



Contribution ID: 96

Type: oral presentation

## A history-based estimation for LHCb job requirements

*Monday, April 13, 2015 2:45 PM (15 minutes)*

The main goal of a Workload Management System (WMS) is to find and allocate resources for the jobs it is handling. The more and more accurate information the WMS receives about the jobs, the easier it will be to accomplish its task, which will directly translate into a better utilization of resources. Traditionally, the information associated with each job, like expected runtime or memory requirement, is in the best case defined at submission time by the Production Manager or fixed by default to arbitrary conservative values. In the case of LHCb's Workload Management System, no mechanisms are provided that automatize the estimation of job requirements. As a result, in order to be conservative, much more CPU time is normally requested than actually needed. Particularly, in the context of multicore jobs this represents a major problem, since single- and multi-core jobs shall share the same resources. Therefore, in order to allow an optimization of the available resources, an accurate estimation of the necessary resources is required. As the main motivation for going to multicore jobs is the reduction of the overall memory footprint, the memory requirement of the jobs should also be correctly estimated.

A detailed workload analysis of past LHCb jobs will be presented. It includes a study of which job features have a correlation with runtime and memory consumption. Based on these features, a supervised learning algorithm has been developed relying on a history-based prediction. The aim is to learn over time how jobs' runtime and memory evolve due to changes in the experimental conditions and the software versions. It will be shown that this estimation can be notably improved if the experimental conditions are taken into account.

**Primary author:** RAUSCHMAYR, Nathalie (CERN)

**Presenter:** RAUSCHMAYR, Nathalie (CERN)

**Session Classification:** Track 4 Session

**Track Classification:** Track4: Middleware, software development and tools, experiment frameworks, tools for distributed computing