Job submission & monitoring on the PHENIX Grid*

A. Shevel[1] [†], B. Jacak[2], R. Lacey[1] and M. Reuter[2], D. Morrison[3], I. Sourikova[3], A. Withers[3] and T. Thomas[4]
[1]Dept. Chemistry, State University of New York at Stony Brook, NY 11794, USA
[2]Dept. Physics, State University of New York at Stony Brook, NY 11794, USA
[3]Dept. Physics, Brookhaven National Laboratory, NY 11973, USA
[4]Dept. Physics, State University of New Mexico, NM 87131, USA

Abstract

The PHENIX collaboration currently records ∼ 1/4 PB of raw data per year for each experimental run. Efficient and timely analysis of this data benefits from a framework for distributed analysis via the PHENIX Grid. Initial experience with this newly implemented grid infrastructure is presented with particular emphasis on job monitoring and job submission.

THE PHENIX GRID IN BRIEF

The PHENIX Grid is currently comprised of the RHIC Computing Facility (RCF), located ant Brookhaven National Laboratory (BNL) and five moderately sized clusters located in five different geographical locations across the United States and Europe. A schematic layout of the grid is given in Fig. 1. The clusters have different computing power, independent administration, different job schedulers, different Internet connectivity, etc, which makes for many interesting challenges. In particular no multi-cluster administrative privileges are currently available to the grid administrator/s.

In its current configuration, the PHENIX Grid provides invaluable CPU cycles for data analysis and simulations important to physics interpretation and/or detector efficiency studies. In what follows, we give a brief description of our early experience with this Grid.

EARLY EXPERIENCE WITHE THE PHENIX GRID

Infrastructure of the PHENIX Grid

The PHENIX Grid relies on existing Grid/Globus infrastructure. Specifically, for each cluster a dedicated or **Globus GateWay** (**GGW**) computer is assigned and all required components of the Globus server software (ie. GT-2.4 from http://www.globus.org) are deployed. That is,

- the GGW is a local machine in the cluster and is connected to the Internet.

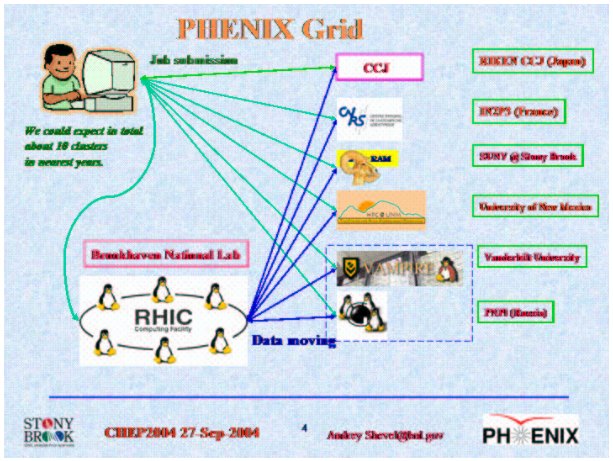


Figure 1: General scheme for PHENIX Grid computing resources

- All required Globus components are installed and running.
- The data and home directory of each user are accessible on the GGW.
- A batch scheduler software for the cluster is available on the GGW (i.e. it is possible to submit jobs from the GGW).

The procedure for achieving this implementation is relatively straightforward and works rather well. On the other hand, the flexible nature of the current management scheme for the clusters does not guarantee that all Grid enabled software components (see for example http://www.globus.org, http://www.ppdg.net, http://www.griphyn.org and http://www.edg.org) are uniformly available on on all clusters.

A primary focus of the PHENIX Grid is is to provide each user and administrator with easy access to:

- Information about data location. This is accomplished via a file catalog.
- Seamless job submission across clusters.
- Seamless job monitoring across clusters.
- Grid maintenance with minimal down-time.

*

† shevel@mail.chem.sunysb.edu

Job submission and monitoring on the PHENIX Grid

Multi-cluster/multi-site job processing requires a reliable set of job monitoring and control functions. For example, it is often the case that a user wants to;

- test the ability of a remote cluster to run a job over the Globus infrastructure.
- Query which of his jobs are currently running and where they are running (on which site or cluster);
- Query whether or not there is output from his job and find its location (directory file, cluster).
- Query where and how results will be collected (on which cluster, by which tool, script, etc.);
- Kill all or a specific selection of his submitted jobs on all clusters.

In addition to the points raised above, it is easy to envisage that rather detailed information about each job (in the multi-cluster environment) could be stored in a database for later access via a web-based interface. A similar (or the same) interface could provide primary control functions to submit, kill, and resubmit jobs. Such control functions would also provide fine granularity selections for each job such as job name, output length, execution time, input file names, etc.

Several administrative and technical considerations are important to the realization of seamless job submission and job monitoring on the PHENIX Grid. We list a few of these below

Administrative issues In addition to the requirement that each user has to be given the privilege to use clusters across sites, special consideration had to be given to various issues related to the fact that;

- each cluster is independently administered;
- all clusters are not singularly devoted to PHENIX activity;
- each cluster does not necessarily run the the same version of the Linux operating system;
- each cluster has a different local batch systems;

Given this reality, we found that it was particularly useful to implement a common cluster status board (see http://ram3.chem.sunysb.edu/phenix-grid) which allowed a user to interrogate several parameters for each cluster.

Other issues included;

- Reliable porting and testing of PHENIX software on each cluster. Typically the porting step was achieved via the Packman packaging scheme or by direct mirroring over AFS. After porting, each software of interest was tested to ensure that each cluster gave the same result.

A number of interesting challenges were encountered.

- *Problems associated with hardware (CPU architecture).* Porting is obviously straightforward between clusters having the same hardware and can be achieved by simply moving binary programs to new clusters. If the hardware on a new cluster is different then the porting process is much more involved.
- *Operating system.*

  This issue is similar to the one above. If one has exactly the same operating system on two clusters, one can copy most of ones scripts and configuration files and be pretty sure that most of them will work as intended. However if one cluster is running under different type of operating system the port of the software will require more effort.
- *Versions of system software in operating system.*

  It is also important to know the version of libraries of type 'glibc', the version of the kernel, libtool, and related utilities. Serious difference in different versions of these components serve to make the port of the software more difficult.
- Available resources.

  Issues concerning the availability of resources on each cluster is of course important. For example, in many clusters a user might not have access to enough disk space to accommodate the files generated by his job.
- *Batch scheduler.*

  Usually the user has no choice and have to use on new remote cluster standard for remote cluster batch scheduler. From the experience we know it might be: PBS, LSF, BQS, PBSPro, Condor, Sun Grid Engine, or something else. That means we have no possibility to copy user scripts for job submission as they are in the case when the user scripts use specific command names or/and parameters for batch job scheduler. From the real experience we know that batch scheduler may have some peculiar configuration. For example, at the university U the scheduler ProPBS has default queue the name 'default'. The permitted number of running jobs per the user is limited to 10. That means if user will submit 15 jobs first 10 jobs will be really submitted and will run but other 5 jobs will wait till the moment when at least one running user jobs will finish. It will happen in spite of the number of free machines in the cluster and the activity of other users in the cluster.

Job monitoring Two types of monitoring have been investigated for the PHENIX Grid; service monitoring and job/task monitoring. A wide range of service monitoring tools are readily available from the Internet. A short list which proved to be quite useful follows;

- Monalisa
- Ganglia
- Zabbix
- Big Brother

These tools provide seamless access to a variety of services such as:

- network status;
- computer status (CPU load, memory usage, swap usage, etc.);
- web service status;
- other types of service,

which proved to be immensely useful for monitoring the current status or past history of the activities of each cluster ie. CPU usage, memory usage, IO channel usage, etc.

In contrast to service monitoring tools, a typical user also has a need for actual job monitoring. For example, when a job is submitted to the PHENIX Grid it can run on any of the six clusters which comprise the Grid. A user needs to know his;

- job ID (probably simple unique number which does not depend of hostname, site name, etc);
- job name;
- job submission time;
- host name where job was submitted;
- job start execution time;
- other suitable parameters.
- etc, etc

Job monitoring tools are not pervasive. This is especially true for monitoring tools which can provide detailed information abut a users job. One monitoring tool that we have found to be useful is the Batch Object Submission System **BOSS** (see http://www.bo.infn.it/cms/computing/BOSS/).

BOSS consists of a binary interpreter which performs Command Line Interface (CLI) commands, a database which stores job status information, and a daemon, which is launched for each user job, that sends job information data to the database. A web interface (BOss Database Explorer or **BODE** (see http://filine.home.cern.ch/filine/bode/bode-2.1.1.tgz) provides easy access to the database. An overview of the job flow with BOSS is illustrated in Fig. 2. More than often, it is desirable to keep the history for each job for an extended period of time. We found this

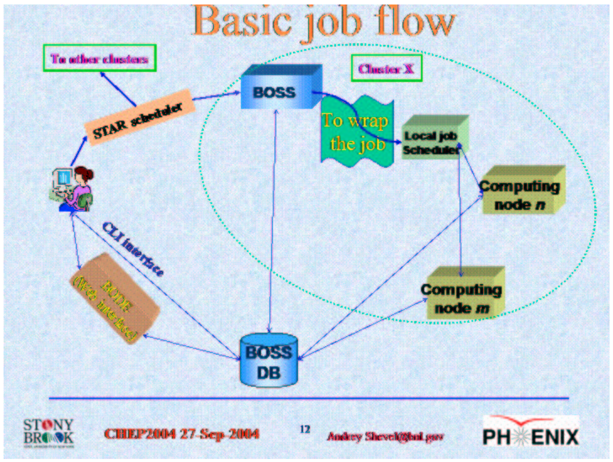
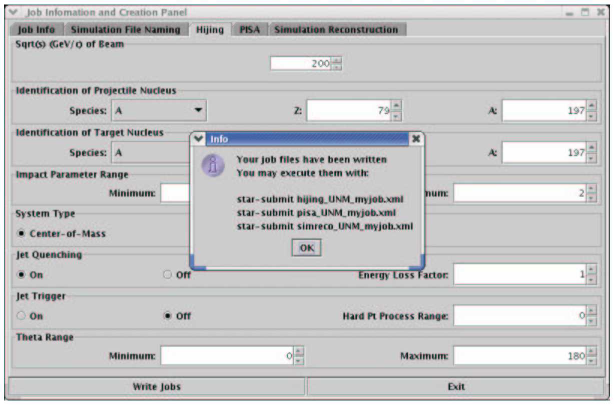


Figure 2: Basic job flow with use of BOSS and STAR scheduler.



Sqrt(s) (GeV/c) of Beam: 200

Identification of Projectile Nucleus
Species: A   Z: 79   A: 197

Identification of Target Nucleus
Species: A   A: 197

Impact Parameter Range
Minimum: Maximum: 2

System Type
[✓ Center-of-Mass]

Jet Quenching
[✓ On]   Off   Energy Loss Factor: 1

Jet Trigger
On   [✓ Off]   Hard Pt Process Range: 0

Theta Range
Minimum: 0   Maximum: 180

```
Your job files have been written
You may execute them with:

star-submit hijing_UNM_myjob.xml
star-submit pisa_UNM_myjob.xml
star-submit simreco_UNM_myjob.xml
```

Figure 3: The panel to generate the production jobs for simulation. The user can choose a range of physics parameters for the simulation.

to be particularly important for jobs executed on multiple clusters. For this, it was necessary to implement additional databases so that a user can store information about a current job for up to a year or more.

As stated earlier, it is important to have the possibility to submit and control jobs on multiple clusters. In order to do this, a tool kit GSUNY (see ftp://ram3.chem.sunysb.edu/pub/suny-gt-2/gsuny.tar.gz) was developed. Thus if a user wants to cancel all all jobs which are running on different clusters he would use the command **G_KillAllRun**. Currently, all commands in the GSUNY toolkit interacts with the **BOSS** database.

Job submission and control is possible from the command line or via a web interface. Fig. 3 shows a sample panel which is used for job submission. In this panel the user is able to enter several physics parameters for concrete simulation and after that start the simulation production (many jobs through STAR scheduler).

Despite the success of the PHENIX Grid, a number of challenges still remain. A brief list of some of the more important ones follow.

- Administration service (maintenance).
- More sophisticated job control and accounting across clusters.
- More complete technology for automatic deployment of the PHENIX software.
- More checking tools for regular testing (once an hour?) to be sure that all components of multi cluster multi site PHENIX Grid is running properly.
- A portal to integrate all PHENIX Grid tools in one user window.

SUMMARY

Initial investigations for implementation of a PHENIX Grid across a diverse range of institutions have proved to be quite successful. This success is reflected in the ability of many users to simultaneously marshal computational resources from several participating institutions for data analysis.

It is anticipated that that our current strategy of using existing subsystems as bricks to build a robust PHENIX distributed computing environment will continue.