NEC’2019
XXVII International Symposium on Nuclear Electronics & Computing
Montenegro, Budva, Becici, 30 September - 4 October 2019

Contribution ID: **211** Type: **Sectional**

Optimizing resource usage with HTCondor

Tuesday, 1 October 2019 17:30 (15 minutes)

HTCondor is a very flexible job management system, but for site administrators it is not always easy to come with optimal configuration to fulfill local policies and requirements. Everybody would expect that normal job execution follow fairshare configuration and recent resource usage, but with few additional quite natural requirements like minimum idle resources it can pretty fast become difficult to achieve all these very simple goals together.

HTCondor batch system by design maximize utilization of all resources and this approach works fine for jobs with same resource requirements (CPU, memory, …). With a mixture of smaller and bigger jobs available resources can be sufficient to start only the small job while big job will wait idle almost indefinetely. Condor can runs DEFRAG daemon to consolidate fragmented resources, but this often lead to unnecessary high number of idle resources.

Several different approaches exists to optimize draining and consolidation of the fragmented resources, but most of them focus just on grid multicore jobs. We think this is unnecessary restriction especially for our local users and that lead to the implementation of our own mechanism for machine draining to support arbitrary multi-dimensional resource requirements while trying to minimize draining time and optimize resource utilization. Details about our draining mechanism will be presented and compared with of other solutions.

Summary

Effective job execution with non-standard resource requirement within HTCondor batch system.

Primary author: Mr VOKAC, Petr (Institute of Physics of the Czech Academy of Sciences)

Presenter: Mr VOKAC, Petr (Institute of Physics of the Czech Academy of Sciences)

Session Classification: Distributed Computing. GRID & Cloud computing

Track Classification: Distributed Computing. GRID & Cloud Computing