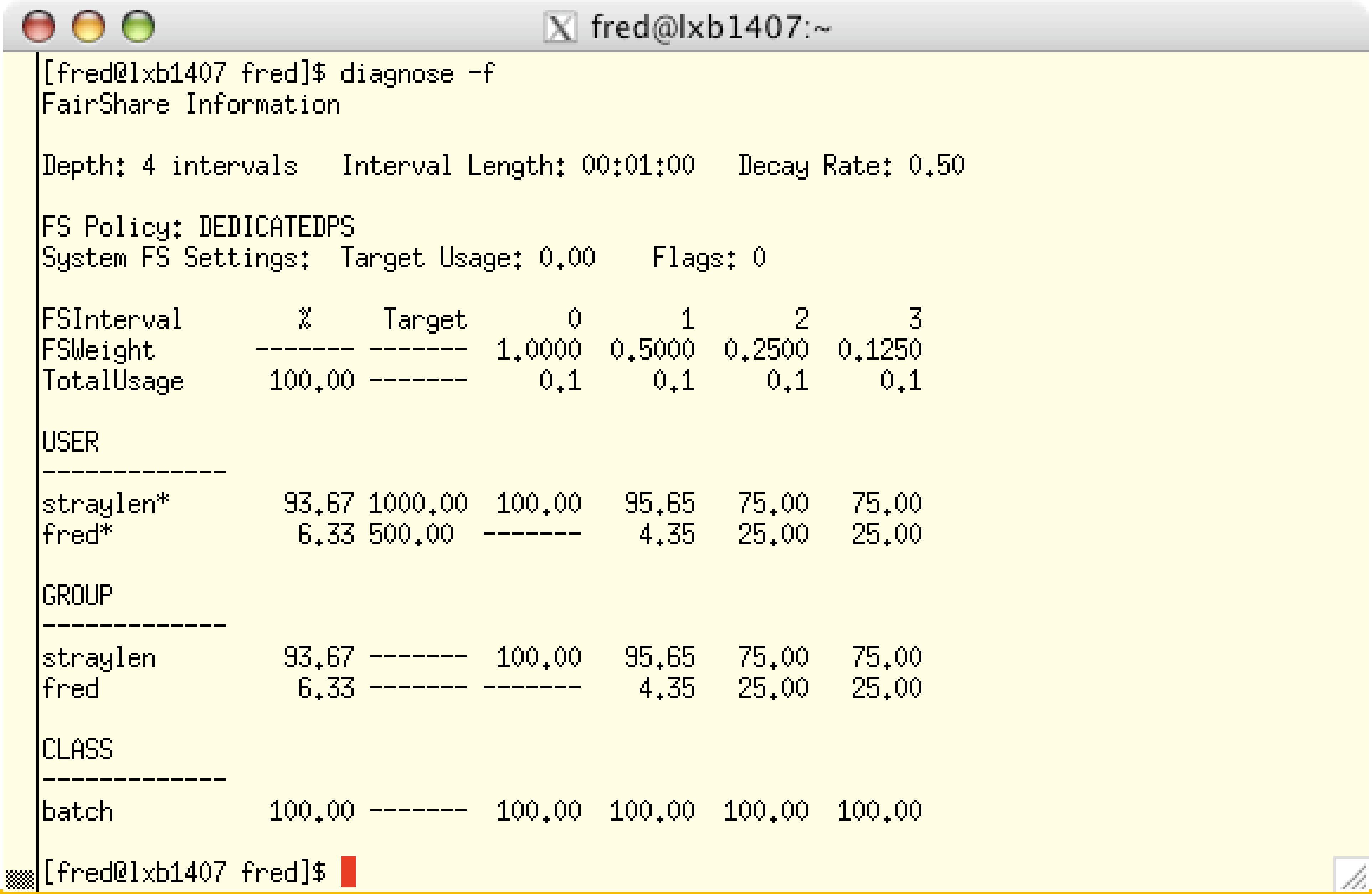
- A comparison of the target and usage for the user, group or class then gives the contribution to the jobs overall priority.
- There are two configurations for this calculation:
  - Difference - FSPOLICY=DEDICATEDPS is rubbish.
  - Ratio - FSPOLICY=DEDICATEDPS% is much better.

PRIORITY = CREDWEIGHT * (CREDComp) + FSWEIGHT * (FSComp) + ...

FSComp = FSUSERWEIGHT * (1 - straylen’s fsusage/straylens’ fstarget)
+ FSGROUPWEIGHT * (1 - dteam’s fsuage/dteam’s fstarget) + ...

Examining Fairshare

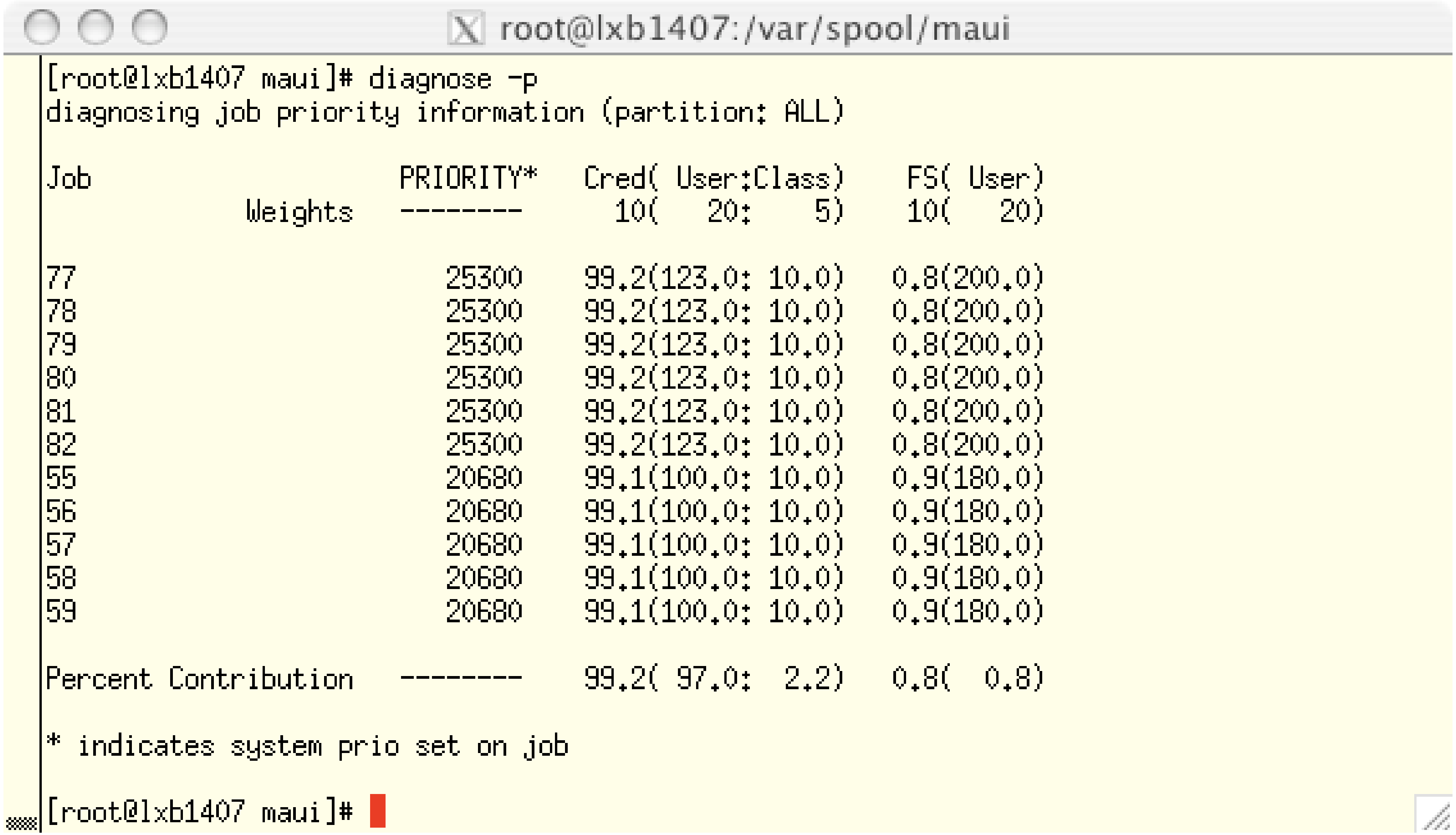
- To interrogate fairshare status use “diagnose -f”.



```
[fred@lxb1407 fred]$ diagnose -f
FairShare Information

Depth: 4 intervals   Interval Length: 00:01:00   Decay Rate: 0.50

FS Policy: DEDICATEDPS
System FS Settings:  Target Usage: 0.00    Flags: 0

FSInterval        %     Target       0       1       2       3
FSWeight       ------- -------  1.0000  0.5000  0.2500  0.1250
TotalUsage      100.00 -------     0.1     0.1     0.1     0.1

USER
-------------
straylen*        93.67 1000.00  100.00   95.65   75.00   75.00
fred*             6.33 500.00  -------    4.35   25.00   25.00

GROUP
-------------
straylen         93.67 -------  100.00   95.65   75.00   75.00
fred              6.33 ------- -------    4.35   25.00   25.00

CLASS
-------------
batch           100.00 -------  100.00  100.00  100.00  100.00

[fred@lxb1407 fred]$ 
```

Comparing Priority Components

- **Now we are using two components, CRED and FS.**
  - The components are in direct competition with another, they must be tuned. Use “diagnose -p” again.



```
[root@lxb1407 maui]# diagnose -p
diagnosing job priority information (partition: ALL)

Job                     PRIORITY*   Cred( User:Class)    FS( User)
             Weights    --------      10(   20:     5)    10(   20)

77                         25300    99.2(123.0: 10.0)   0.8(200.0)
78                         25300    99.2(123.0: 10.0)   0.8(200.0)
79                         25300    99.2(123.0: 10.0)   0.8(200.0)
80                         25300    99.2(123.0: 10.0)   0.8(200.0)
81                         25300    99.2(123.0: 10.0)   0.8(200.0)
82                         25300    99.2(123.0: 10.0)   0.8(200.0)
55                         20680    99.1(100.0: 10.0)   0.9(180.0)
56                         20680    99.1(100.0: 10.0)   0.9(180.0)
57                         20680    99.1(100.0: 10.0)   0.9(180.0)
58                         20680    99.1(100.0: 10.0)   0.9(180.0)
59                         20680    99.1(100.0: 10.0)   0.9(180.0)

Percent Contribution    --------    99.2( 97.0:  2.2)   0.8(  0.8)

* indicates system prio set on job

[root@lxb1407 maui]#
```

MAUI Quality of Service (QOS)

- **Allows us to group types of jobs together based on a credential. Can be queues, users, groups,....**
- **Required for recommendations of job priority working group.**
  - Starting point is jobs are submitted in groups lhcba, lhcbb, lhcbc, cmsa, cmsb, cmsc representing different roles with LHCb and CMS.

```
GROUPCFG[lhcba]   FSTARGET=20   QDEF=qlhcb
GROUPCFG[lhcbb]   FSTARGET=20   QDEF=qlhcb
GROUPCFG[cmsa]    FSTARGET=80   QDEF=qcms
GROUPCFG[cmsb]    FSTARGET=20   QDEF=qcms
QOSCFG[qcms]        FSTARGET=40
QOSCFG[qlhcb]       FSTARGET=60
FSGROUPWEIGHT   100
FSQOSWEIGHT     1000
```

FSComp(cmsa) = FSGROUPWEIGHT * (1 - cmsa’s fsusage/cmsa’ fstarget)
+ FSQOSWEIGHT * (1 - qcms’s fsuage/qcms’s fstarget) + ...

Throttling Policies

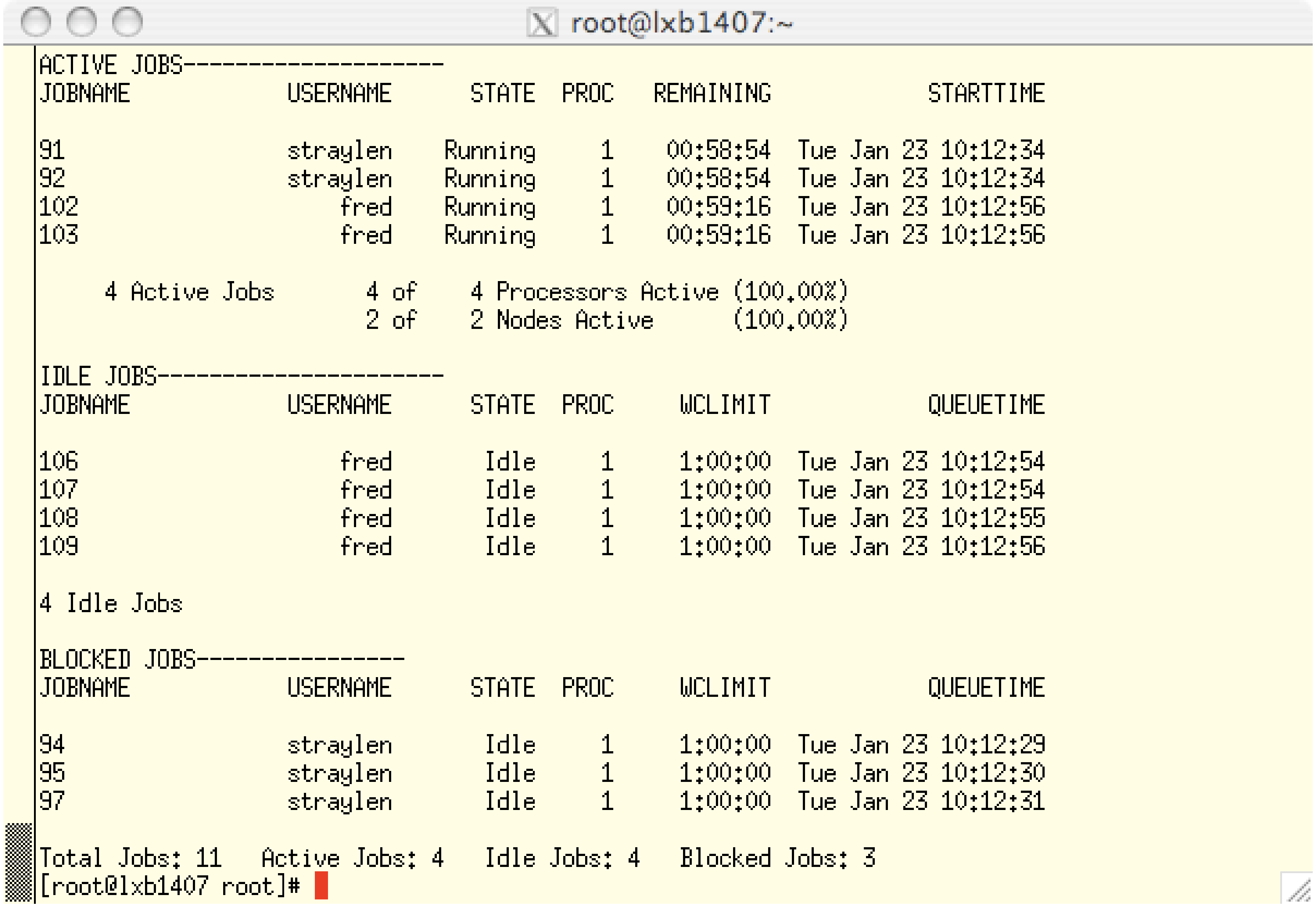
- **Hard Limits**
  - Allow an absolute cap to be introduced for a credential.

Credential	Value	Details
USERCFG[straylen]	MAXJOB=20	Limits me to 20 running os.
GROUPCFG[dteam]	MAXV	mits dteam to only have our of walltime maining.
CLASSCFG[short]	MAXJ	y group can run 5 jobs the short queue.
CLASSCFG[short]	MAXJOB [GROUP:dteam]=10	Group dteam can run 10 jobs in the short queue.

- Jobs can be in three states:
  - RUNNING (on cpu), IDLE (elgible to run), BLOCKED (Non-Eligible)
- Can easily result in idle CPUs , not good......

Examing Queued Job State

- showq is your friend. USERCFG[straylen] MAXJOB=2



```
ACTIVE JOBS--------------------
JOBNAME            USERNAME      STATE  PROC   REMAINING            STARTTIME

91                 straylen    Running     1    00:58:54  Tue Jan 23 10:12:34
92                 straylen    Running     1    00:58:54  Tue Jan 23 10:12:34
102                    fred    Running     1    00:59:16  Tue Jan 23 10:12:56
103                    fred    Running     1    00:59:16  Tue Jan 23 10:12:56

     4 Active Jobs       4 of    4 Processors Active (100.00%)
                         2 of    2 Nodes Active      (100.00%)

IDLE JOBS----------------------
JOBNAME            USERNAME      STATE  PROC     WCLIMIT            QUEUETIME

106                    fred       Idle     1     1:00:00  Tue Jan 23 10:12:54
107                    fred       Idle     1     1:00:00  Tue Jan 23 10:12:54
108                    fred       Idle     1     1:00:00  Tue Jan 23 10:12:55
109                    fred       Idle     1     1:00:00  Tue Jan 23 10:12:56

4 Idle Jobs

BLOCKED JOBS----------------
JOBNAME            USERNAME      STATE  PROC     WCLIMIT            QUEUETIME

94                 straylen       Idle     1     1:00:00  Tue Jan 23 10:12:29
95                 straylen       Idle     1     1:00:00  Tue Jan 23 10:12:30
97                 straylen       Idle     1     1:00:00  Tue Jan 23 10:12:31

Total Jobs: 11   Active Jobs: 4   Idle Jobs: 4   Blocked Jobs: 3
[root@lxb1407 root]#
```

Soft and Hard Limits

Credential	Value	Details
GROUPCFG[atlas]	MAXJOB=2,3	Run a max 2 jobs unless all soft limits are reached.
GROUPCFG[alice]	MAXJOB=3,4	Run a max 3 jobs unless all soft limits are reached.

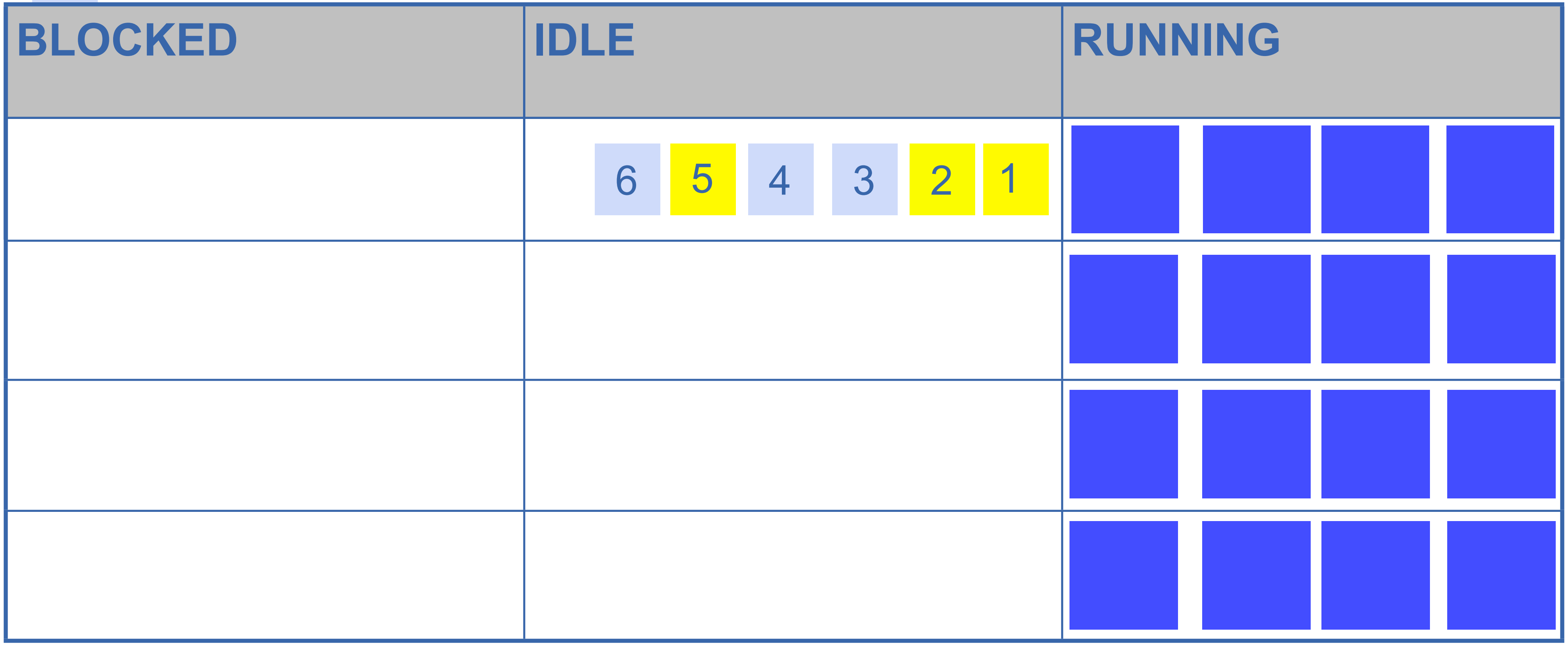
- Soft limits apply unless all soft limits are met.
- Can be used for non historical fairshare.
  - e.g 100 slot farm, MAXJOB=25,1000 will give 25%
- Can be used for offering a basic level of service.
  - e.g 100 slot farm, GROUPCFG[DEFAULT] MAXJOB=10,1000
  - Will block any queued jobs when a group < 10 running.

Example of Soft and Hard Limits

🟨 ATLAS JOB MAXJOB=2,4

▢ CMS JOBs MAXJOB=4,5

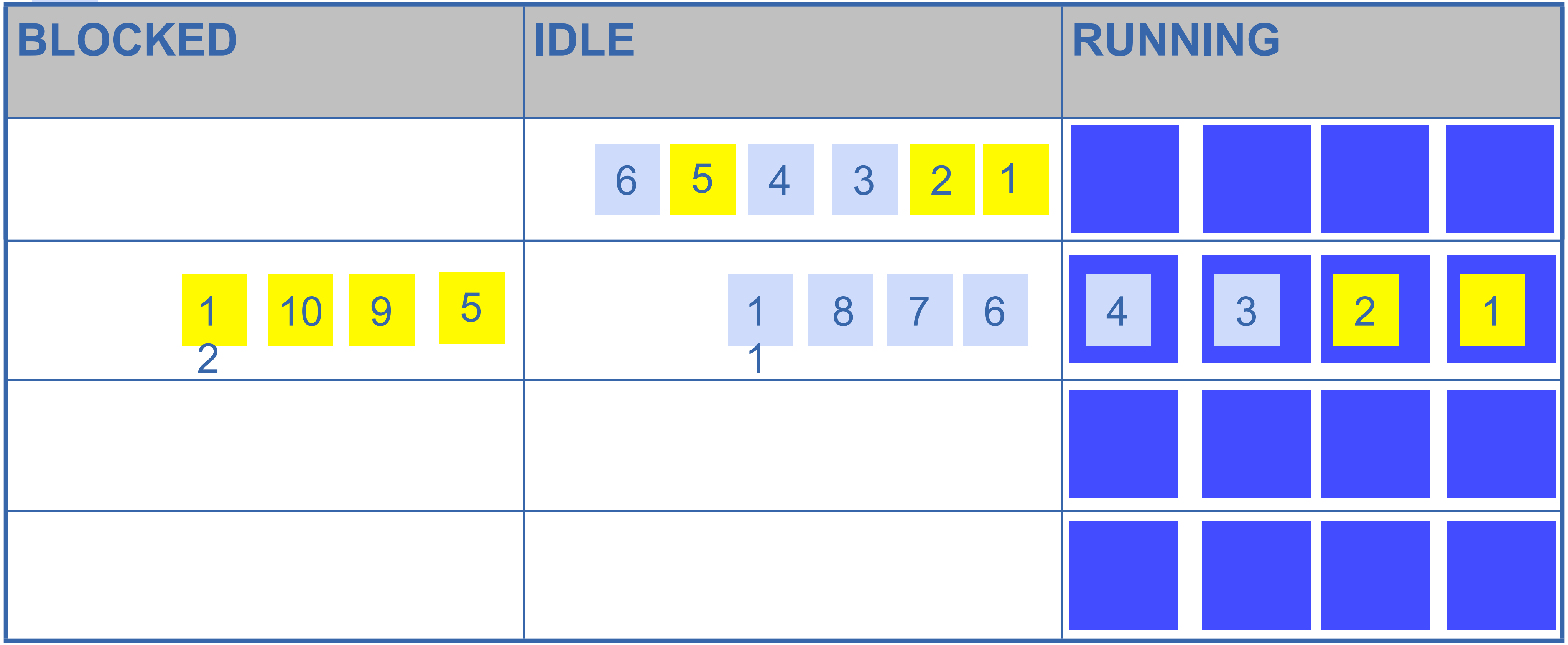
🟦 Job Slot



BLOCKED	IDLE	RUNNING
	6 5 4 3 2 1	🟦 🟦 🟦 🟦
		🟦 🟦 🟦 🟦
		🟦 🟦 🟦 🟦
		🟦 🟦 🟦 🟦

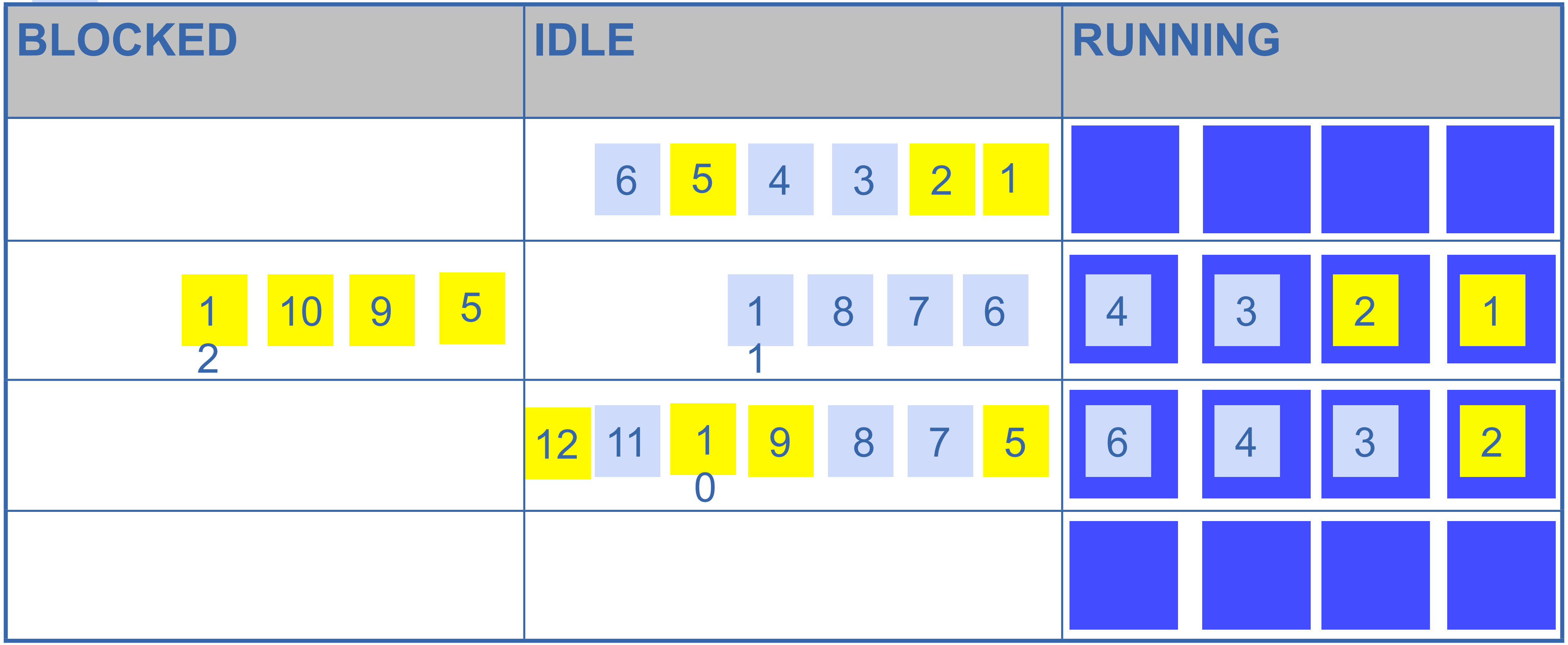
Example of Soft and Hard Limits

🟨	ATLAS JOB	MAXJOB=2,4	🟦	Job Slot
⬜	CMS JOBs	MAXJOB=4,5		



Example of Soft and Hard Limits

🟨	ATLAS JOB	MAXJOB=2,4	🟦	Job Slot
⬜	CMS JOBs	MAXJOB=4,5		



BLOCKED	IDLE	RUNNING
	⬜6 🟨5 ⬜4 ⬜3 🟨2 🟨1	🟦 🟦 🟦 🟦
🟨12 🟨10 🟨9 🟨5	⬜11 ⬜8 ⬜7 ⬜6	⬜4 ⬜3 🟨2 🟨1
	🟨12 ⬜11 🟨10 🟨9 ⬜8 ⬜7 🟨5	⬜6 ⬜4 ⬜3 🟨2
		🟦 🟦 🟦 🟦

Example of Soft and Hard Limits

🟨	ATLAS JOB	MAXJOB=2,4
⬜	CMS JOBs	MAXJOB=4,5
🟦	Job Slot	

BLOCKED	IDLE	RUNNING
	⬜6 🟨5 ⬜4 ⬜3 🟨2 🟨1	🟦 🟦 🟦 🟦
🟨12 🟨10 🟨9 🟨5	⬜11 ⬜8 ⬜7 ⬜6	⬜4 ⬜3 🟨2 🟨1
	🟨12 ⬜11 🟨10 🟨9 ⬜8 ⬜7 🟨5	⬜6 ⬜4 ⬜3 🟨2
🟨11	⬜12	🟨9 ⬜8 ⬜7 🟨5

MAUI Reservations

- Used to reserve particular resources to a certain type of job.
- Reserve a CPU for a queue, say the short one.

```
SRCFG[sdj] HOSTLIST=grid21.lal.in2p3.fr
SRCFG[sdj] PERIOD=INFINITY
SRCFG[sdj] ACCESS=DEDICATED
SRCFG[sdj] TASKCOUNT=1
SRCFG[sdj] RESOURCES=PROCS:1
SRCFG[sdj] CLASSLIST=short
```

- 1 task (slot) is reserved of a size 1 processor.
- The reservation can only be accessed using the short queue(class).
- ACCESS=DEDICATED blocks the slot being used by any jobs not in the short list.
- ACCESS=SHARED allows res’ to be used by others.....?

MAUI Reservations Example

- **Overlaying Jobs. Running say 4 jobs on a 2 CPU node under certain conditions.**
  - e.g. You may want to run monitoring jobs everywhere on top of existing jobs.
  - e.g. System administrators may want their whole farm to stress test their latest dcache.
- **You must lie in TORQUE first. i.e. np=4 for each node.**
  - Any published information needs fixing afterwards.
- **Set up two reservations on each node for two queues.**

```
SRCFG[ad] HOSTLIST=grid21.lal.in2p3.fr
SRCFG[ad] PERIOD=INFINITY
SRCFG[ad] ACCESS=DEDICATED
SRCFG[ad] TASKCOUNT=2
SRCFG[ad] RESOURCES=PROCS:1
SRCFG[ad] CLASSLIST=ops
```

```
SRCFG[lhc] HOSTLIST=grid21.lal.in2p3.fr
SRCFG[lhc] PERIOD=INFINITY
SRCFG[lhc] ACCESS=DEDICATED
SRCFG[lhc] TASKCOUNT=2
SRCFG[lhc] RESOURCES=PROCS:1
SRCFG[lhc] CLASSLIST=atlas,cms
```

Defualt Values

- MAUI and TORQUE both have default values.
- Many of these may need changing.

Tuning MAUI for Large Farms

- **RMPOLLINTERVAL default 60 seconds.**
  - MAUI runs after this time if it has not run.
- **JOBAGGREATIONTIME default 0 seconds.**
  - MAUI will not run within this time of running last time.
- **Specifies the minimum and maximum times between schedule runs.**
- **By default since a MAUI run is triggered at every job submission or completion by TORQUE it will run sequentially for large sites.**
  - Since physics jobs are high rate (single CPU) this should be tuned.

Tuning TORQUE for Large Farms

- **poll_jobs**
  - default is FALSE in current gLite version but now TRUE.
  - Previously a qstat would contact every node to get it's status every time.
  - When TRUE the pbs_server will poll each node periodically to check there status. qstat will not block as a result.
- **job_stat_rate**
  - default is 30 seconds.
  - This is the TTL for the polled information from batch workers.
  - This value should definitely be increased on large farms.
  - SuperCluster vaguely recommends as much as 5 minutes.

Negative Priorities

- **-’ve priorities are by default handled in an “odd” way.**
- **FairShair components include 1 - (used/target)**
  - It is very easy to have a -’ve priority for a job.
- **ENABLENEGJOBPRIORITY default is FALSE**
  - With this seeting -’ve priorities will be reset to 1.
  - This is not what you want, set it to true.
- **REJECTNEGPRIOJOBS default is TRUE**
  - Defines that -’ve priority jobs will never start.
  - This is not what you want, set it to false.

SERVICEWEIGHT is non 0

MAUI config	Default Value	Details
SERVICEWEIGHT	1	Priority Component Weight
QUEUETIMEWEIGHT	1	Sub-Component Weight

- **The only Priority component and sub-component that are not disabled by default.**
- **By default queued jobs increase their priority by**
  - SERVICEWEIGHT * QUEUETIMEWEIGHT * minutes queued *
- **This is the fifo component.**
  - If you have fairshare configured then you may wish to switch this off. i.e. SERVICEWEIGHT=0.

Conclusions

- **Fairshare, Throttling and Reservations are probably enough for LHC jobs.**
  - Multi CPU jobs not covered, e.g. Backfill policies are critical for this.
- **Extra Help**
  - Please submit GGUS tickets.
  - MAUI and Torque mailing lists. http://www.clusterresources.com/pages/resources/mailing-lists.php
  - MAUI and Torque Documentation. http://www.clusterresources.com/pages/resources/documentation.php
  - Purchase MOAB, the commercial version of MAUI.
- **Any Questions**
  - What else is needed to help admins.
  - What other “whacky” configurations do people need?