glideinWMS Training

How is glideinWMS different from vanilla HTCondor

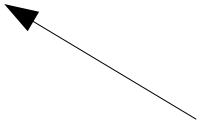
by Igor Sfiligoi, UC San Diego

Overview

- These slides provide an overview of why glideinWMS installations behave differently than dedicated, LAN-based HTCondor ones

Very heterogeneous res. pool

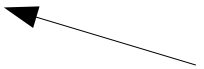
- Many user jobs have data constraints
  - And data access varies from site to site
- Each site basically results in a different "type of resource"
  - Making the resources very heterogeneous, for matchmaking purposes
  - O(100) types of resources not unusual
- Leads to autocluster number explosion



In dedicated HTCondor pools, 5 classes of resources is typically already a lot

Provisioning vs matchmaking

- Glideins are provisioned
  (i.e. requested from sites)
  because some user jobs need more resources
  - But once provisioned they may not match any jobs
- Two main reasons
  - Trigger jobs already gone (i.e. not idle anymore)
  - Mismatch between provisioning and matchmaking requirements



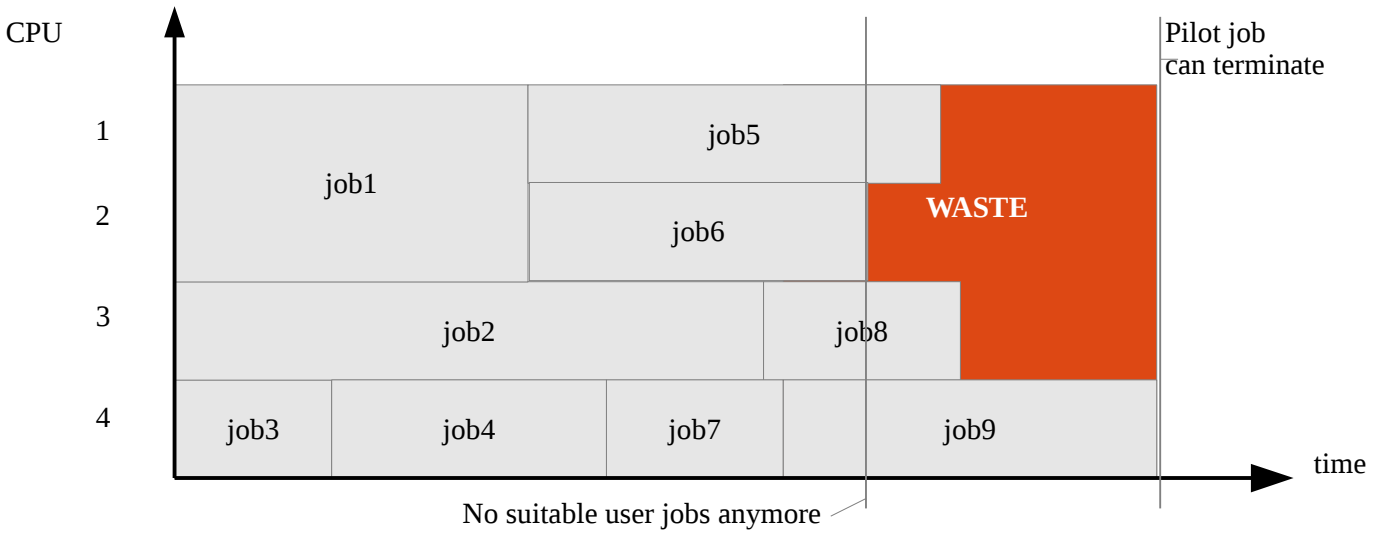
Dedicated HTCondor installations don't have 2 levels of matchmaking

Limited lease lifetime

- Glideins are basically leased execute nodes
  - And they come with a limited lifetime
- Lease times usually in the order of one day
  - Each glidein typically runs less than 10 user jobs
- User jobs must fit in the remaining lifetime
  - Or they will be killed
- Makes for more complex matchmaking decisions
  - And requires user help

Multicore and limited lifetimes

- Limited lifetimes particularly problematic for multi-core jobs, resulting in significant waste
  - Since it is unlikely all jobs will terminate at exactly the same time



Automatic shut down

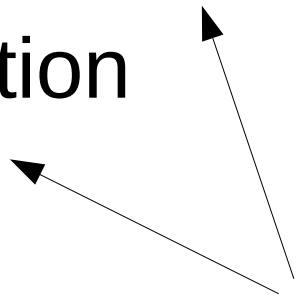
- Glideins are configured to shut down automatically if not used for some time
  - Those resources could be used by someone else
  - HTCondor not the only user of the resources
- Default Unclaimed threshold quite low
  - About 10 minutes

- This puts stringent limits on Matchmaking
  - If a Startd is not matched in time, it is “lost”
  - And restarting glideins is expensive

Unlike a dedicated HTCondor pool

Strong end-to-end security

- A glideinWMS system will typically span many different locations
- x509 authentication between all nodes required
  - At daemon startup, then sec. session cached
  - With the exception of Schedd<->Startd, where security mediated through the Collector
- All over-the-wire communication Integrity checked
  - Requires auth.



Neither typically used in LAN deployments

Not privileged on execute side

- HTCondor daemons on the execute side do not have system privileges
  - Limits what HTCondor can do
- UID switching can be achieved with glexec
  - But requires proxy delegation from schedd
  - Only possible if users collaborate
  - Relatively expensive (at least one per job startup)
- Many other functions not an option
  - e.g. cgroups

Firewalls

- HTCondor basically a P2P system
  - But execute nodes are often behind firewalls
- Requires the use of CCB and shared_port_daemon to get around it
  - But this adds complexity to the system
  - Schedd particularly sensible here
- CCB can become single point of failure
  - Either because temp. overloaded
  - Or if it dies and HA not used

Very dynamic resource pool

- Startds tend to come and go often
  - A side effect of limited lease lifetime
  - And provisioning due to new jobs being submitted
- Many HTCondor optimizations less effective
  - e.g. Security session caching

Increased resource pool size

- Most glideinWMS installations bigger than most LAN HTCondor installations
  - At least at the peaks
- Increased scale puts more load on non-execute daemons
  - Even before all the other considerations are applied

The end