Contribution ID: **327** Type: **poster**

# Rapid-response Adaptive Computing Environment

*Wednesday 5 September 2007 08:00 (20 minutes)*

We describe the ideas and present performance results from a rapid-response adaptive computing environment
(RACE) that we setup at the UW-Madison CMS Tier-2 computing center. RACE uses Condor technologies to allow
rapid-response to certain class of jobs, while suspending the longer running jobs temporarily. RACE allows us to use
our entire farm for long running production jobs, but also harness a portion of it for unpredictable shorter period
user analysis jobs. RACE features are ideal at Tier-2 computing centers where farm usage will become less than
optimal if a portions of the farm are dedicated to long and short queues.

## Summary

RACE uses Condor technologies to allow rapid-response to a chosen set of jobs, while suspending the longer running jobs temporarily. We have explored two mechanisms, one that is based on computing-on-demand implementation that does not have queueing and another that uses a parallel scheduler. Both mechanisms use
the operating system services to suspend and release the existing job process. The suspended jobs free-up both
CPU and memory, so the new jobs have access to the complete resources of the system. There is a period of time
during which there is some contention for resources. The Condor computing-on-demand implementation minimizes this contention, but it does not provide any accounting nor prioritization of new jobs. We have used
computing-on-demand with PROOF. After some improvements to Condor and PROOF classes, we were satisfied
with job suspension and resumption times. We will present latency and resumption time results. However, we
were not happy with the restricted services on both Condor job scheduling, monitoring and accounting side, and
by the PROOF limitation of the analysis jobs to those written in ROOT framework only. Therefore, we have explored an alternate mechanism using multiple schedulers for the same set of virtual machines. Condor was configured such that when higher priority scheduler has jobs to run, it suspends the normal priority jobs. This way both schedulers provided complete Condor services. When the higher priority jobs are done, the normal priority jobs resumed. We have tuned the scheduler performance so that the mechanism can be used in practice.
We will also present timing results for this setup.

For high-energy physics usage, large numbers of long running production jobs can be submitted to the normal priority scheduler, and the ephemeral and chaotically appearing analysis jobs to the high priority scheduler. This
way the usage of the computing farms is maximized, and the analysis jobs get processed rapidly. We have written
simple scripts that automatically divide the job into small chunks so that large datasets can be processed in a

distributed way in a short amount of time. We will provide statistics of usage on our farm where CMS simulation
production and CMS high-level trigger exercise related analysis jobs were processed. We will also provide other
ideas for configuration or multi-scheduler Condor operational environments.

**Primary authors:** Mr LAZARIDIS, Christos (UNIVERSITY OF WISCONSIN); Mr BRADLEY, Dan (UNIVERSITY OF WISCONSIN); Prof. DASU, Sridhara (University of Wisconsin); Mr MEHTA, Vishal (UNIVERSITY OF WISCONSIN)

**Co-authors:** Dr MOHAPATRA, Ajit (UNIVERSITY OF WISCONSIN); Mr MAIER, William (UNIVERSITY OF WISCONSIN)

**Presenter:** Prof. DASU, Sridhara (University of Wisconsin)

**Session Classification:** Poster 2

**Track Classification:** Distributed data analysis and information management