Moab, TORQUE, and Gold in a Heterogeneous, Federated Computing System at the University of Michigan

Andrew Caird
Matthew Britt Brock Palen

September 18, 2009

MichiganEngineering

Who We Are

- College of Engineering "centralized" HPC support
- Been trying this for 15+ years
- We aren't the College of Literature, Sciences, and Arts; we aren't the Medical School; we aren't the Department of Astronomy; we aren't any of the other 15 schools or colleges; although on Saturdays in the Fall, we are one University
- We are three full-time employees, one student employee, and much support from Engineering Central IT

Who We Are

- College of Engineering "centralized" HPC support
- Been trying this for 15+ years
- We aren't the College of Literature, Sciences, and Arts; we aren't the Medical School; we aren't the Department of Astronomy; we aren't any of the other 15 schools or colleges; although on Saturdays in the Fall, we are one University
- We are three full-time employees, one student employee, and much support from Engineering Central IT

Who We Are

- College of Engineering "centralized" HPC support
- Been trying this for 15+ years
- We aren't the College of Literature, Sciences, and Arts; we aren't the Medical School; we aren't the Department of Astronomy; we aren't any of the other 15 schools or colleges; although on Saturdays in the Fall, we are one University
- We are three full-time employees, one student employee, and much support from Engineering Central IT

Who We Are

- College of Engineering "centralized" HPC support
- Been trying this for 15+ years
- We aren't the College of Literature, Sciences, and Arts; we aren't the Medical School; we aren't the Department of Astronomy; we aren't any of the other 15 schools or colleges; although on Saturdays in the Fall, we are one University
- We are three full-time employees, one student employee, and much support from Engineering Central IT

Who We Are

- College of Engineering "centralized" HPC support
- Been trying this for 15+ years
- We aren't the College of Literature, Sciences, and Arts; we aren't the Medical School; we aren't the Department of Astronomy; we aren't any of the other 15 schools or colleges; although on Saturdays in the Fall, we are one University
- We are three full-time employees, one student employee, and much support from Engineering Central IT

Who We Are

- College of Engineering "centralized" HPC support
- Been trying this for 15+ years
- We aren't the College of Literature, Sciences, and Arts; we aren't the Medical School; we aren't the Department of Astronomy; we aren't any of the other 15 schools or colleges; although on Saturdays in the Fall, we are one University
- We are three full-time employees, one student employee, and much support from Engineering Central IT

What We Support

- 3,488 cores in 664 systems
- 32 hardware owners
- 450+ unique users over the past 6 months
- 73TB Lustre storage
- 74 unique software titles, 127 versions, 14 license restricted
- 9 Tesla S1070s with 4 GPUs each
- 100 Infiniband-connected nodes in 4 switches
- 2 architectures: Opteron and Xeon
- 19 individual CPU types based on clock speed and core count (15 Opteron, 4 Xeon)
- and some other stuff: SGI Altix with 32 cores of Itanium and an Apple XServe cluster with 400 cores of G5 (that's two more architectures)

What We Support

- 3,488 cores in 664 systems
- 32 hardware owners
- 450+ unique users over the past 6 months
- 73TB Lustre storage
- 74 unique software titles, 127 versions, 14 license restricted
- 9 Tesla S1070s with 4 GPUs each
- 100 Infiniband-connected nodes in 4 switches
- 2 architectures: Opteron and Xeon
- 19 individual CPU types based on clock speed and core count (15 Opteron, 4 Xeon)
- and some other stuff: SGI Altix with 32 cores of Itanium and an Apple XServe cluster with 400 cores of G5 (that's two more architectures)

What We Support

- 3,488 cores in 664 systems
- 32 hardware owners
- 450+ unique users over the past 6 months
- 73TB Lustre storage
- 74 unique software titles, 127 versions, 14 license restricted
- 9 Tesla S1070s with 4 GPUs each
- 100 Infiniband-connected nodes in 4 switches
- 2 architectures: Opteron and Xeon
- 19 individual CPU types based on clock speed and core count (15 Opteron, 4 Xeon)
- and some other stuff: SGI Altix with 32 cores of Itanium and an Apple XServe cluster with 400 cores of G5 (that's two more architectures)

How Do We Do It?

Torque, Gold, and Moab

(surprise)

How Do We Do It?

Torque, Gold, and Moab

(surprise)

Torque

Our Torque set-up is pretty plain:

- we assign properties to nodes
- we rely a lot on a healthcheck script to monitor:
  - local disk space and filesystem state (checking for read-only)
  - NFS, Lustre, and AFS mounts
  - Infiniband connectivity for nodes with IB
  - Out-of-memory warnings
  - sshd dying
- we sometimes run a pro- or epilogue script
- we monitor disk to support job requests for local disk space

Torque

Our Torque set-up is pretty plain:

- we assign properties to nodes
- we rely a lot on a healthcheck script to monitor:
  - local disk space and filesystem state (checking for read-only)
  - NFS, Lustre, and AFS mounts
  - Infiniband connectivity for nodes with IB
  - Out-of-memory warnings
  - sshd dying
- we sometimes run a pro- or epilogue script
- we monitor disk to support job requests for local disk space

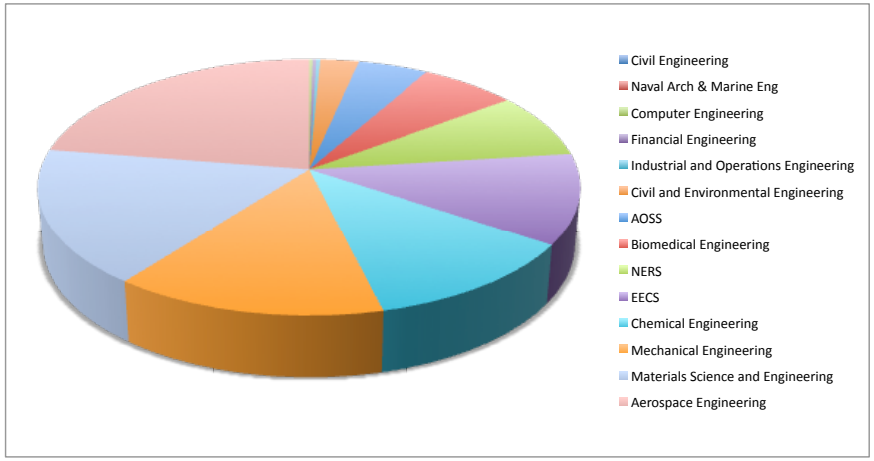
Torque

Our Torque set-up is pretty plain:

- we assign properties to nodes
- we rely a lot on a healthcheck script to monitor:
  - local disk space and filesystem state (checking for read-only)
  - NFS, Lustre, and AFS mounts
  - Infiniband connectivity for nodes with IB
  - Out-of-memory warnings
  - sshd dying
- we sometimes run a pro- or epilogue script
- we monitor disk to support job requests for local disk space

Gold

- We only use Gold for collecting accounting data, not setting policy.
- We allow Gold to auto-create accounts, then we have a manual process (named Matthew) that fills in our local data, like Name, Department, College, Adviser, etc.
- We have developed a handful of scripts to pull together Gold data for internal consumption and presentation.



Moab

To manage our environment, we use:

- standing reservations
- quality of service settings
- accounts
- node sets
- Unix groups
- CPU speed
- rollback reservations
- fairshare
- preemption
- node features from Torque

Policies

We use Moab to represent our policies, the first level of policy is:

- jobs from hardware owners should use their hardware first, overflowing to public nodes if job requirements can be met
- if hardware is idle, anyone can use it as long they agree to be preempted
- jobs can "overflow" from owned nodes to "public" nodes
- no one can use more than 32 cores, plus whatever they own
  - unless they are using preemption, then they can use 196 cores
    - unless they aren't Engineers, then each user constrained to a pool of 32 total cores

Policies

We use Moab to represent our policies, the first level of policy is:

- jobs from hardware owners should use their hardware first, overflowing to public nodes if job requirements can be met
- if hardware is idle, anyone can use it as long they agree to be preempted
- jobs can "overflow" from owned nodes to "public" nodes
- no one can use more than 32 cores, plus whatever they own
  - unless they are using preemption, then they can use 196 cores
    - unless they aren't Engineers, then each user constrained to a pool of 32 total cores

Policies

We use Moab to represent our policies, the first level of policy is:

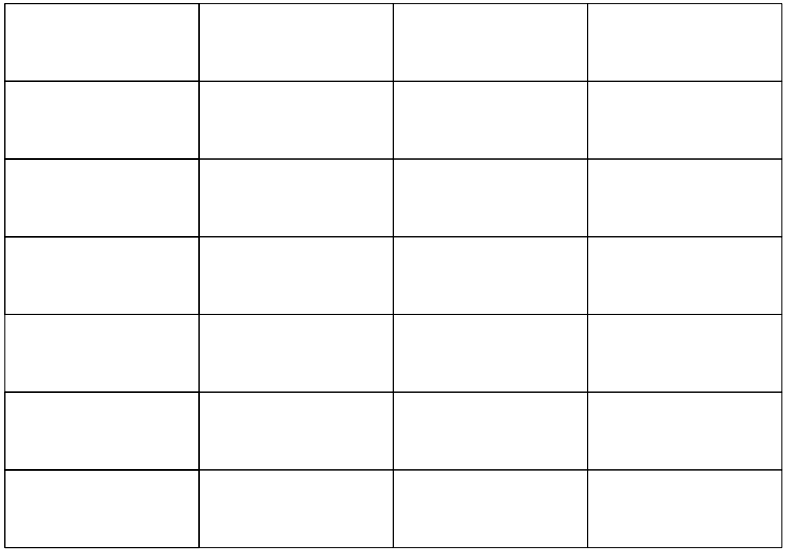
- jobs from hardware owners should use their hardware first, overflowing to public nodes if job requirements can be met
- if hardware is idle, anyone can use it as long they agree to be preempted
- jobs can "overflow" from owned nodes to "public" nodes
- no one can use more than 32 cores, plus whatever they own
  - unless they are using preemption, then they can use 196 cores
    - unless they aren't Engineers, then each user constrained to a pool of 32 total cores

Moab config

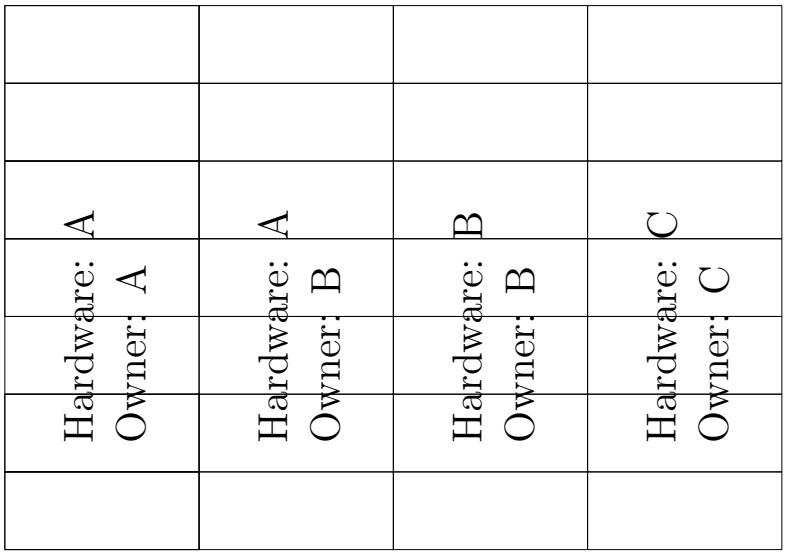
Our simplest case is an owner, a set of nodes, and a set of users, which we configure like this:

```
ACCOUNTCFG[mikehart]
MEMBERULIST=adamvh,ajhunte,[...],mikehart,[...] QDEF=mikehart QLIST=mikehart,cac,preempt
QOSCFG[mikehart]        MAXPROC[USER]=64
SRCFG[mikehart]         ACCOUNTLIST=mikehart+,cacstaff
SRCFG[mikehart]         QOSLIST=~preempt
SRCFG[mikehart]         HOSTLIST=nyx0590,nyx0591,nyx0592,nyx0593,nyx0594,nyx0595,nyx0596,nyx0597
SRCFG[mikehart]         OWNER=ACCT:mikehart
SRCFG[mikehart]         PERIOD=INFINITY
SRCFG[mikehart]         FLAGS=IGNSTATE,OWNERPREEMPT
```

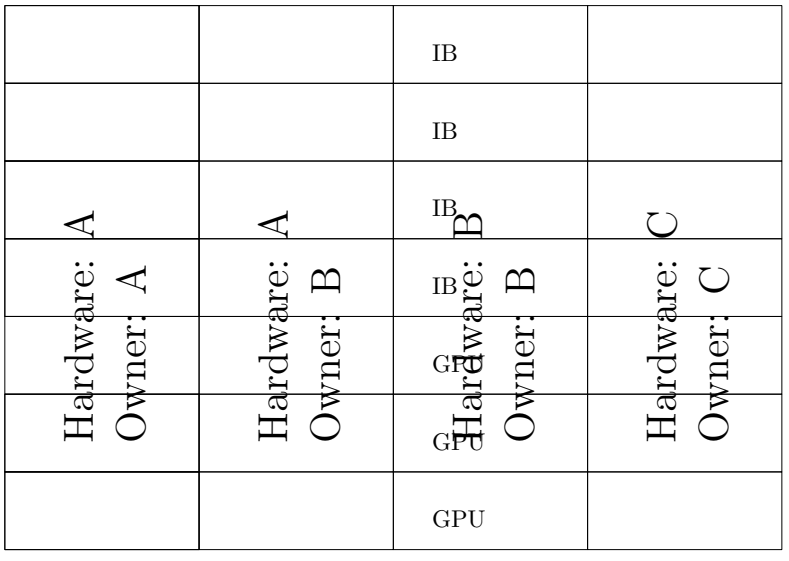
Hardware that Moab must Understand



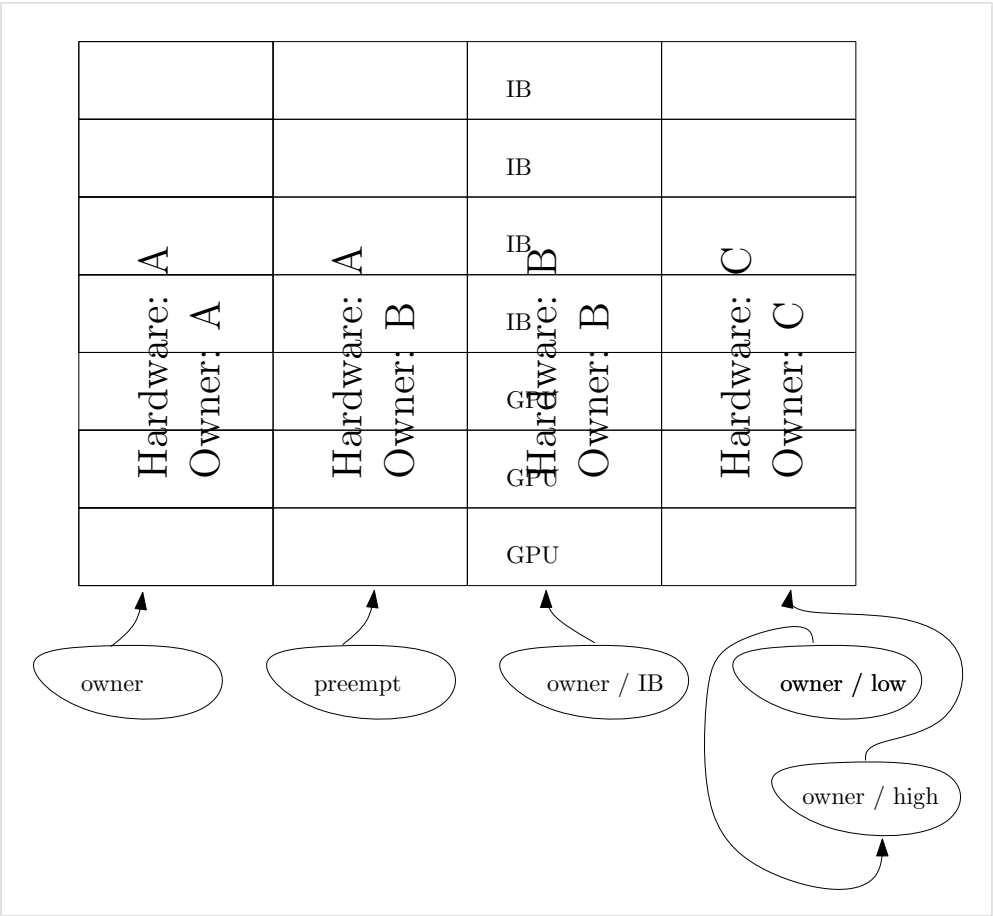
Hardware that Moab must Understand



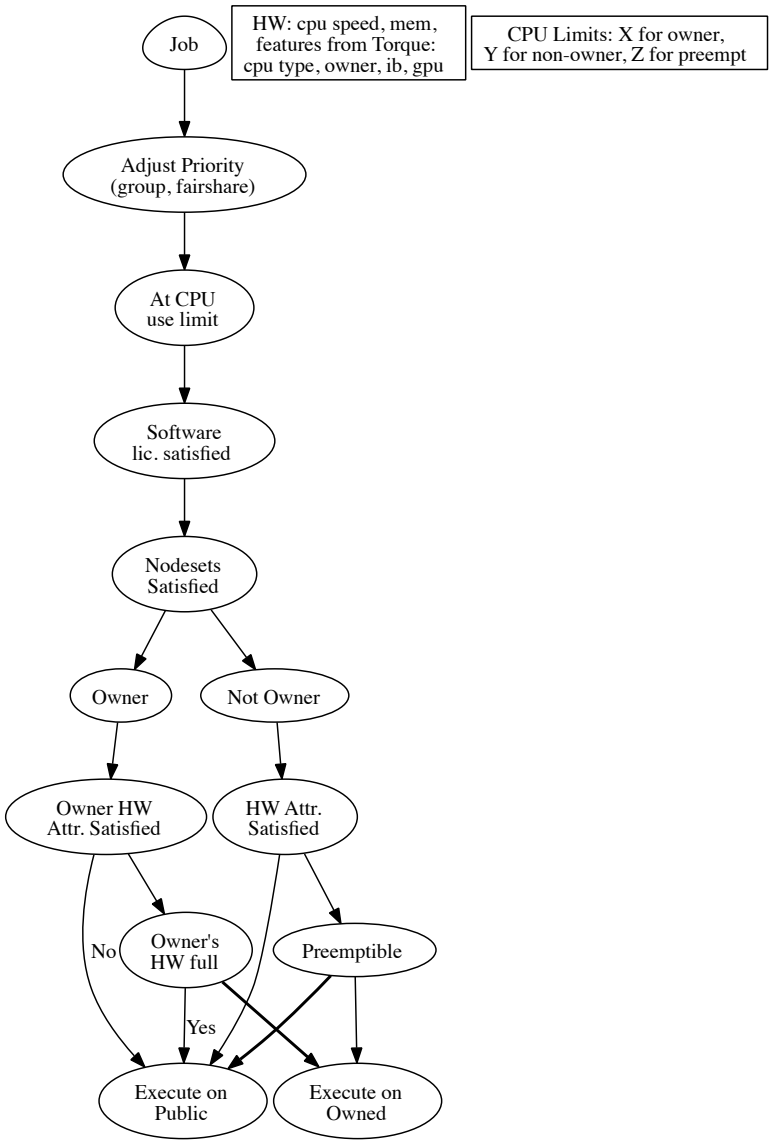
Hardware that Moab must Understand



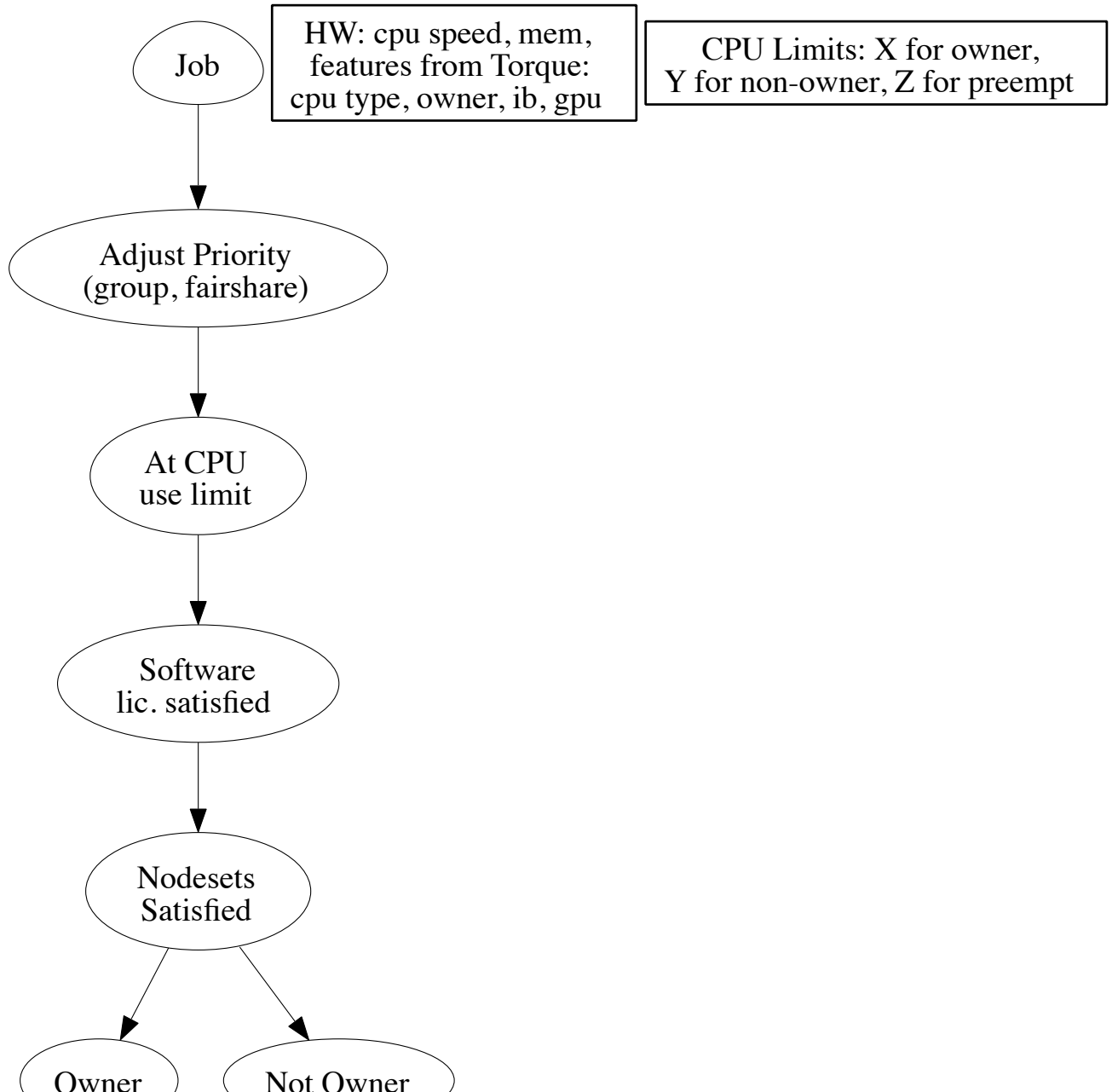
Hardware that Moab must Understand

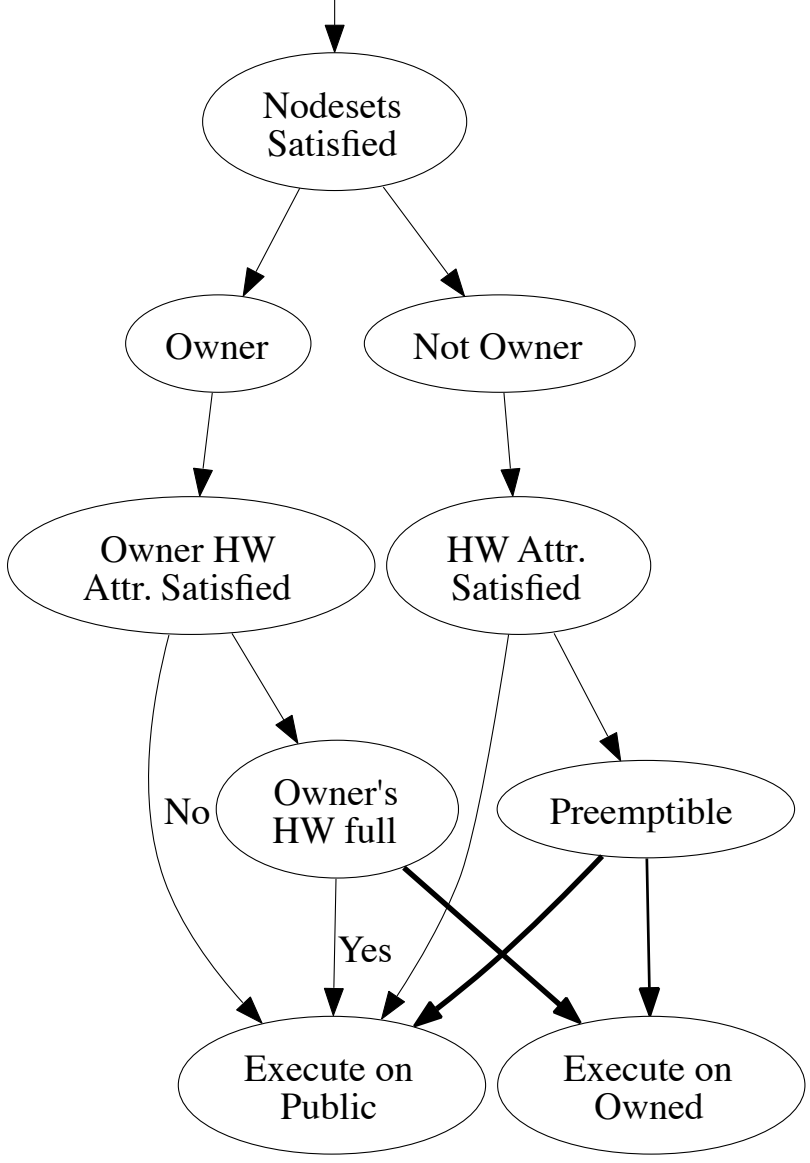


Moab's Decisions



MichiganEngineering


Job
HW: cpu speed, mem, features from Torque: cpu type, owner, ib, gpu
CPU Limits: X for owner, Y for non-owner, Z for preempt
Adjust Priority (group, fairshare)
At CPU use limit
Software lic. satisfied
Nodesets Satisfied
Owner
Not Owner


Nodesets Satisfied
Owner
Not Owner
Owner HW Attr. Satisfied
HW Attr. Satisfied
No
Owner's HW full
Preemptible
Yes
Execute on Public
Execute on Owned

Moab: where the rules live

Moab is where all the rules are:

- there are a lot of rules
- within the overarching set of rules, there can be a lot of rules local to an owner's hardware
- the rules can change
- we are adding owners regularly

Moab is invaluable in enforcing the rules. (Although sometimes we wish it was a little more transparent in what it was doing.)

Moab: where the rules live

Moab is where all the rules are:

- there are a lot of rules
- within the overarching set of rules, there can be a lot of rules local to an owner's hardware
- the rules can change
- we are adding owners regularly

Moab is invaluable in enforcing the rules. (Although sometimes we wish it was a little more transparent in what it was doing.)

Moab: where the rules live

Moab is where all the rules are:

- there are a lot of rules
- within the overarching set of rules, there can be a lot of rules local to an owner's hardware
- the rules can change
- we are adding owners regularly

Moab is invaluable in enforcing the rules. (Although sometimes we wish it was a little more transparent in what it was doing.)

Near Future

- Turning preemption back on
- Using Gold for allocations: reflecting policy
- "Floating" reservations based on node type: encouraging sharing
- More sophisticated preemption rules: preempt based on state of preemptee
- Performance improvements in scheduling and user responsiveness

Distant Future

- Dynamic cloud provisioning based on job attributes
- Dynamic diskless node provisioning from a computer lab environment
- Preemption policies based on any requestable attribute: software, special hardware, disk, etc.
- Multi-layer preemption: A can preempt B, and C; B can preempt C; C just suffers.
- Preemptability based on policy: fairshare, allocation, etc.

Questions?

Andy
acaird@umich.edu

Matt
msbritt@umich.edu

Brock
brockp@umich.edu

