CHTC Policy and Configuration

Greg Thain
HTCondor Week 2017
CHTC Pool Mission
CHTC Pool Mission

“To improve computational research on campus by providing access to high throughput computing to a large number of users”
One consequence: Max Runtime

Why?

- Minimize badput
- Prevent one user from monopolizing pool
- Encourage use of shared resources like OSG

Currently choose 72 hour max runtime
CHTC Pool

CHTC submit machines

jobs

CHTC execute machines

foreign execute machines (flocking, OSG glidein, etc.)

Foreign submit machines
Want max job runtime policy changes in our startds, foreign startds may have other policy.
Policy Question

› Should condor preempt job at 72 hours…

Even if no other job is runnable ????
“Jobs should be evicted after 72 hours of runtime”
WANT_HOLD = \ 
    TotalJobRunTime > ( 72 * 3600)
WANT_HOLD_REASON = \ 
    "Job failed to run in 72 hrs"
Not quite right…

Policy
- “Jobs should be evicted after 72 hours of runtime”

Only care about execution attempt:
What about self checkpointing job?
How to identify checkpointable jobs?

» +Is_Resumable = true

» Should this be 1st class in condor?
WANT_HOLD = Is_Resumable == true &&
            TotalJobRunTime > (72 * 3600)

WANT_HOLD_REASON = \\
    "Job failed to run in 72 hrs"
Is_Resumable has more uses

- More we know about a job, better we do
  - Even a little bit of knowledge – not full runtime guess

  - Used for backfilling clusters
  - Running in other specialized environments.
Moral of this story

› Be very careful with policy statements
› We need users to tell us more about jobs
› “Specific is Terrific”!
CHTC Pool

CHTC submit machines

jobs

CHTC execute machines

foreign execute machines (flocking, OSG glidein, etc.)

Foreign submit machines
Consequence

▷ Want to have many places to run

▷ But we don’t control foreign machines:
  • Operating System, policies, preinstalled software
  • Leads to user surprises
Jobs only run “at home” by default, can opt into foreign resources

- Two levels:
  - On campus foreign pools
  - OSG Pool (really foreign)
Policy must be at schedds, with id help from startds.
PoolName = "CHTC"
Site = "ServerRoomNumber"
STARTD_ATTRS = PoolName, Site
condor_status output

condor_status -af Site PoolName

...  
Site = “ServerRoomNumber”
Pool = “CHTC”
...
Digression: good condor style

- Jobs choose based on attribute of machine
- NOT on a-priori knowledge
  - Machine name, etc.
- Machines publish info about themselves
Our local machines have CVMFS, others don’t. If a job needs CVMS, how should it indicate this?
Wrong way

Requirements = PoolName == "CHTC"
Right way – startd cron

STARTD_CRON_JOBLIST    = $(STARTD_CRON_JOBLIST) CVMFS
STARTD_CRON_CVMFS_EXECUTABLE  = /path/to/check_cvmfs
STARTD_CRON_CVMFS_PERIOD    = 10m
STARTD_CRON_CVMFS_MODE     = periodic

Check_cvmfs is a script which emits
HasCVMFS = true (or not)
condor_status output

condor_status -af HasCVMFS

... HasCVMFS = true ...

End of digression, back to flocking

- Want jobs to only travel if they opt in

- By providing default schedd requirements
APPEND_REQ_VANILLA =
  MY.WantFlocking || TARGET.PoolName =?= "CHTC"

requirements = (USER STUFF) &&
  MY.WantFlocking || TARGET.PoolName =?= "CHTC"

My.WantFlocking comes from job ad

Trivia: What if not defined?
Policy: defragmenting partitionable slots

- With partitionable slots, and mixed size jobs, can have job starvation
- Problem: can take long time to fully defrag machine
Defrag policy

DEFRAG_REQUIREMENTS = PartitionableSlot \ 
    && State != "Owner" && TotalCpus > 39
DEFRAG_WHOLE_MACHINE_EXPR = PartitionableSlot && \TotalCpus > 39 && ( State != "Owner" ) && \
    ( ( cpus / TotalCpus ) >= 0.8)

Only defrag down to 80%
Problem: high prio job goes first

› May wait a long time to defrag a machine
› But then a high prio serial job comes along…
› Negotiation hard coded to match in prio order
Solution: delayed start

START = $(START) &&

RequestCpus >= IfThenElse(Cpus < 4,1,4) || \
(time() - EnteredCurrentState) > 20 * 60)

After defrag, only start big jobs for a few minutes
Condor has all kinds of assembly language for policy, but we need to spend more time thinking about what the actual, english-language policy we want is.