PROCEEDINGS OF SCIENCE

Studies of the performances of an open source batch system /scheduler (TORQUE/MAUI) implemented on a middle sized GRID site.

Leonello Servoli[*a], **Francesco Cantini**[a], **Mattia Cinquilli**[ab], **Mirko Mariotti**[b], **and Claudio Tanci**[b]
[a] *INFN - Perugia*
Via A. Pascoli, 06123 Perugia (Italy)
[b] *Physics Department - University of Perugia*
Via A. Pascoli, 06123 Perugia (Italy)
E-mail: leonello.servoli@pg.infn.it.

Open source computing clusters for scientific purposes are growing in size, complexity and eterogeneity; often they are also included in geographically distributed computing GRID. In this framework the difficulty of assessing the overall efficiency, identigying the bottlenecks and tracking the failures of single components is increasing continously. In previous works we have formalized and proposed a set of metrics to make such an evaluation and built an analysis and simulation infrastructure to find a quantitative measure of the efficiency of a TORQUE/Maui based system. The main idea is to use the Maui internal simulator, fed by workloads produced by a real computing cluster to test several scheduler configurations to choose the optimal solution. The analysis presented in this work is based on the comparison of real data and simulated data, the latter assumed as maximum theoretical limit.

13th *International Workshop on Advanced Computing and Analysis Techniques in Physics Research*
February 22-27 2010
Jaipur, India

[*]Speaker.

PoS(ACAT2010)042

Contents

1. Introduction

The INFN Perugia unit hosts a computing infrastructure to satisfy part of the scientific computing needs of the local research groups. The computing cluster (> 200 cores/CPUs, > 40 TB disk space) have been built during the past 4 years using all the available resources from each group, hence the hardware is heterogeneous. Besides the cluster belongs to the WLCG [1] computing grid, built to satisfy the needs of the LHC experiments, who dictates the Operating System (Linux SLCx) and Batch System /Scheduler environments (Torque/MAUI).

The Virtual Machine paradigm has been used (Xen Paravirtualization) whenever needed to solve compatibility problems between hardware and Operating System or among different computing environments requested by different users. The access to the facility should then be granted either to the local users than to the WLCG grid requests, with some local users who could use both channels to submit jobs.

The task of defining the policy to use the computing resources is a complex one due to the time varying user requests and to the conflicting priorities of the research groups. As an example, one group could ask to use for a short period of time as many job slots as possible to produce a certain result, while for the rest of the year its requests could be relaxed. So it is important to understand how to satisfy the requests, changing the scheduler configuration, and also how the other groups will be afffected by these changes. The basic tool that has been used to answer this problem is the MAUI simulator, which allows to simulate the response of a given batch system. In fig. 1 the flow of the analysis is described: real (or simulated) job submission patterns during a certain time interval are fed to the MAUI simulator together with the configuration parameters; the results are used to compute the specific optimization metrics; the process is repeated, in an automated way, using the same input data and varying the scheduler configuration parameters over the allowed parameters space. The behaviour of the system is studied and the optimal solution for the specific case is choosen.

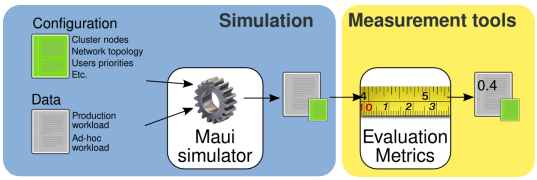


Figure 1: Conceptual schema of the workflow.

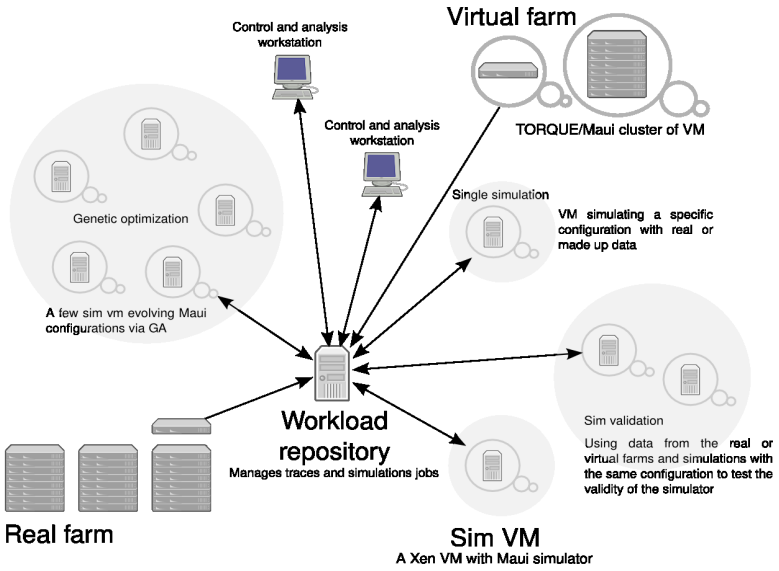


Figure 2: Architecture of the infrastructure to automate the data treatment, with all the relevant data flows.

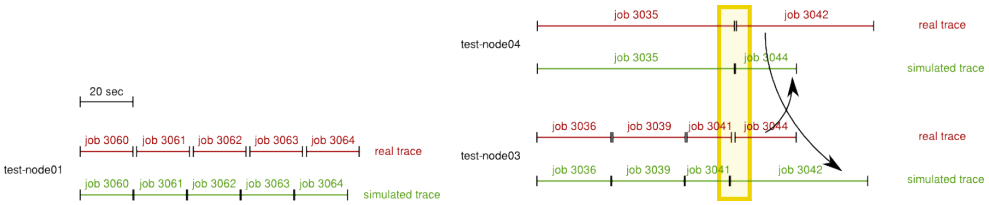
2. The infrastructure and its implementation

To fully exploit the potential of the MAUI simulator an infrastructure for the automatic generation of different set of configuration parameters and for the subsequent execution of the simulation was needed. Hence a flexible and complex hardware infrastructure has been built (Fig. 2):

- the Real Farm, the actual computing facility;
- a test-bed, the Virtual Farm, (4+1 virtual machines hosting the TORQUE + MAUI system) to generate workloads according to controlled patterns and conditions;
- a workload repository (the core of the structure) to hold workload traces from the real cluster and the test-bed, and all the simulations directives;
- few virtual machines equipped with Maui to run the simulations.

A set of software tools was developed to manage pre and post-simulation steps, to automatically generate cluster and uses-cases configurations, to arrange for batches of simulations and real scheduling cases, to retrieve and submit simulations jobs and finally to analyze the results.

The Maui simulator [2] is a special run mode of Maui developed to reproduce the behaviour of a real MAUI scheduler starting from configuration and workload data. A first phase of tests



(a) (b)

Figure 3: (a) Comparison among real (red) and simulated (green) data. A time overhead in the real system, due to hardware delays and time quantization, is observed. (b) Description of the node swap mechanism; one job executed in the real cluster on one node, in the simulated one is assigned to another node.

has been carried out, using a virtualized test batch system, in order to identify its peculiarities and limits. Fig. 3a (red lines refer to real data and green ones to simulated data) illustrate the comparison among the real behaviour in the test system and the simulated one. It can be appreciated that the simulation does not reproduce the time overhead due to the hardware delays in the real system. This difference can be easily neglected because we are not interested in the exact time ordering of the jobs and the time shift due to time quantization is of the order of 1 sec, i.e. negligible with respect to the duration of normal batch jobs. In fig. 3b a node swap between real and simulated data is also illustrated; again this effect is not relevant, because we are not interested to the exact node that has been assigned to execute a certain job.

To evaluate each configuration and to compare its effects a specific metric (*global efficiency*) has been defined [3]: when a job is queued, its waiting is justified only if there are no free resources. Global efficiency quantifies if the waiting time is justified; the *job efficiency* is defined according to eq. (2.1):

$$E_{job} = 1 \;\; if\ the\ waiting\ time\ is\ 0;\;\; E_{job} = \frac{\sum_{i=1}^{m} E_{\varepsilon i}\Delta T_i}{\sum_{i=1}^{m} \Delta T_i} \;\; in\ all\ other\ cases. \tag{2.1}$$

where $E_{\varepsilon i} = N_{job}/N_{resources}$ is the ratio between the number of jobs running in the system in a given time interval ΔT_i and the total of the available resources; the sum runs over all the time intervals (0 to m) where the job is in queued state. In eq. (2.2) the total efficiency for the computing cluster over a given period is defined as:

$$E_{total} = \frac{\sum_{i=1}^{n} (E_{job})_i \Delta Q_i}{\sum_{i=1}^{n} \Delta Q_i} \tag{2.2}$$

where n is the number of jobs present in the batch system in the considered period; each job efficiency is weighted using its queue time, ΔQ_i. The total efficiency has values between 0 and 1, being a value close to one an indication that all the cluster's resources have been used in a close to optimal way.

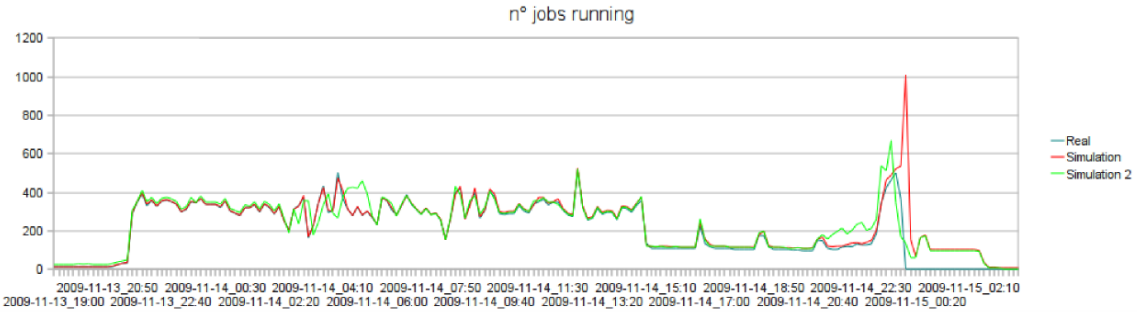


Figure 4: Number of running jobs during 10' time interval. Real data (blue), simulated data using fair-share policy (red), simulated data using a static policy (green).

3. Preliminary Analysis

The entire simulation procedure has been tested using a real data sample coming from the production farm during 33 hour time interval. In the following plots are shown the real data (blue line), the simulated data with the same Maui configuration (*fair-share*) used in the real case (red line) and the simulated data where the configuration used for MAUI implements a *static policy* where each group has a weight proportional to the amount of contributed resources(green line).

In Fig. 4 the number of jobs running during a 10' time interval is shown; it can be appreciated that no big discrepancy between real and simulated data exists, even for different MAUI configurations. This is due to the constant number of available CPUs.

In Fig. 5 the number of jobs in queue during the same 10' period is shown; now a discrepancy starts to build between real and simulated data when the number of queued jobs grows over 2000. The static policy simulation (green) shows a better behaviour than the fair-share one (red) reducing the number of queued jobs, suggesting a possible reduction of the average job queue time.

In Fig. 6 the global efficiency has been plotted. A big discrepancy between real data and simulated ones with the same configuration, and the simulation with the static policy configuration could be observed from the beginning; the static policy, if used, would have granted a better use of the system. This is confirmed by Fig. 7 where is plotted the time spent in queue by each job. The advantage obtained switching from fair-share (2450 seconds) to static policy (2282 seconds) configuration is evident.

4. Conclusion

A full architecture to study the behaviour of a MAUI scheduler with different configurations on the same usage pattern has been implemented for a middle sized computing cluster with a mixed access pattern (local/GRID). A suitable metric (global efficiency) has been defined to compare the performance of the system. The initial data analysis has shown the potential for such a tool in finding a configuration to implement a predefined resource allocation policy allowing at the same time to maximize the use of the system.

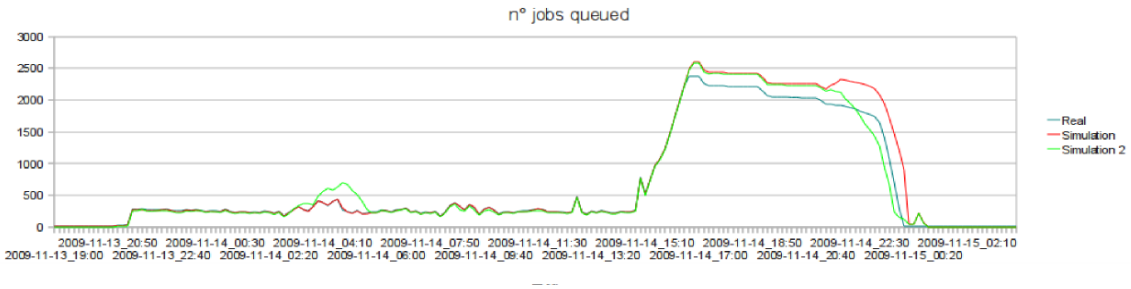


Figure 5: Number of queued jobs during 10' time interval. Real data (blue), simulated data with fair-share policy (red), simulated data with a static policy configuration (green).

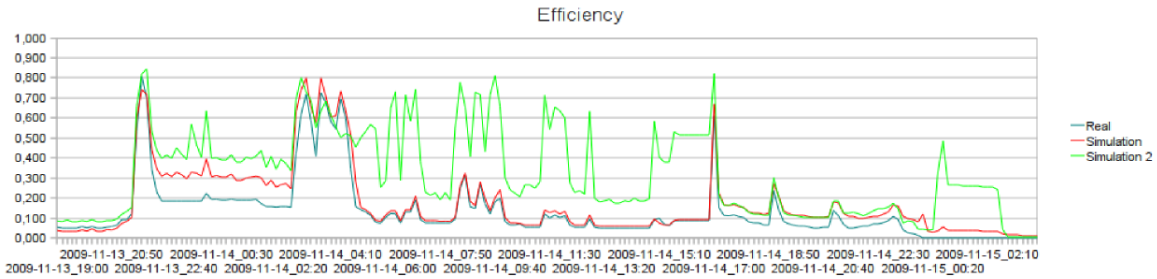


Figure 6: Global efficiency of the system: real data (blue), simulated data with fair-share policy (red), simulated data with a static policy configuration (green).

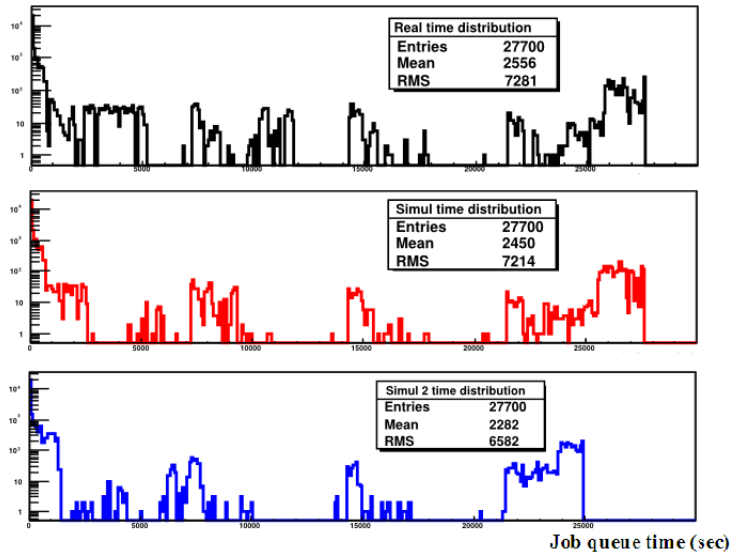


Figure 7: Job queue time distribution for real data (black), fair-share policy simulation (red), static policy simulation (blue).

References

[1] LHC Computing Grid `http://cern.ch/LHCGrid/`

[2] D. B. Jackson, H. L. Jackson, Q. O. Snell, *Simulation Based HPC Workload Analysis* 15th International Parallel and Distributed Processing Symposium (IPDPS'01), vol. 1,(2001) pp.10047a,

[3] L. Servoli, F. Cantini, M. Mariotti, and Claudio Tanci, *Development of a tool to optimize the performance of a Maui Cluster Scheduler* Nuovo Cimento C Vol. 032 (2009) 173-178.