Machine/Job Features TF

Andrew McNab
University of Manchester,
GridPP, and LHCb

Overview

- MJF TF aims
- Specification
- Transport mechanisms
- Keys/values
- Implementations
- SAM probe
- Rollout

https://twiki.cern.ch/twiki/bin/view/LCG/MachineJobFeatures

Aims of Machine/Job Features Task Force

- A common API that jobs can use to discover the parameters of their environment
  - eg wall clock time limit
- Otherwise requires a patchwork of environment variables and command call-outs
  - Different for each batch system: qstat etc
  - Not available in VM-based environments
- So N experiments have to write implementations for M batch systems (N x M)
  - With MJF, goes more like N + M

Specification

- Several iterations, starting from the HEPiX virtual machines working group, then into this WLCG TF, as talks/Twiki pages
- Last autumn and start of 2016 the task force agreed a specification for MJF
  - The set of key/value pairs to publish
  - How jobs can get the key/value pairs
- Published as HEP Software Foundation technical note (HSF-TN-2016-02)
- Consistent definitions with APEL and Infosys TF

Transport mechanisms

- Jobs expect $MACHINEFEATURES and $JOBFEATURES to point to “directories” containing key/value pairs
  - File name is key; content is value
- Simple cases: $MACHINEFEATURES=/etc/machinefeatures
- For worker nodes, usually local directories
- For VMs though, “directory” is a URL on a web server populated by the VM lifecycle manager
  - EC2/OS metadata keys to discover URLs

Key / values

- The technical note has the full list with definitions
- Sites should supply them if they know the value (eg HS06)
- Values can typically be discovered from batch system, with OS values as a fall-back
- shutdowntime allows sites to declare a cut-off when draining

$MACHINEFEATURES

total_cpu
hs06
shutdowntime
grace_secs

$JOBFEATURES

allocated_cpu
hs06_job
shutdowntime_job
grace_secs_job
jobstart_secs
job_id
wall_limit_secs
cpu_limit_secs
max_rss_bytes
max_swap_bytes
scratch_limit_bytes

Implementations

- Vac/Vcycle supply MJF directories to their VMs
- PBS/Torque and now HTCondor scripts exist in GitLab and as RPMs
  - Common code where possible; same ideas
  - Tested/running at Manchester, PIC, Cambridge, Liverpool (thanks!)
- Need more volunteer sites to test scripts please
  - Need volunteers from sites with other batch systems too, to help develop scripts
- See https://twiki.cern.ch/twiki/bin/view/LCG/MachineJobFeaturesImplementations

SAM probe

- WN-mjf.py script in the GitLab repo
  - Runs inside jobs to look for MJF keys/values
- Volunteer sites passing tests when script is run:
  - By hand inside jobs
  - In the ETF pre-prod (SAM) service
- Key to rollout and debugging
  - Provides an objective test as to whether the site has “got there”
- Same script ok for any batch system or experiment
  - That’s the point of MJF!

Rollout strategy

- Now we have the SAM probe working, aim to go beyond sites that volunteered in testing
  - Will still find problems due to inter-site differences
- Intend to approach some communities directly for this initial rollout
  - eg GridPP sites, LHCb Tier-1s
- Then follow usual pattern of cvmfs etc rollouts
  - Chasing up sites with difficulties
- Key is making it simple to do: minimally, you just install the RPM and maybe change a config file

Summary

- Long and winding road from original HEPiX VMs proposal
- Now on firm footing with the specification
- Implementations already for PBS/Torque, HTCondor, Vac/Vcycle
- SAM probe allows us to monitor rollout
- Now we need to recruit more sites to take rollout beyond the testing phase
- **Please mail me to volunteer! Thanks!**