

Development of a tool to optimize the performance of a Maui Cluster Scheduler

F. Cantini¹, M. Mariotti², L. Servoli¹, C. Tanci²

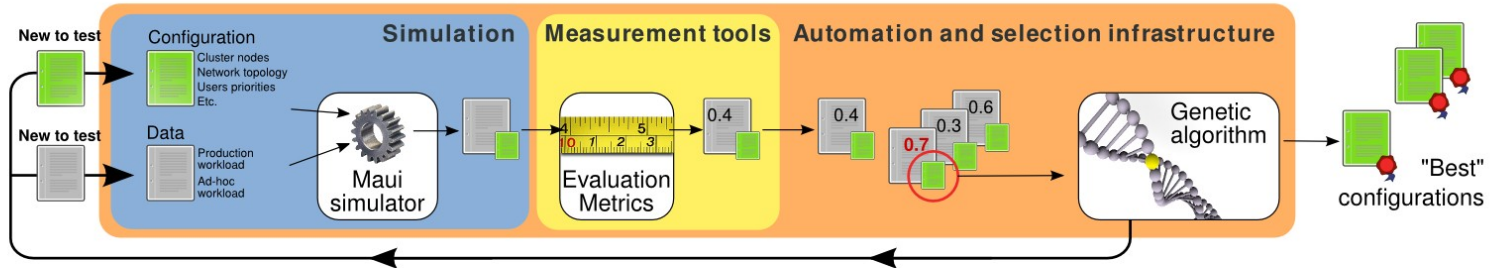
1) INFN - Sezione di Perugia; 2) Dipartimento di Fisica - Università di Perugia

In our Physics Department we have a **Batch System** based on a **GNU/Linux cluster** (about 200 CPUs) and Torque/Maui as Resource Manager/Scheduler. Our farm has some non standard characteristics:

- It's a GRID site connected with the EGEE / WLCG european infrastructure.
- It serves both GRID jobs and local jobs.
- It has been built with contributions of different local groups (that work on different physics experiments).
- It uses virtualization technologies.

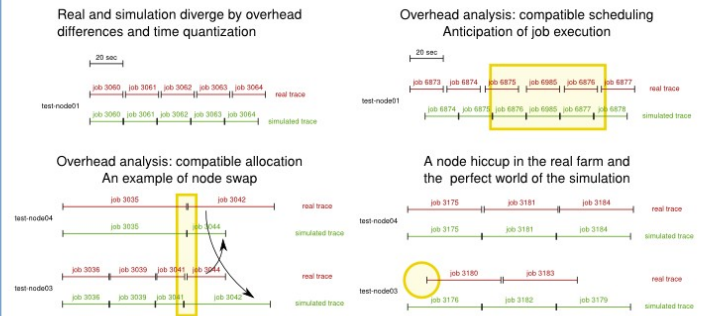
In this scenario the scheduling policies are a key point of the whole infrastructure: there are many needs to be satisfied and often requirements are not compatible one another. The number of variables that influence the scheduling behaviour is so high that the optimization is an hard task.

This research is to assess via a genetic algorithm the goodness of different scheduling policies; this has been achieved using Maui internal simulator (fed on workload either from the real cluster than from an ad hoc one) to test several scheduler configurations. In our implementation, the goodness of each configuration is evaluated using measurement tools based on specifically developed metrics. We have set up an automatic procedure to find the best configuration based on genetic algorithm that creates the scheduler configuration to be used by the simulator to process the data; the output is then evaluated using the metrics and the results are sent to the genetic algorithm to start a new cycle.



Inspection and validation of the simulator

The Maui simulator was put through a control and validation stage with the help of a **virtualized test batch system** in order to identify its peculiarities and limits. Some differences between **real** and **simulated** cases have been found as shown in the following images where they are represented for each node time lines of jobs for the same workload.



Anyway the validation tests proved that Maui simulator peculiarities do not influence in a significant way its ability of prediction.

Evaluation metrics

Fairness metric

Every user and group is entitled to use its own share of hardware resources; *fairness* measures how much the system succeeds in ensuring this goal.

$$Eq_i = 1 - \frac{\min(Q, res - R)}{res}$$

res = each user share of resources
Q = jobs at any time in queue
R = jobs running

$$\Delta T_i$$

time interval

$$n$$

number of time interval

$$Eq = \frac{\sum_{i=1}^n Eq_i \Delta T_i}{\sum_{i=1}^n \Delta T_i}$$

fairness

Efficiency metric

When a job is queued, its waiting is justified only if there are no free resources. *Efficiency* quantifies if the waiting time is justified.

$$Eff_{job} = \frac{jobs}{procs}$$

jobs in execution per processors available in the i-time interval

$$\Delta T_i$$

time interval

$$\Delta Q_i$$

time spent by a job in the queue

$$Eff_{job} = 1$$

if queue time = 0

$$Eff_{job} = \frac{\sum_{i=1}^m Eff_{job,i} \Delta T_i}{\sum_{i=1}^m \Delta T_i}$$

otherwise

$$Eff = \frac{\sum_{i=1}^n Eff_{job,i} \Delta Q_i}{\sum_{i=1}^n \Delta Q_i}$$

efficiency

Fitness metric

Fitness is used as a guide for the convergence of the genetic algorithm. In this specific case it measures the degree to which the algorithm achieves the goal to minimize the difference in the average queue time between jobs of different users. It takes values in the range 0..1, with 1 for a null difference and a perfect result.

$$n$$

number of jobs of user i

$$Q_{ij}$$

time in queue for job j of user i

$$\bar{Q}_i = \frac{\sum_{j=1}^n Q_{ij}}{n}$$

average queue time of process for user i

$$u$$

number of users

$$n, m$$

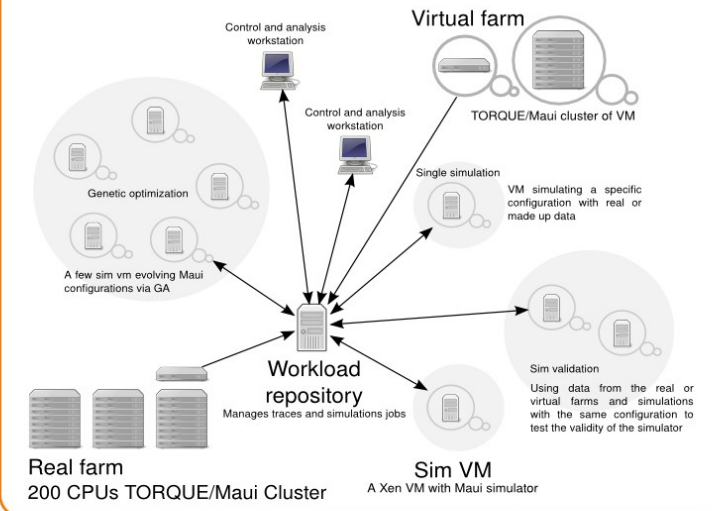
indexes of users

$$Fitness = 1 - \frac{\sum_{n=1}^u \sum_{m=1}^u |\bar{Q}_n - \bar{Q}_m|}{2}$$

fitness function

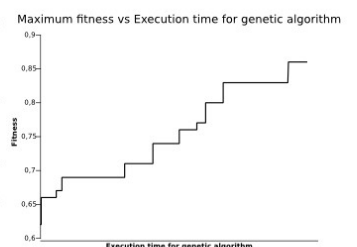
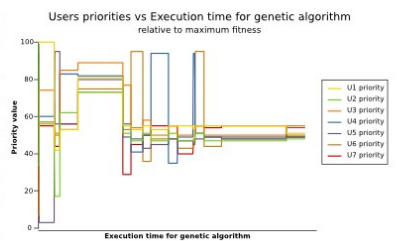
Infrastructure for automation

An infrastructure for automation was needed to efficiently exploit the simulator: in the hardware side a virtual TORQUE/Maui cluster (**virtual farm**) was set up as a test bed to study use patterns, together with a **workload repository** to hold workload traces and simulations directives and a few **virtual machines** equipped with Maui to run the simulations. In the software side a set of 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.



Early analysis: Simple evolution of user priorities

A simple test case was selected to verify that both simulation framework and **genetic algorithm** work properly: we simulated a batch system of 4 nodes, 1 processor per node with 8 users, each submitting the same job sequence. The priority for the first user was fixed at a value of 50, while the others were free to evolve in the range of 1..100; we implemented a genetic algorithm to minimize the difference in the average queue time between all the users. The expected best configuration was the one with the evolving users priorities all matching the value of 50.



Analyzing the trend of the best fitness versus time of execution we can observe fitness metric approaching the value of 1 as expected.

Next steps

A **refinement of the metrics** is in progress to quantify different aspects of quality of service concerning the batch system. A **further validation** of Maui simulator using **Monte Carlo** methods is ongoing. The **optimization of the real cluster** will be carried out together with the **monitoring** in quasi real-time of the system performances.

Presented at **Calcolo Scientifico nella Fisica Italiana - CSFI08**
Rimini May 27 - 30, 2008
Senigallia May 31, 2008