

HTCondor European Workshop Summary

[Michel Jouvin](#)

jouvin@lal.in2p3.fr

CNRS/LAL

Workshop Facts

- HTCondor (European) + ARC CE workshop
 - HTCondor: 3 days + 2 days of Office Hours
 - ARC CE : 1 days + 2 days of Office Hours
 - <https://indico.cern.ch/event/467075/timetable/#all.detailed>
- 60 participants in total from ~40 institutions/labs
 - Including a few US and Asia
- 5 HTCondor developers from U. of Wisconsin/Madison
 - Including Miron Livny, HTCondor's father!
- (Unfortunately) less interaction than anticipated between HTCondor and ARC CE workshops
 - Made worse by one of the main ARC CE workshop contributor being sick and absent

HTCondor Workshop Program

- Presentation of HTCondor features by HTCondor developers (+ B. Bockelman)
 - Basic and advanced features
 - Organised by topics: pools, usage/scheduling policies, security, management...
 - Many concrete examples
 - Very good presentations
- HTCondor CE
- Site reports: PIC, CERN, GRIF, DESY
- Experienced user presentations
 - Multicore job support
 - ARC CE integration
 - DIRAC integration
 - CMS Experience with HTCondor and glide-in WMS
- Round tables / panels

HTCondor Philosophy and Design Principles

- HTCondor already 32 years: a DevOps precursor
 - Designed to address a specific need
 - Developed by a core team having strong links with its user community
- HTCondor designed to address the High Throughput Computing (HTC) needs
 - Run in the shortest possible time a large number of independent jobs
 - Different from HPC: make one (parallel) job running as fast as possible
 - HTC is a 24x7 operation requiring a lot of automation
- Key principles: central control, component autonomy
 - Nothing is considered reliable, must handle every kind of failure
 - A common language spoken by all components to exchange information: ClassAds
 - Global information namespace + a central repository: collector
 - Scheduling = match making between job and WN information. No queue.
 - Glide-in overlay to achieve distributed computing efficiency

HPC



From HTCondor Team, 1st HTCondor European workshop, Dec. 2014)

HTC



From HTCondor Team, 1st HTCondor European workshop, Dec. 2014)

Policies

- At the heart of the scheduling: HTCondor alternative to queues
- Defined using REQUIREMENTS and RANK
 - Machines use them to express what they accept/prefer to run
 - Jobs use them to express where they want/prefer to run
- Support for killing a job in favor of a preferred one (aka preemption)
 - Can be used to implement backfilling
- Accounting groups used to define target shares and to implement a fairshare scheduling
 - Hierarchical group/quota definition and enforcement
 - Possible to borrow “unused quota” of another group
- Users can define/change the relative priority of their jobs

Security

- Every communication between HTCondor components can be authenticated
- HTCondor supporting multiple authentication mechanisms
 - File system
 - Password
 - SSL
 - Certificates
 - Kerberos
- User payload never run as root
 - Low-level circuit breaker
- Support for encrypting the working directory
- `condor_ping` command to troubleshoot security problems

Advanced Features

- Python bindings: write your preferred user/admin interface to HTCondor
 - Very Pythonic: very powerful, ClassAds support/building/parsing
 - Lots of potential for sharing
- Monitoring: `condor_gangliad`
 - Originally exported aggregated information from ClassAds into Ganglia: easy to display time series about HTCondor usage
 - Extended as a feed mechanism for database/analytics system, e.g. ELK, Graphite, Influx: producing JSON files
 - Will become `condor_metricd`

Virtualization and Docker

- HTCondor can interact with many different backend computing infrastructures
 - Concept of Universe
 - vm universe has been there for a while, supporting all hypervisor technologies
 - Grid universe can connect to clouds using the EC2 interface
- Docker emerging as the lightweight solution for user payload isolation
 - private root filesystem, separate PID namespace, NATed network, private UIDs
 - Also provides the ability to customize execution environment
 - Docker universe enabled as soon as Docker is installed on the machine
 - A docker job is just a job... usual tools/limits apply
- Other user containment tools
 - Cgroups: cpu, memory,
 - Named chroot
 - Mount under scratch: “private mount” of some file systems

Multicore jobs

- Tradeoff between overall efficiency and MC job turnaround
 - With most batch systems, managed to keep the resulting “inefficiency” reasonable
 - 2 approaches: dedicated slots versus WN (partial) draining
- HTCondor supports “partitionable slots”
 - A way to use multicore slots for single core jobs when there is not enough MC jobs queued
 - `condor_drain` tool to reconsolidate partitioned slots: quite flexible but not very different from the traditional drain approach
- New approach proposed by A. Lahiff (RAL): include preemptable (killable) jobs in the job mix to avoid draining slots
 - Ensure that there is a minimum of number of preemptable jobs running: when a MC job has to run, HTCondor preempts (kill) these jobs
 - When below the threshold of “MC slots”, accept only preemptable jobs on a set of machines
 - Everything controlled by HTCondor policies and scheduling: no external script/tool

Office Hours

- Basically, free consultancy by developers/experts on your particular problems
 - Just needed to turn up in the room where experts were
- Very much appreciated both by people thinking about migration and by experimented people with advanced problems
 - Several of us took this opportunity to bootstrap a HTCondor CE
- Hands on (development/testing) work on particular problem like Kerberos integration

Structuring the Community

- HTCondor community is part of its value
 - Reflected by workshop agenda: 1/3 of the talks given by people from the community
- Currently, mainly through the HTCondor users mailing list
 - Very responsive and valuable to get help
 - Not enough to build “horizontal collaboration” and to achieve tool sharing
 - Growing contributions from the community: Andrew Lahiff’s work is a good example
 - Experience sharing is important for sites considering migration
- Several ideas discussed
 - Stackoverflow
 - GitHub (or similar) project/repository
 - Miron insisted: ensure that we are doing something useful **for small(er) sites**

Conclusion

- Second European workshop: larger participation, increasing interest
 - 1st one in December 2014
 - Focus changed from introduction tutorials to more advanced discussions and experience sharing
 - Wide range of attendee profiles that turned out to be a successful mix, appreciated by everybody
- European community is growing: this kind of European event is very helpful and complementary to the main HTCondor events occurring in the US
- A new workshop foreseen by summer 2017
 - Growing interest/hype in HTCondor CE: could probably one of the hot topic then (hope to see more concrete experiences by then)

HTCondor Community... as seen by Miron!!

