SuperB: pushing the limits of Torque and Maui

how to federate clusters while keeping them independent

Matteo Guglielmi
Vittoria Rezzonico
EPFL, SB-IT

May 18, 2011

Everybody wants a cluster

The Beowulf Cluster

- Commodity machines interconnected by commodity interconnect and running commodity software

Everybody wants a cluster

The Beowulf Cluster

- Commodity machines interconnected by commodity interconnect and running commodity software
- You can order your cluster off the internet and have a geek install some linux distro on it

Everybody wants a cluster

The Beowulf Cluster

- Commodity machines interconnected by commodity interconnect and running commodity software
- You can order your cluster off the internet and have a geek install some linux distro on it
- That's what was done before we came into play

Plan of attack

- choose some clusters according to the criteria:
  - under-utilised
  - have the same Linux distribution
  - have the same administrator
  - owner is a cool person

Plan of attack

- choose some clusters according to the criteria:
  - under-utilised
  - have the same Linux distribution
  - have the same administrator
  - owner is a cool person
- install a super-master
  - authentication
  - unified installation
  - scheduler

Plan of attack

- choose some clusters according to the criteria:
  - under-utilised
  - have the same Linux distribution
  - have the same administrator
  - owner is a cool person
- install a super-master
  - authentication
  - unified installation
  - scheduler
- migrate clusters into new system

Plan of attack

- choose some clusters according to the criteria:
  - under-utilised
  - have the same Linux distribution
  - have the same administrator
  - owner is a cool person
- install a super-master
  - authentication
  - unified installation
  - scheduler
- migrate clusters into new system
- buy more nodes and storage to integrate to the system

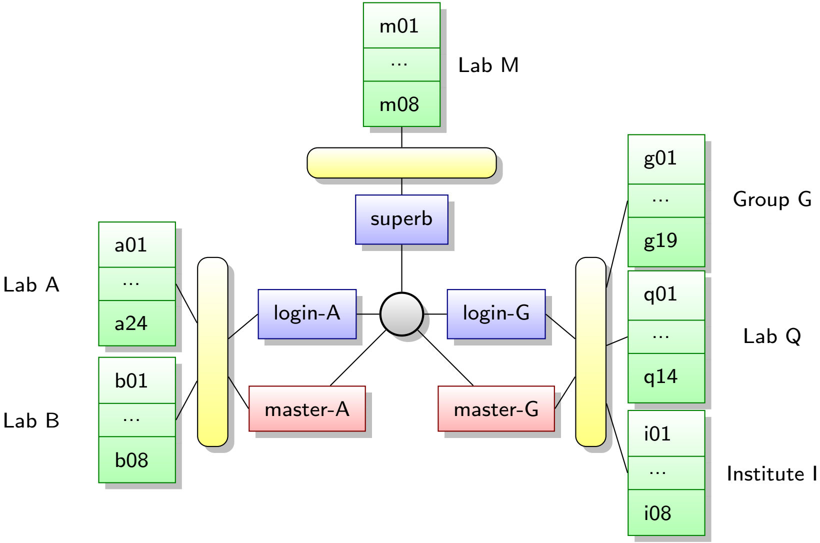
Plan of attack

- choose some clusters according to the criteria:
  - under-utilised
  - have the same Linux distribution
  - have the same administrator
  - owner is a cool person
- install a super-master
  - authentication
  - unified installation
  - scheduler
- migrate clusters into new system
- buy more nodes and storage to integrate to the system
- happy users!

What is SuperB

- the Super Beowulf Cluster of the Basic Sciences School
- a federation of clusters
- sharing authentication, scheduler and installation method
- clusters (and node groups) keep their independence
- priorities are based on a owner-guest model (more on that later)

SuperB: architecture



A and B are good friends (they can share a frontend), $Q \subset I \subset G$

SuperB rules

Owners

- have purchased their own nodes
- have absolute priority on their nodes
- can access other nodes on SuperB with some restrictions

Guests

- do not *own* nodes
- are given access to all nodes in SuperB under certain restrictions
- sometimes after a while decide to buy their own nodes

Owners do not want to notice they're in a federation

- want to have instant access to all their nodes
- priority over guests in the queue
- if a guest is running on their nodes and they submit a job, guest job must stop
- want to enforce their own rules on their share of the cluster

Requirement for the scheduler

Must:

- node-to-users mapping
- preempting (of all types of jobs)

Should:

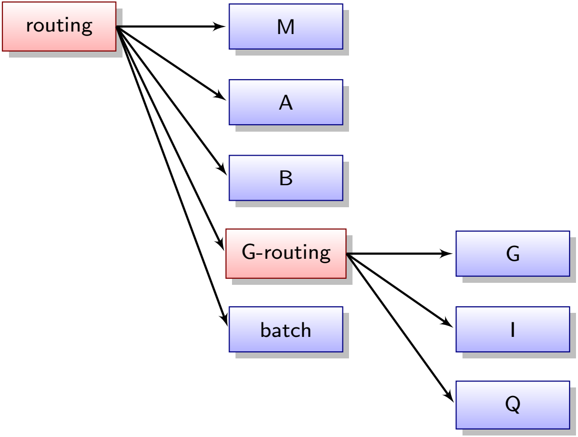
- shortpool

The Scheduler Today

- Torque resource manager + Maui scheduler
  - historical reasons
  - they're free and open-source
  - we're open to re-evaluate our choice
- following musts are fulfilled
  - node-to-users mapping via node to queue mapping + nodes ACLs
  - preempting of (most) jobs

Queues hierarchy

A, B, M are at the same level, and $Q \subset I \subset G$ are a separate entity. The batch queue welcomes guests:



The configuration

Part I: Torque

The main routing queue will take care of redirecting jobs to the Labs' queues, Group G's routing queue and the guests' queue batch:

```
set server default_queue = routing
create queue routing
set queue routing queue_type = Route
set queue routing route_destinations = M-queue
set queue routing route_destinations += A-queue
set queue routing route_destinations += B-queue
set queue routing route_destinations += routing-G
set queue routing route_destinations += batch
set queue routing route_waiting_jobs = True
```

If a user is in a queue's ACL, he gets in. Otherwise, he goes to the next queue. At a last resort, he goes to the batch queue.

The configuration

Part I: Torque

For each Lab or Institute (A, B, M, Q, I), hosts and users ACLs are specified in its queue:

```
create queue lab_queue
set queue acl_host_enable = False
set queue acl hosts += ...
set queue acl_user_enable = True
set queue acl users += ...
```

Group G coordinates its nodes' access in a special way. The routing queue will act as a sorting point for users:

```
create queue routing-G
set queue routing-G queue_type = Route
set queue routing-G acl_host_enable = False
set queue routing-G acl_hosts += ...
set queue routing-G acl_user_enable = True
set queue routing-G acl_users += ...
set queue routing-G route_destinations = I-queue
set queue routing-G route_destinations += Q-queue
set queue routing-G route_destinations += G-queue
```

Everybody that passed thru routing-G has access to G-queue, so no need to define user acl:

```
create queue G-queue
set queue G-queue queue_type = Execution
set queue G-queue acl_host_enable = False
set queue G-queue acl_hosts += ...
```

The configuration

Part I: Torque

Each lab's nodes are different. We use node properties to indicate for example the infiniband network or the processor type:

```
qmgr -c 'set node a01 properties = ib-a'
qmgr -c 'set node a01 properties += X5355'
```

Users will be able to choose a specific processor by the mean of node properties (more on this later).

The configuration

Part II: Maui

Preempting: the act of stopping a task with the intention of resuming it at a later time.
The jobs in the batch queue (guests' jobs) are declared as preemptees:

```
QOSCFG[batch]  QFLAGS=PREEMPTEE
```

The owner jobs are declared as preemptors:

```
QOSCFG[owner]    QFLAGS=PREEMPTOR:IGNSYSTEM
```

A preemptor can preempt a preemptee. Then queues are classified as owner or batch, for example:

```
CLASSCFG[A-queue] QDEF=owner PRIORITY=10000
CLASSCFG[batch]   QDEF=batch PRIORITY=0
```

In our case, a batch job is stopped (suspended) in order to give priority to a owner job.

```
PREEMPTPOLICY              SUSPEND
```

The configuration

Part III: what the user should know

How to submit jobs

```
#PBS -l walltime=00:15:00,nodes=8:ppn=2:PROPERTIES
#PBS -q QUEUE
```

- if a user does not specify a queue, his job will go to the more restrictive one:
  - if he belongs to unit Q, it will go to Q-queue
  - if he belongs to institute I but not unit Q, it will go to I-queue
  - if he is a guest, it will go to the batch queue
- a user can specify a less restrictive queue (for example if he wants to access a colleague's nodes), typically the batch queue
- node properties can be used to pick specific nodes

  ```
  #PBS -l walltime=00:15:00,nodes=8:ppn=2:ib-a:X5355
  #PBS -q batch
  ```

Some numbers

Participating entities	9
Users	153 (active: around 100)
Number of nodes	89
Number of cores	752
Amount of RAM	1608GB
TFLOPS peak	7.673
Separate InfiniBand networks	3

If we had more time and money, we would...

- buy or set up some parallel filesystem appliance to be shared among the clusters in SuperB
- buy some slow storage space for backup and archiving
- change the scheduler (work in progress)

Questions