Flocking at BNL: Implementing a General Use Queue

HEPiX Fall 2005 at SLAC

Alexander Withers
Brookhaven National Laboratory
alexw@bnl.gov
10/10/05

BROOKHAVEN NATIONAL LABORATORY

Summary of Condor Setup

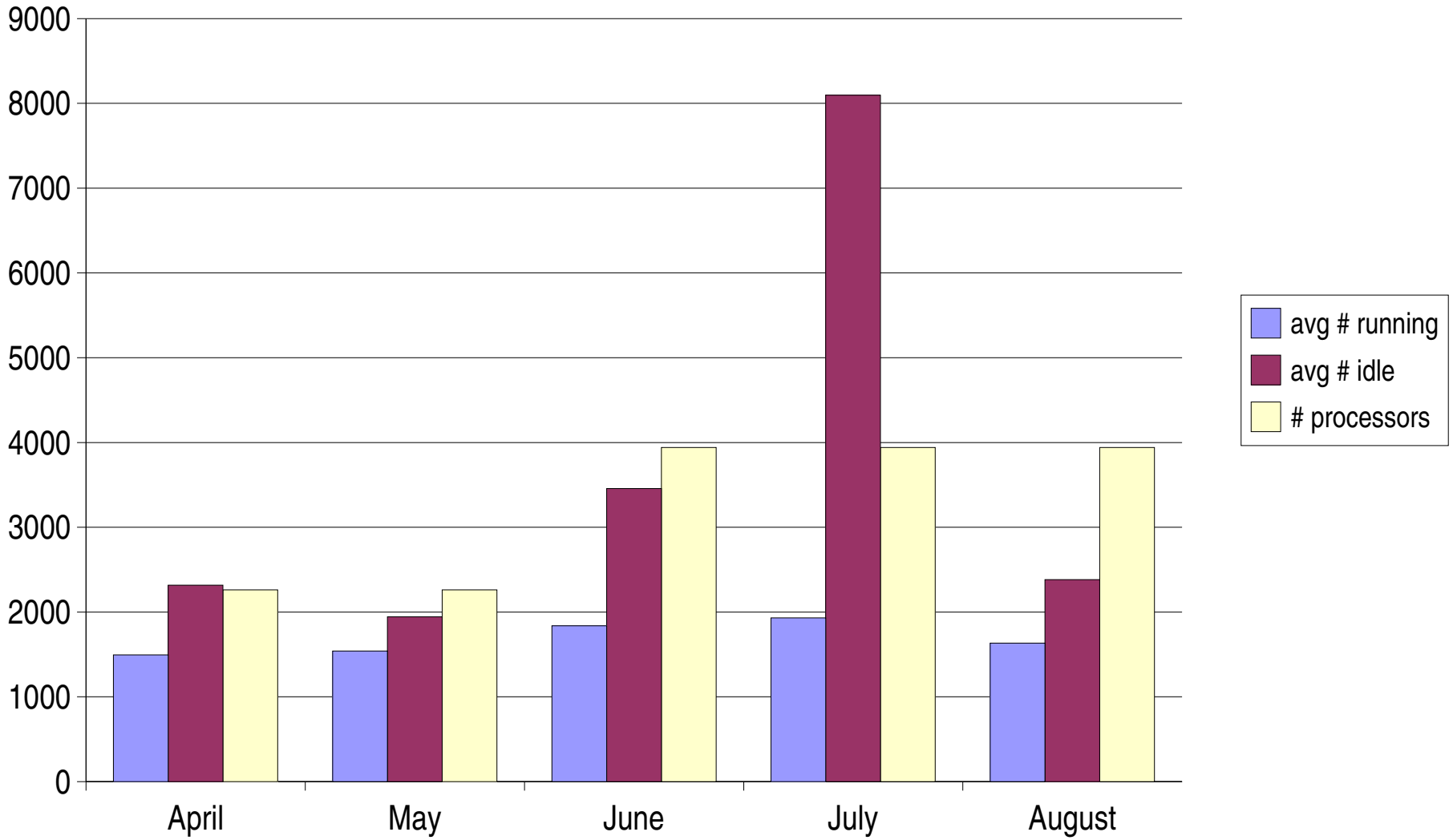
- 5 Condor pools on two central managers.
- ~2050 nodes each capable of running 2 jobs.
  - 113 of these nodes act as submit.
- Rough distribution of nodes: 25%, 25%, 25%, 20%, and 5%.
- One monitoring/Condorview server and one backup central manager.

BROOKHAVEN NATIONAL LABORATORY

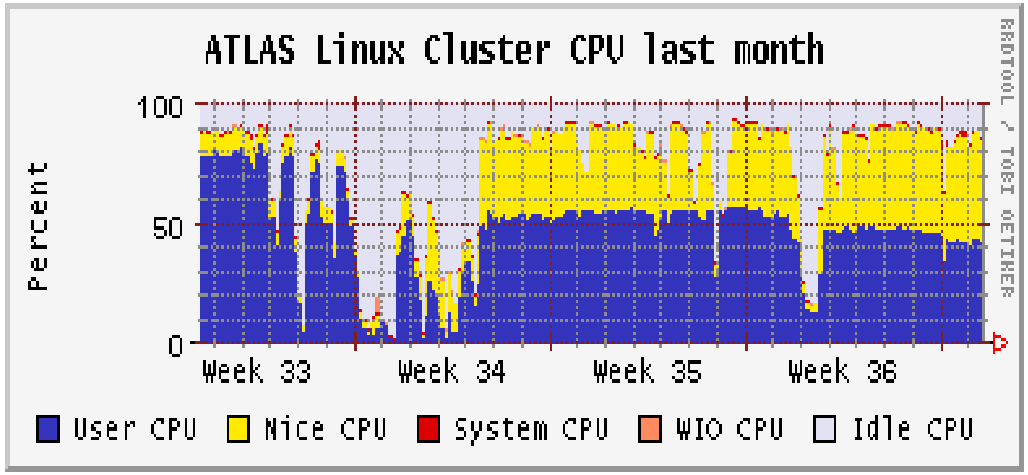
Computing Resources Divided

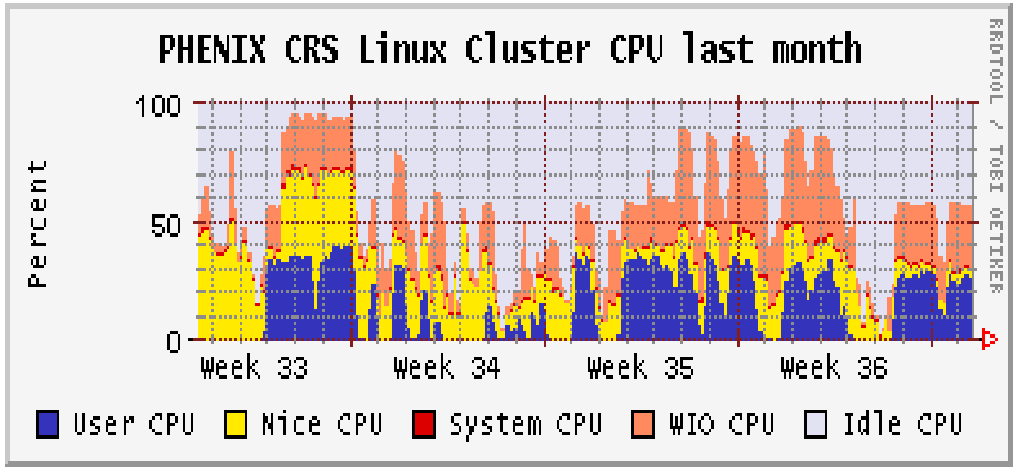
- Five experiments each with their own share of CPUs, disks, etc.
- Many computing cycles wasted since usage varies from experiment to experiment.
- The demand for computing resources increases each year.
- Need to make efficient use of existing resources.

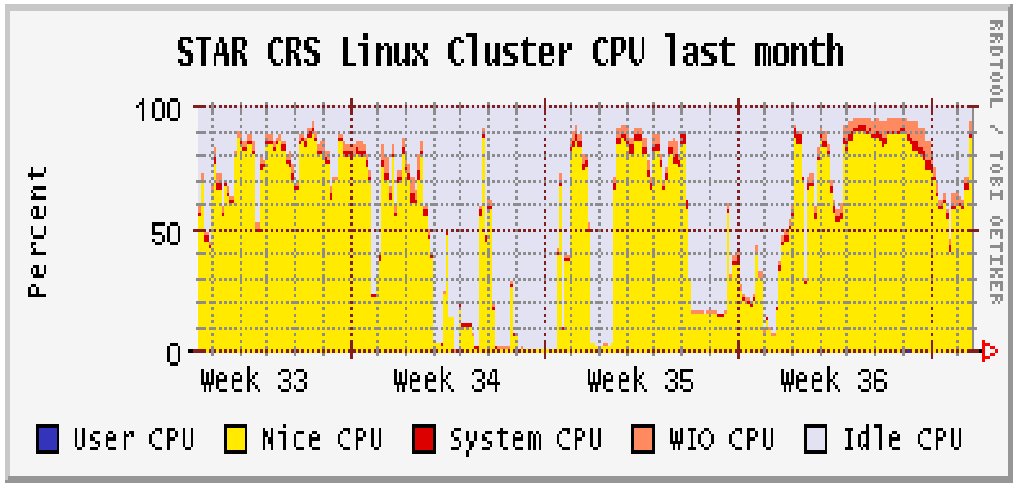
Usage History of All Condor Pools



BROOKHAVEN NATIONAL LABORATORY







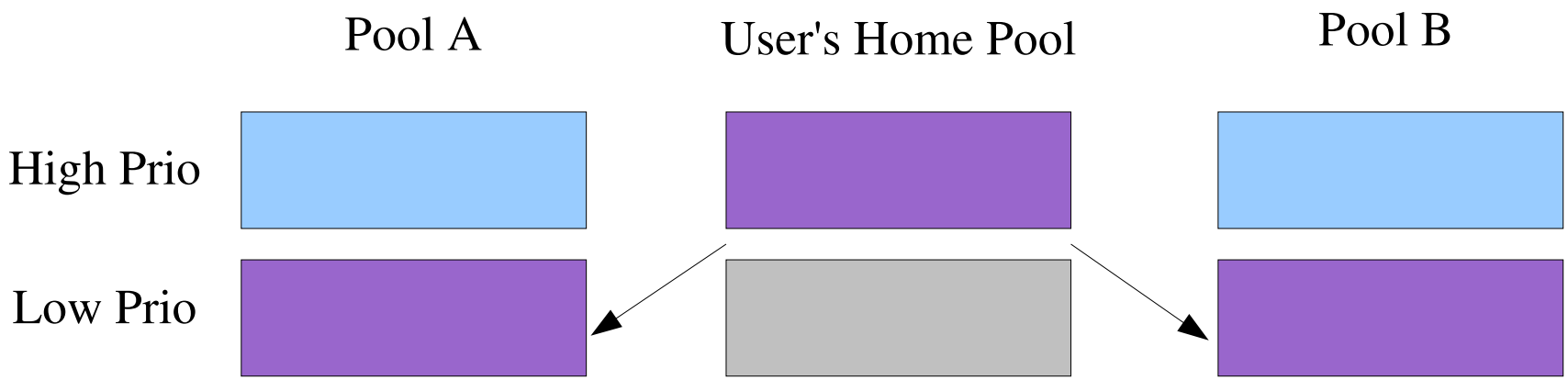
Typical Ganglia Snapshot

BROOKHAVEN NATIONAL LABORATORY

General Plan for Sharing Resources

- Flocking allows for one Condor schedd to pass a job off to another pool when it cannot find a match in its own pool.
- This allows for two or more pools to share resources in an efficient manner.
- Jobs from other pools cannot impact jobs running on their own pool.
  - This means that flocked jobs run at a lower priority.
  - Dealing with lower priority jobs should be delayed for as long as possible.

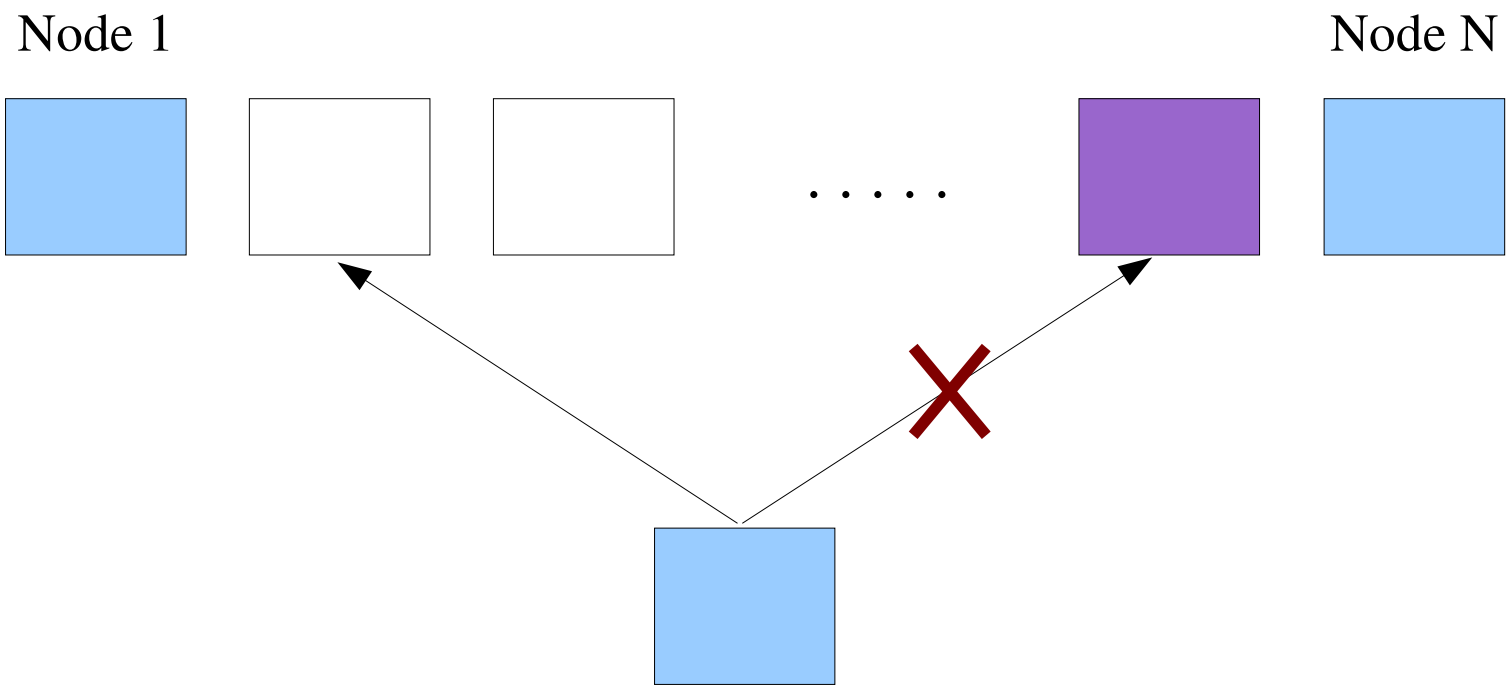
Flocking Scheme



- Jobs flow outside of their pool when they've filled their own.
- When jobs run outside of their pool they're treated with lower priority.

BROOKHAVEN NATIONAL LABORATORY

Job Placement Scheme within a Pool



- Have jobs prefer nodes without low priority jobs.
- The hope is to give low priority jobs as much time as possible.

BROOKHAVEN NATIONAL LABORATORY

Most of the Work is Done

- Use virtually the same OS image for all nodes, regardless of the experiment.
- All NFS and Panasas mounts are mounted on each node (via automount and directflow).
- All experiments use the same LDAP database.
- Have yet to address specific requirements such as ROOTD, Proofd, dCache, etc.

General Issues with Flocking

- Local data disks: who gets to use them and how will they be restricted.
- Large scaling testing of sharing between many pools has not been thoroughly tested.
  - Main concern: load on the negotiators and collectors.
  - Also, concerned about load on the schedds.
- Flocked jobs cannot affect the pool's existing local priority scheme.
  - Need to use same tools to establish lower priority for flocked jobs.
  - Flocked jobs are subject to local setup.

BROOKHAVEN NATIONAL LABORATORY

How to Deal with Flocked Jobs

- Flocked jobs are to be treated with a lower priority.
- "lower priority" can mean many things in Condor. For example:
  - Suspend jobs.
  - Evict jobs nicely.
  - Evict jobs immediately.
  - Adjust user priority settings.
  - Group accounting and quotas.

Suspend Jobs

- Doesn't consume CPU but does consume memory. Can be a problem for experiments that use many VMs.
- A suspension policy has no way to guarantee a certain amount of runtime.
- Would have to add more VMs to all experiments. This might strain the policy of some experiments.

Evict Jobs Immediately

- Works by giving each machine a RANK statement.
- This means that the machines would prefer jobs from one experiment over another.
- The action of “prefering” translates into eviction for Condor. Thus, “prefering” doesn't necessarily happen at scheduling.
- General queue jobs may get throttled if the system is busy.
- May be able to quell throttling by having jobs flow towards the least loaded nodes.

BROOKHAVEN NATIONAL LABORATORY

Evict Jobs Nicely

- Same technique as in “Evict Jobs Immediately”.
- When a job is evicted Condor can wait for a given amount of time.
- This allows for the possibility of the job finishing before it gets evicted.
- A common time limit would need to be decided upon.
- Would not interfere with experiments that already have this feature in place.

Adjust User Priority Settings

- This requires eviction based upon user priority.
- The idea is to give local users (those who belong to the experiment) a lower priority factor than non-local users.
- Those with a lower priority factor will get a greater share of the resources.
- A lot of user priority settings have to be done manually.

Group Accounting and Quotas

- Users belong to a specific group.
- Each group gets a quota which guarantees a certain amount of resources.
- The idea is to have one pool and use the negotiator to split the pool up.
- Jobs without a group are treated as “general queue” jobs.
- Only available in 6.8.x and has not been thoroughly tested.

The Implementation

- Allow jobs to start on a node provided:
  - The job belongs to that experiment or...
  - The job does not belong to that experiment but it wants to run in the general queue.
  - Jobs in the general queue will run on their own machines as normal jobs.
- Have each node RANK by its experiment.
- Jobs running in the general queue will be evicted immediately.
- Jobs will automatically flow towards the least loaded nodes.

BROOKHAVEN NATIONAL LABORATORY