HTCondor European Workshop Summary

HEPIX, March 2015

Michel Jouvin

jouvin@lal.in2p3.fr

Meeting Facts

- European HTCondor Admins Workshop
  - At CERN, December 8th-9th 2014
  - Idea at HEPiX in Nebraska, organized as a pre-GDB
  - Several years since last European Condor Week
  - 30-40 people in the room
  - ~10 remote
- 2 Condor developers present
  - Greg Thain & Todd Tannenbaum
- Agenda & slides: https://indico.cern.ch/event/272794/
  - Excellent presentations by Greg, Todd and sites (RAL, FNAL, Nebraska…)
  - Summary: https://twiki.cern.ch/twiki/bin/view/LCG/GDBMeetingNotes20141208

HTCondor History

- A more than 25 years old product…
  - Developments driven by community needs is probably the main reason for this success
- Not designed at a batch system to manage clusters
  - An early implementation of the desktop grid/volunteer computing idea: « A Hunter of Idle Workstations » (first presentation of Condor in 88)
  - No concept of queues but a match making between job with requirements and machines offering resources and expressing their usage constraints
- No pre-defined membership to a Condor pool (« cluster »)
  - Machine joins dynamically, handles volatile resources
- Since the origin, focused on « High Throughput Computing »
  - Optimize the sum of completed job run times in a given wall clock interval
  - No attempt to optimize one job runtime (« high performance computing »)

Introductory Sessions

High performance

Introductory Sessions

High throughput

Job Requirements

Jobs state their requirements and preferences, and attributes about themselves:

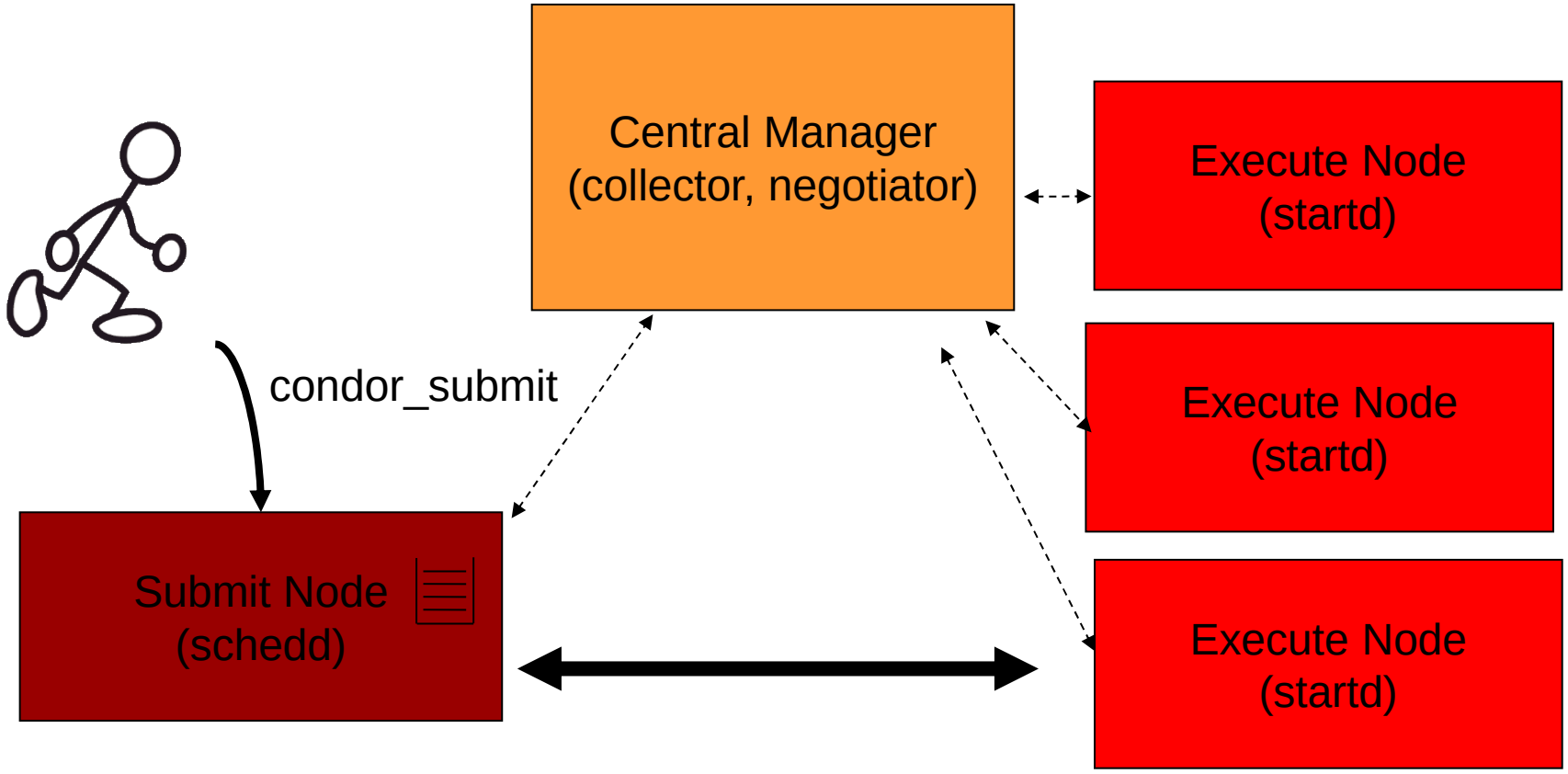
- Requirements:
  - I require a Linux/x86 platform
  - I require 500MB RAM
- Preferences ("Rank"):
  - I prefer a machine in the chemistry department
  - I prefer a machine with the fastest floating point
- Custom Attributes
  - I am a job of type “analysis”
- Everything in ClassAds

Machine Requirements and Resources

- Machines specify:
- Requirements:
  - Require that jobs run only when there is no keyboard activity
  - Never run jobs labeled as “production”
- Preferences ("Rank"):
  - I prefer to run Todd’s jobs
- Custom attributes
- I am a machine in the chemistry department

Match-Making

HTCondor brings them together



Other Key Features…

- Manages data without the need of a shared file system
- Monitorable: lots of information published in ClassAds
- Machine's owner is a King!
  - Owner's policy trumps all
- Support many security protocols, including X509
- Can accomodate any network configuration, including firewalls
- Support DAG workflows
- Partionable slots: multicore slots can be partitioned into several smaller slots
  - condor_defrag allows to drain a partitioned slot to make it available again for multicore jobs
  - Drain a slot, not a machine: better efficiency

… Other Key Features

- User job confinement: protect machine from job and vice versa, job from jobs
  - cgroups
  - File system isolation: jobs see a private mount of file systems like /tmp
  - PID private namespace
- Python bindings
  - Easy scripting of everything, including submission
- Fairshare: user and group, hierarchical
- Demonstrated a very large scalability
  - Up to 200K jobs per schedd machines
  - Scale out possible by adding schedd machines
- A lot of details in the presentations

Site Experiences

- Several sites with already a long experience (FNAL, Milano…)
- Significant (and impressive!) work done at RAL (Andrew Lahiff)
  - Real effort not to reimplement queues in Condor and use the « Condor way »
  - A lot of new ideas: use of condor_roster to start/stop VMs according to load, use of partitionable slots for multicore support
  - Several tools, in particular for monitoring
- Many sites moving or considering moving to it: a real community
- Again look at presentations from the workshop… and today!
- Growing adoption in the WLCG community
  - In particular in the UK after RAL successful migration
  - You will never be alone!
  - Condor community and developers reported to be very responsive

Conclusions

- Condor is a very featureful product, well adapted to our HTC computing models
- Workshop was a very successful event
  - A lot of valuable information from the developers
  - Sites shared their experiences and demonstrated the good integration into the grid infrastructure
- Significant take up in the community
  - Emerged as the most promising open source alternative to Torque/MAUI
  - Several sites with advanced migration projects or prototypes
- An area for tighter collaboration between HEPiX and WLCG
  - Batch systems have always been a traditional HEPiX topic and a lot of experience here
  - WLCG sites under pressure of moving away from Torque/MAUI