Proposal for deployment of a communication layer between VO and resource provider

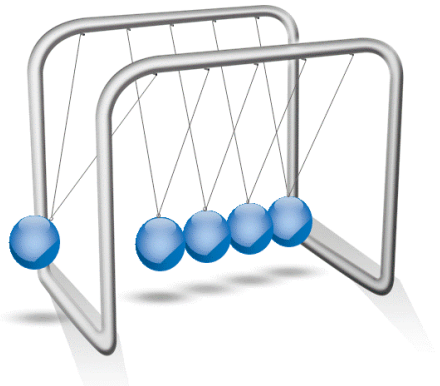
Stefan Roiser

GDB

12 June ‘13

History of "Machine/Job Features"

- [June, Oct 2012] Presentations in GDB
  - Initiated from Hepix working group
- [2012] Specification and several implementations
  - CERN/LSF, NIKHEF/Torque, KIT/SGE&PBSpro, but no heavy testing (NB 272 of 277 sites use one of the systems above)
- [2013/5/23] Discussion during LHCb computing workshop on use cases for machine features
- [2013/5/30] Proposition in WLCG Ops Coord meeting
  - Positive feedback from sites and VOs
- [2013/6/5] Brain storming meeting CERN/IT OIS+PES+SDC
  - Possible integration of virtual and batch environments

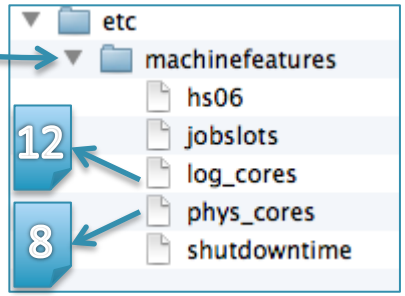


Goal of this Talk

- Propose one interface
  - to all interested VOs to retrieve machine and job specific information
  - in batch and virtualized environments
  - implementable by all resource providers
  - in the environment of “compute intensive” data processing

Machine / Job Features

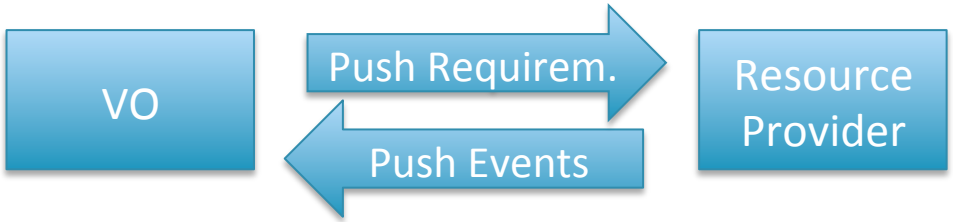
- A way to provide individual information about the worker node and job constraints to the VO pilot
- VO pulls info via two environment variables pointing to directories containing files with specific values
  - $MACHINEFEATURES
    - Info on: node power, # log/phys cores, shutdown time
  - $JOBFEATURES
    - Info/job on: # cpus alloc, cpu/wall/mem/disk limit, ...



How to pass this info in a virtualized environment?

Extending to IaaS systems

- Idea taken from “Poncho” system developed by Argonne
  - Pass a URL (and more info) to IaaS
    - VO->Resource: min runtime, min notification time, internal priority
    - Resource->VO: dealloc one of your instances, shutdown instance X in Y time



- Stay within the same architecture: Pull information VO<-Resource
  - Communication via “magic IP” (http) or “config drive” (file system)

  - Need script inside VM to determine whether Info is pulled via http or FS and set environment variable accordingly

Use Cases

- Get info on cores allocated to the payload for multi-core submission
- Get the HepSpec06 power, # of physical / logical cores of a node
- Calculate the “queue length” by individual WN
- “Monte Carlo Factory”, fill the end of a queue with MC and gracefully shutdown
- Drain a worker node / hypervisor, eg. for dynamic rolling interventions on a site
- Notify all machines in HLT farms to shutdown
- Rebalancing of VO shares

Interface for the VO

```
def getMachineFeatures():
  var = os.environ['MACHINEFEATURES']
  if var[0] == '/' : return getInfoFS('MF')
  elif var[:4] == 'http' : return getInfoHttp('MF')
  else : sys.exit(1)

>>> getMachineFeatures()
{u'hs06': 16.35, u'jobslots': 12, u'log_cores': 16,
u'phys_cores': 8}
```

Proposed agreement VOs / Sites

- In case the VO payload implements this mechanism it has the chance to read info and react on events from the site
  - If the mechanism is not implemented the site shall be free to handle the VM/pilot whatever seems appropriate after an agreed grace period

What is needed to continue

- Interest from VOs and sites
- Extend the specification for the interface
  - Specify syntax and return value ranges
  - In case of interfacing to IaaS systems
- Provide the missing implementations
  - Batch systems: SLURM, Condor
  - IaaS systems: OpenStack, CloudStack, ...
- Plan for deployment and its monitoring

Thanks

- Maite, Manuel, Marek, Mattia, Tim, Ulrich,

Links & Docs

- Specs https://twiki.cern.ch/twiki/bin/view/LCG/WMTEGEnvironmentVariables
- Oct '12 GDB http://cern.ch/go/Qv9v
- June '12 GDB http://cern.ch/go/D7RG
- June '13 WLCGOps http://cern.ch/go/JW9T
- [4] S Devoid, L Hochstein, N Desai; Poncho: Enabling Smart Administration of Full Private Clouds