



Contribution ID: 47

Type: **Oral presentation to parallel session**

Minimizing draining waste through extending the lifetime of pilot jobs in Grid environments

Thursday, 17 October 2013 12:06 (22 minutes)

The computing landscape is moving at an accelerated pace to many-core computing.

Nowadays, it is not unusual to get 32 cores on a single physical node.

As a consequence, there is increased pressure in the pilot systems domain to move from purely single-core scheduling and allow multi-core jobs as well.

In order to allow for a gradual transition from single-core to multi-core user jobs, it is envisioned that pilot jobs will have to handle both kinds of user jobs at the same time, by requesting several cores at a time from Grid providers and then partitioning them between the user jobs at runtime.

Unfortunately, the current Grid ecosystem only allows for relatively short lifetime of pilot jobs, requiring frequent draining, with the relative waste of compute resources due to varying lifetimes of the user jobs. Significantly extending the lifetime of pilot jobs is thus highly desirable, but must come without any adverse effects for the Grid resource providers.

In this paper we present a mechanism, based on communication between the pilot jobs and the Grid provider, that allows for pilot jobs to run for extended periods of time when there are available resources, but also allows the Grid provider to reclaim the resources in a short amount of time when needed. We also present the experience of running a prototype system using the above mechanism on a couple US-based Grid sites.

Primary authors: BOCKELMAN, Brian Paul (University of Nebraska Lincoln); WURTHWEIN, Frank (University of California San Diego); SFILIGOI, Igor (University of California San Diego); MARTIN, Terrence (University of California San Diego)

Presenter: SFILIGOI, Igor (University of California San Diego)

Session Classification: Distributed Processing and Data Handling B: Experiment Data Processing, Data Handling and Computing Models

Track Classification: Distributed Processing and Data Handling B: Experiment Data Processing, Data Handling and Computing Models