WLCG Multicore Deployment task force meeting on April 8th.

Attendance: A. Sedov, A. Lahiff, A. McNab, D. Traynor, J. Templon, J. Belleman, J. Hernandez, M. Alef, T. Hartmann, A. Forti A. Perez-Calero.

Meeting dedicated to a summary session concerning the experience discussed in the first months of activity.

Initial discussion regarding the validity of jobs running time estimations in relation to input waiting time caused for example from jobs running with xrootd dependance. It is clear that the estimation is substantially affected in such cases. This is of course not exclusive for multicore, however its impact on the efficient usage of resources is stronger when dealing with a mixture of single core and multicore jobs.

Slides by Alessandra and Antonio with the summary of the main points discussed so far in the task force, followed by comments.

Regarding job running time estimation, the CMS and ATLAS model differ in that it is essentially a different problem. ATLAS needs to estimate the lifetime of the pilot depending on the actual running time of the payload, as one pilot essentially pulls one job before exiting the batch system. ATLAS proposes to use a variety of queues for jobs with diverse running times. The pilots will then provide the queue limit from where the payload comes as running time estimate for the remote batch system in order to help scheduling at the sites.

The argument is reversed for CMS pilots, which will try to use the total length allowed by the queue in order to continue pulling payload jobs for as long as they can run.

The fundamental problem with scheduling multicore jobs seems to be twofold: how to create multicore slots by cleverly draining some resources and how to keep the multicore slots alive so that no additional draining is needed.

It is pointed out that solving the second part for a steady flow of multicore jobs immediately makes the first one irrelevant. This is in line with the CMS model, which tries to keep multicore slots always occupied, even if with multiple single core jobs running inside a multicore pilot.

However Jeff mentions that this is not in general a desired solution as it would remove entropy from the site, removing flexibility, which helps to serve multiple VOs in a shared site. For example. it would make the opportunistic usage of resources to be more difficult, due to the slower rotation of jobs. A certain VO would not be allocated idle machines pledged to other VO if it is known that they could not be taken back for a quite long period.

Concerning the experience with HTCondor at RAL, Antonio asks if any matching expression which maximizes the chance of multicore jobs being used to fill multicore slots has been introduced. This is not the case. The results could potentially benefit from using such configuration, as it would ensure that no additional resources need to be drained, thus no constant draining would be required in order to provide resources to an stable flow of multicore jobs.

With respect to text in slide 7, Jeff points out that job running time predictability being affected by variable luminosity and event complexity actually refers to real data reconstruccion. In contrast, running time for Monte Carlo production jobs should be easy to estimate.

As a final remark to slide 7, Jeff disagrees with the text regarding the usability of the middleware to pass job requirements to the remote batch systems. His point is that the MW itself works, however is the sites responsibility to ensure that it is used properly by providing instructions adequate for their particular CE and batch system.

This topic: LCG > Minutes20140408
Topic revision: r2 - 2014-04-15 - AntonioPerezCalero