

Fair Grid Scheduling

Tuesday, February 12, 2008 4:00 PM (0 minutes)

Jobs durations vary a lot among groups or users depending of the kind of applications. For instance a group dedicated to test the middleware have 90% of their jobs of duration less than 5 minutes. Biomed jobs have a quarter of jobs running less than 2 minutes, another quarter between 2 minutes and 8 hours, another quarter between 8h and one day and the last quarter between one day and 3 days. Other groups have many jobs running up to 3 days.

Each site have to schedule jobs coming from the grid in an efficient and fair way. Site needs sometimes to be able to justify part of resources granted to groups or users.

Consequences of unfairness are possible leaving of groups or users. These differences in the way jobs are scheduled could lead to unfair treatment of users or groups. We propose to take fairness into account when dealing with multi-users scheduling problems and to seek how scheduling could be improved in order to be fair.

4. Conclusions / Future plans

Multi-user scheduling is a NP-hard problem as soon as we have only one machine with 2 users and release dates. For the moment , we have studied fair on-line storage allocations and described an algorithm able to list all possible solutions. We still have to study the online case with unknown duration in order to propose algorithm suitable for the grid.

Provide a set of generic keywords that define your contribution (e.g. Data Management, Workflows, High Energy Physics)

Grid scheduling, Fairness, Workflows,

3. Impact

As in load balancing resources may be badly distributed at the expense of everyone. The objective is not to starve users but on the contrary to maximize fairness between users. How to measure the quality of a schedule with respect to fairness ? One criterion is evaluated for all users, this leads to a vector of evaluation indexed by users. We have to compare these vectors with an order expressing fairness. An order is fair is and only if it is symmetrical and if it is decreasing if 2 values are changed so that their minimum decrease. For instance [5, 3, 2, 8] is less fair than [5, 4, 2, 6] because of the second user. By maximizing such order, it is impossible to increase an user without decreasing another user who has less than the first one. Such order is the Leximin ordering and cannot be represented with a scalar function. This means that we cannot always obtain fair solutions when maximizing a scalar function in a multi-user setting.

URL for further information:

<http://clrwww.in2p3.fr/>
www.isima.fr/limos/

1. Short overview

Computing grids allow sharing of distributed resources such as CPU or storage to the service of many scientific communities. Users submit irregular bursts of jobs. This irregular usage of the grid allow an efficient sharing of resources because unused resources can be

collected this way. Jobs are mono-processors and arrive on-line to clusters. Jobs on a cluster are independents, they could be run in any order. Pre-emption is not used because of memory and communication costs.

Primary author: Mr MEDERNACH, Emmanuel (LPC - Clermont, France)

Presenter: Mr MEDERNACH, Emmanuel (LPC - Clermont, France)

Session Classification: Posters

Track Classification: Existing or Prospective Grid Services