The UERJ T2-HEPGRID BRASIL

A. Santoro, A. Sznajder, J. A. Sanches, E. Revoredo, D. Damião, A. Pereira , R. Saucedo, C. Borges

Departamento de Física Nuclear e de Altas Energias, Instituto de Física
Universidade do Estado do Rio de Janeiro - UERJ

INTRODUCTION

The concept of a HEPGRID (High Energy Physics GRID) was conceived for the needs of the new experiments of LHC (Large Hadron Collider), at CERN, the particle accelerator where the four main detectors will work, ATLAS, CMS (Compact Muon Solenoid), LHCb and ALICE. Many countries from Europe and the United States decided to build their own sites to be part of the HEPGRID world.

Our UERJ-T2 HEPGRID BRASIL is one of those sites, and it is working connected with the Tier 1 of FERMILAB, federated with CALTECH tier 2. Our T2 will serve to CMS and Dzero experiment at Fermilab (Chicago, US). Furthermore, T2-HEPGRID BRASIL has just been accepted as an OSG institution. OSG is a project to build a Grid environment in order to provide the infrastructure and services not only for LHC/CMS production, but also for another scientific experiments.

Figure 1 – The T2 HEPGRID BRASIL room at UERJ

STRUCTURE

The T2 HEPGRID-BRASIL structure contains 174 Intel 2.66 Xeon CPUs, a 2.0 Tbyte RAID disk array and 12 smart-UPS devices. It currently consists in two clusters: one is called *Production Cluster* and the other is called *Development Cluster* (figure 2). Both of them are externally connected to our new RNP/ANSP/AMPATH 1.2 Gigabit link and to the RNP Giga network.

The Production Cluster is composed by one central node, the *prod-frontend*, five Gigabit switches (four *peripheral* switches connected to a *central* Gigabit switch) and 84 (168 processors) workhorse, or *compute* nodes. This is the cluster where experiments run to produce valid results.

The Development Cluster is used for testing new installed/developed helper applications' behavior in a distributed environment. It is composed by a central node, the *devel-frontend,* a 10 Mbit switch and three compute nodes (6 processors).

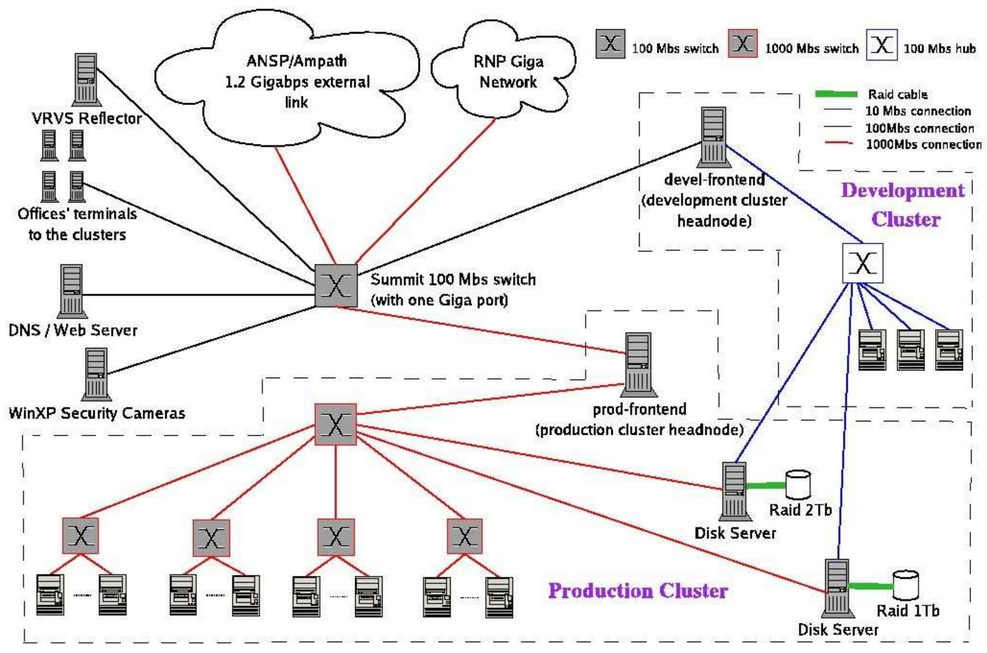

Figure 2 – T2 HEPGRID structure

Two *Disk Servers* are shared between the two clusters. They are nodes that mount the RAID devices, which contain all the applications and experiments' results. The *Disk Servers* communicate with the *prod-frontend (*and its 84 nodes) via the central switch. Also, they communicate with *devel-frontend* (and its 3 nodes) via a 100Mbits hub.

BASIC SOFTWARE AND GRID MIDDLEWARE

The operating system we are currently using is Linux Rocks version 3.3.0. It is based on Red Hat Enterprise Linux 3 (kernel 2.4.21-20) and totally free.

Last year, UERJ-HEPGRID was accepted as an OSG (Open Science Grid) member. The Open Science Grid is a grid computing infrastructure that supports scientific computing via an open collaboration of science researchers, software developers and computing, storage and network providers. In order to set up a OSG Resource, it is necessary to install and configure the OSG middleware software provided at http://osg.ivdgl.org/twiki/bin/view/Integration/OSGCEInstallGuide.

OSG software includes:

- Grid services, like Condor and Globus;
- Software tools developed by the Griphyn project to work with virtual data, particularly the Virtual Data System;
- Utilities like MonaLisa, the monitoring software for the sites.

Our version of OSG package is 0.2.1. As a new OSG resource, T2-HEPGRID has been already serving many other related experiments' jobs, like SDSS (Sloan Digital Data Survey) and GADU (for computational biology)

As we also serve Dzero at Fermilab, the SAM/SAMGRID softwares were needed to prepare our grid for becoming a SAMGRID resource.

CMS SOFTWARE

Our first production tasks for CMS were to simulate two different processes for CMS/LHC experiment, diffractive J/ψ and $t\bar{t}$ production, respectively. In order to do that, we used FAMOS, which is a dynamically configurable system for fast Monte Carlo simulation and reconstruction that has been developed for CMS to allow fast studies of large samples of Monte Carlo events. This application belongs to the CMS Core & Computing Software package (CCS).

In the production environment, we used OSCAR 3.6.5 and ORCA 8.7.3 for simulation and reconstruction, respectively. Those applications are included in CCS as well.

FIRST RESULTS

After installing and configuring the mentioned applications, jobs for J/ψ and $t\bar{t}$ were locally submitted to the cluster. In order to do that, Condor 6.6.9 was the job scheduler used.

The execution of J/ψ jobs was part of Dilson Damião's M.Sc. thesis, and it was made in two different topologies of diffractive events: the Diffractive Pomeron Exchange (DPE) and Single Diffractive (SD). For each topology, we ran jobs for each of the four distinct channels: *Signal*, *Lightmeson*, $b\bar{b}$ and $c\bar{c}$. Table 1 shows the average number of events generated and simulated per hour, for every channel in the two topologies.

	Signal	Lightmeson	$b\bar{b}$	$c\bar{c}$
DPE	65666	70376	62500	4687
SD	86957	95802	62868	3030

Table 1 – Average number of events/hour generated (by DPEMC) and simulated (by FAMOS) for all channels in the DPE and SD topologies.

Antônio Pereira has produced 1.000.000 FAMOS events for diffractive $t\bar{t}$ using just 25% of the cluster, in 48 hours. Those results will also be part of his M.Sc thesis.

SPEEDUP AND EFFICIENCY MEASURES

Some *speedup* and *efficiency* values were obtained for the “signal” channel jobs, in order to give us an first slight idea of how our cluster is operating. First, we launched groups of 100, 200, 400, 600 and 800 jobs in the cluster and calculated speedup for each group. *Speedup* is:

Speedup = time of sequential execution/time of parallel execution

For example, in the 100-jobs group case, we took the sequential execution time of 100 jobs and divided by the parallel execution time in the cluster. Based on speedups, then we obtained the cluster’s efficiencies for each jobs group. *Efficiency* is calculated by:

Efficiency = speedup/number of CPU’s

We can see the efficiency graphic for all jobs’ groups below (figure 3)

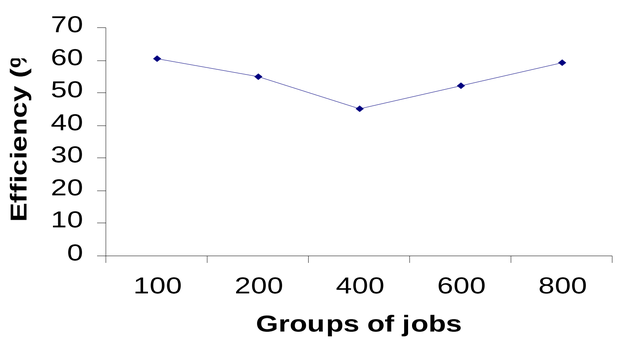

Figure 3 – Efficiences’ measures for different groups of jobs

It must be noticed that those measures were taken in an “hyper-threaded” environment – that is, the Production Cluster has 168 physical CPUs. As each one of them can emulate two virtual CPUs, thanks to “hyperthreading”, there are 336 virtual CPUs. However, we must keep in mind that a virtual CPU is pretty slower than a physical one – a job was scheduled to a virtual CPU.

We are currently working on jobs from other kinds of channels ($c\bar{c}$, lightmeson, $b\bar{b}$)