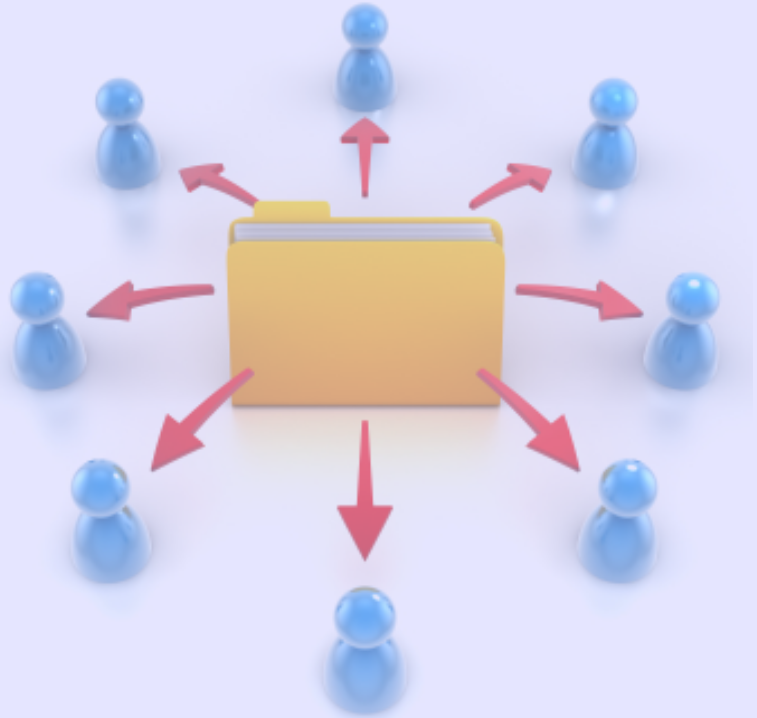
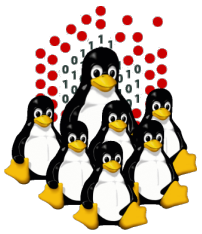
Moreno Baricevic
Stefano Cozzini

CNR-IOM DEMOCRITOS
Trieste, ITALY



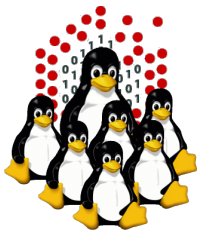
Resource Management



RESOURCE MANAGEMENT

We have a pool of users and a pool of resources, then what?

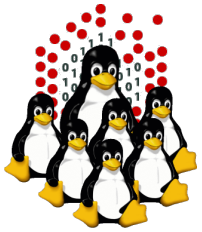
- some software that controls available resources
- some other software that decides which application to execute based on available resources
- some other software devoted to actually execute applications



RESOURCE MANAGEMENT

The resource manager allows:

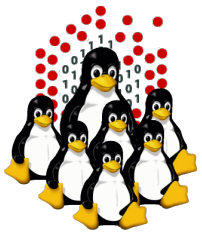
- better resource control
- better resource utilization
- better access control



RESOURCE MANAGEMENT

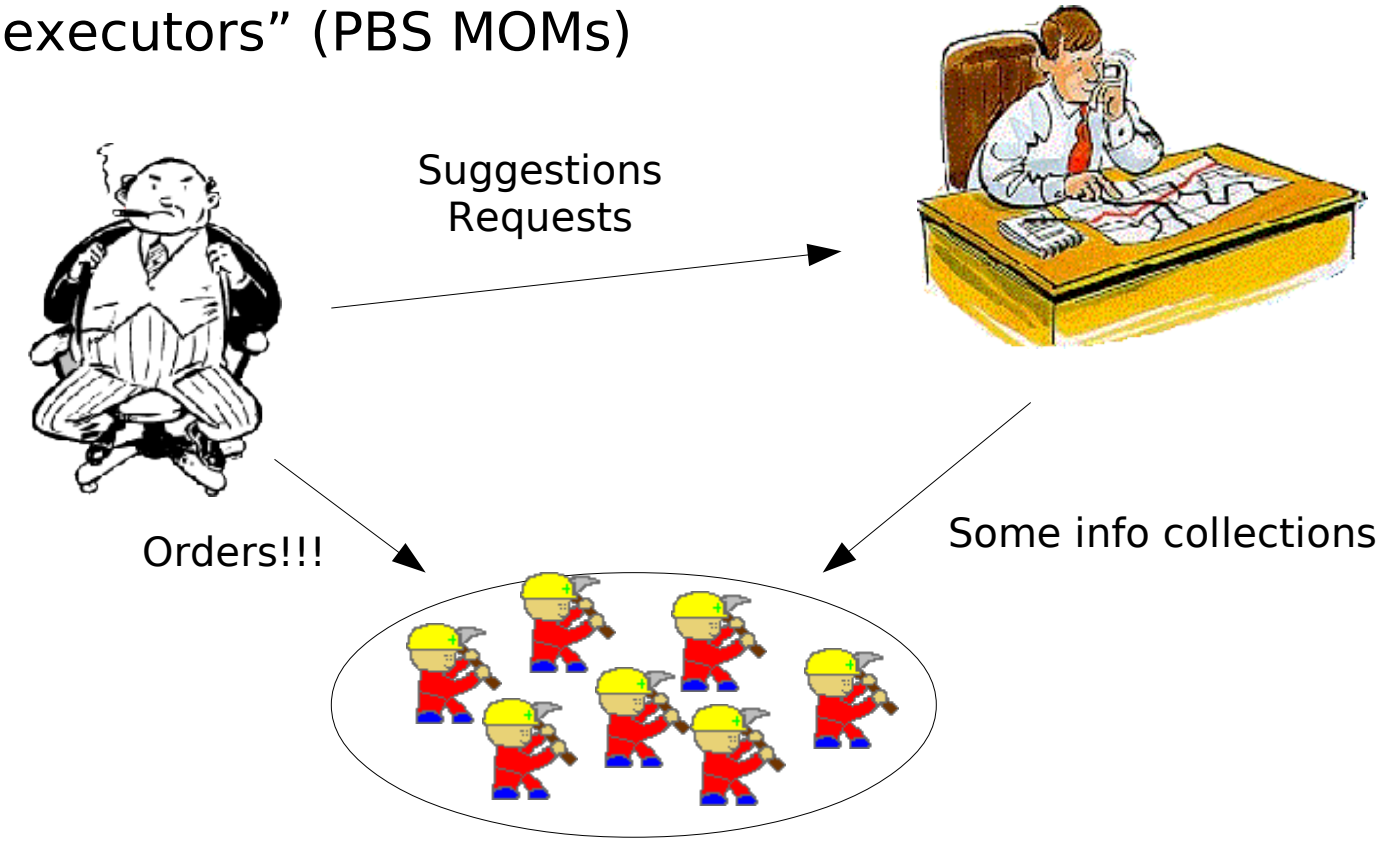
The scheduler should have:

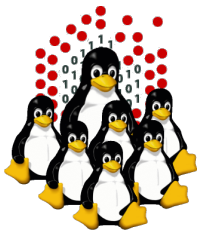
- **Fair Share mechanism**
- **Backfill scheduling algorithm**
- reservations for high priority jobs
- more control parameters on users
- commands for querying the scheduler



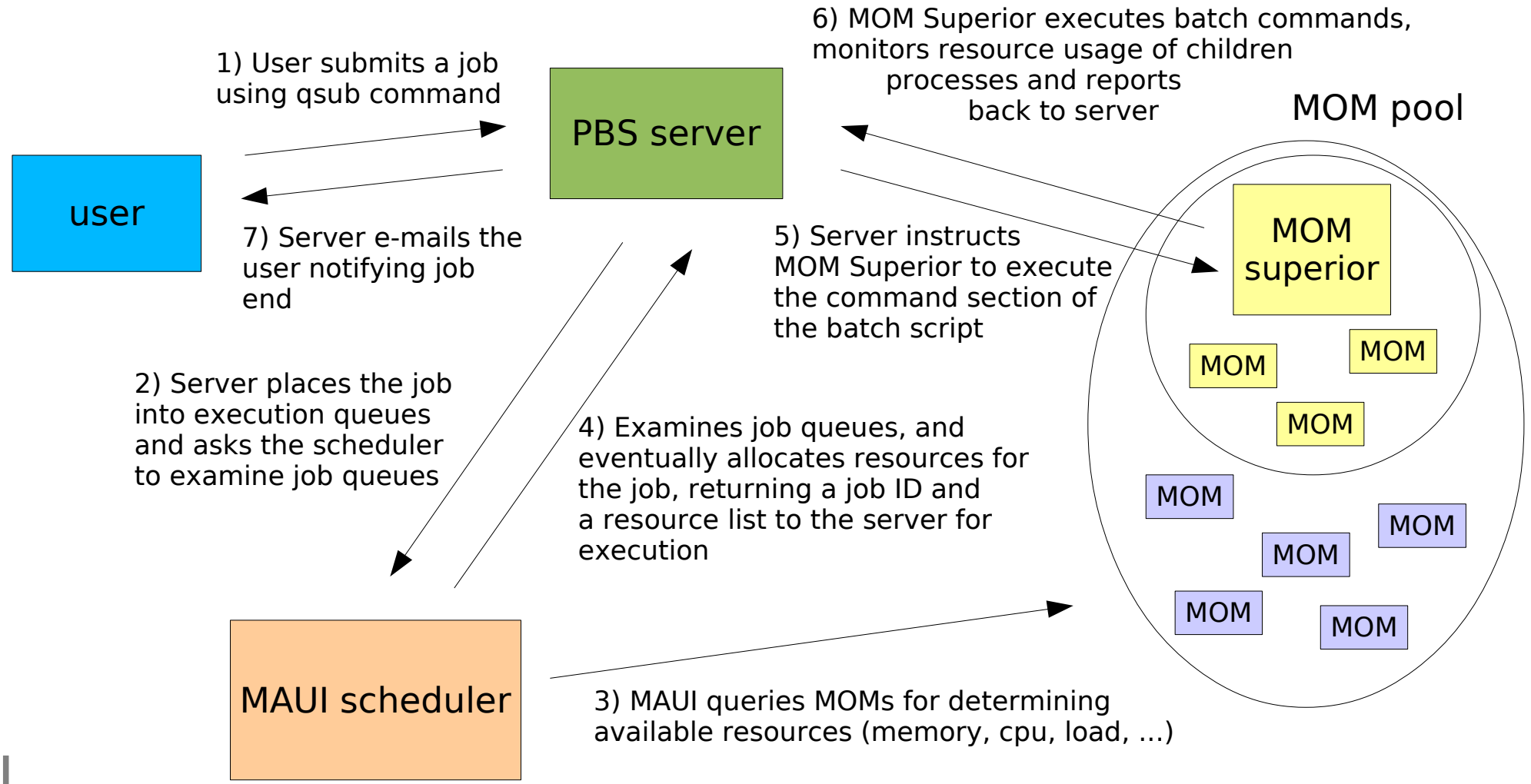
The Queue System - PBS/TORQUE + MAUI

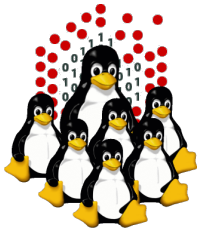
- General Components
  - A resource manager (PBS server)
  - A scheduler (MAUI scheduler)
  - Many “executors” (PBS MOMs)





A typical job session





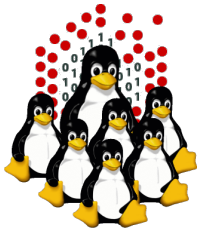
Fair sharing

Fairshare is a mechanism which allows historical resource utilization information to be incorporated into job feasibility and priority decisions.

Fairshare information only affects the job's priority relative to other jobs.

Using the standard fairshare target

- the priority of jobs of a particular group which has used too many resources over the specified fairshare window is lowered
- the priority of jobs which have not received enough resources will be increased



Fair sharing – How it works

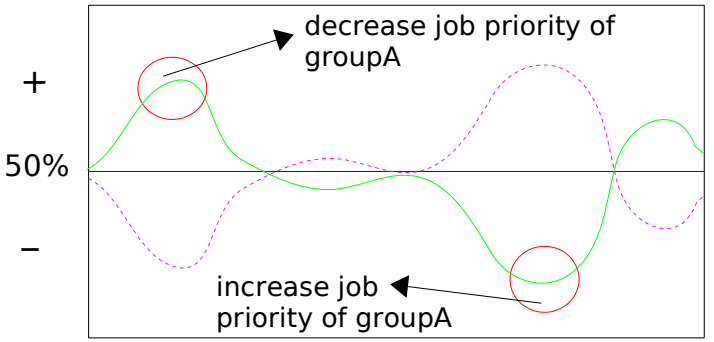
- At the beginning all the jobs are created equals (in term of priority)
- However some jobs are more/less equal than others
- Priority is increased/decreased when the fair sharing quota is below/above from its target
- Gain/lost in priority:
  - is configurable
  - 1% far from fair share means 4 hours on queues (DEMOCRITOS example)

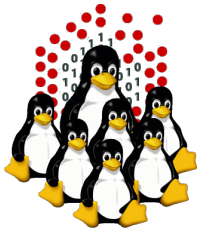
```
GROUPCFG[groupA]    FSTARGET=50%    PRIORITY=5000
GROUPCFG[groupB]    FSTARGET=50%    PRIORITY=5000
```

Assume groupA has 50% of fairshare usage.
When it uses more resources than those assigned, the priority of the jobs will be decreased; when it uses less resources, the priority of its jobs will be increased.

When a group is not computing, the other groups can benefit from the available resources

- better resource utilization
  - no idle CPUs





Backfill 1/2

Backfill is a scheduling optimization which allows a scheduler to make better use of available resources by running jobs out of order.

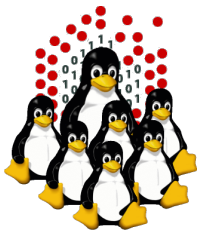
Consider this example with a 10 CPUs machine:

Job1 (priority=20 walltime=10 nodes=6)
Job2 (priority=50 walltime=30 nodes=4)
Job3 (priority=40 walltime=20 nodes=4)
Job4 (priority=10 walltime=10 nodes=1)

1) When Maui schedules, it prioritizes the jobs in the queue according to a number of factors and then re-orders the jobs into a 'highest priority first' sorted list.

Sorted list:

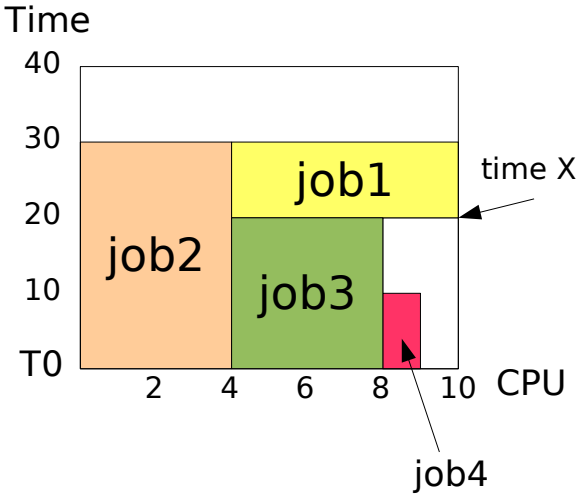
Job2 (priority=50 walltime=30 nodes=4)
Job3 (priority=40 walltime=20 nodes=4)
Job1 (priority=20 walltime=10 nodes=6)
Job4 (priority=10 walltime=10 nodes=1)



Backfill 2/2

2) It starts the jobs one by one stepping through the priority list until it reaches a job which it cannot start.

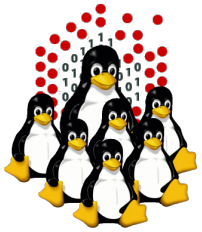
3) All jobs and reservations have a start time and a walltime limit, so MAUI can determine:
- the completion time of all jobs in the queue
- the earliest the needed resources will become available for the highest priority job to start (time X)
- which jobs can be started without delaying this job (job4)



Job2 (priority=50 walltime=30 nodes=4)
Job3 (priority=40 walltime=20 nodes=4)
Job1 (priority=20 walltime=10 nodes=6)
Job4 (priority=10 walltime=10 nodes=1)

➔ Enabling backfill allows the scheduler to start other, lower-priority jobs so long as they do not delay the highest priority job, essentially filling in holes in node space.

➔ Backfill offers significant scheduler performance improvement:
- increased system utilization by around 20% and improved turnaround time by an even greater amount in a typical large system
- backfill tends to favor smaller and shorter running jobs more than larger and longer running ones: It is common to see over 90% of these small and short jobs backfilled.



Questions?

Any?