Contribution ID: **224** Type: **Oral**

Opportunistic Computing with Lobster: Lessons Learned From Scaling up to 25k Non-Dedicated Cores

Monday, October 10, 2016 2:00 PM (15 minutes)

We previously described Lobster, a workflow management tool for exploiting volatile opportunistic computing resources for computation in HEP. We will discuss the various challenges that have been encountered while scaling up the simultaneous CPU core utilization and the software improvements required to overcome these challenges.

Categories: Workflows can now be divided into categories based on their required system resources. This allows the batch queueing system to optimize assignment of tasks to nodes with the appropriate capabilities. Within each category, limits can be specified for the number of running jobs to regulate the utilization of communication bandwidth. System resource specifications for a task category can now be modified while a project is running, avoiding the need to restart the project if resource requirements differ from the initial estimates. Lobster now implements time limits on each task category to voluntarily terminate tasks. This allows partially completed work to be recovered.

Workflow dependency specification: One workflow often requires data from other workflows as input. Rather than waiting for earlier workflows to be completed before beginning later ones, Lobster now allows dependent tasks to begin as soon as sufficient input data has accumulated.

Resource monitoring: Lobster utilizes a new capability in Work Queue to monitor the system resources each task requires in order to identify bottlenecks and optimally assign tasks.

The capability of the Lobster opportunistic workflow management system for HEP computation has been significantly increased. We have demonstrated efficient utilization of 25K non-dedicated cores and achieved a data input rate of 9 Gb/s and an output rate of 400 GB/h. This has required new capabilities in task categorization, workflow dependency specification, and resource monitoring.

Primary Keyword (Mandatory)

Distributed workload management

Secondary Keyword (Optional)

Tertiary Keyword (Optional)

Primary authors: WOODARD, Anna Elizabeth (University of Notre Dame (US)); WOLF, Matthias (University of Notre Dame (US))

Co-authors: YANNAKOPOULOS, Anna (Notre Dame / Florida State University); Dr TOVAR, Benjamin (University of Notre Dame); THAIN, Douglas (University of Notre Dame); HURTADO ANAMPA, Kenyi Paolo (University of Notre Dame (US)); LANNON, Kevin Patrick (University of Notre Dame (US)); HILDRETH, Mike (University of Notre Dame (US)); DONNELLY, Patrick (University of Notre Dame); Dr BRENNER, Paul (University of Notre Dame); LI, Wenzhao (University of Notre Dame)

Presenters: WOODARD, Anna Elizabeth (University of Notre Dame (US)); WOLF, Matthias (University of Notre Dame (US))

Session Classification: Track 3: Distributed Computing

Track Classification: Track 3: Distributed Computing