DATA MANAGEMENT FOR PHYSICS ANALYSIS IN PHENIX (BNL, RHIC)

Barbara Jacak, Roy Lacey, Andrey Shevel, and Qiu Zhiping
SUNY at Stony Brook, NY, 11794, USA
Dave Morrison and Irina Sourikova
BNL, NY, 11973, USA

Every year the PHENIX collaboration deals with increasing volume of data (now about 1/4 PB/year). Apparently the more data the more questions how to process all the data in most efficient way. In recent past many developments in HEP computing were dedicated to the production environment. Now we need more tools to help to obtain physics results from the analysis of distributed simulated and experimental data. Developments in Grid architectures gave many examples how distributed computing facilities can be organized to meet physics analysis needs. We feel that our main task in this area is to try to use already developed systems or system components in PHENIX environment.
We are concentrating here on the followed problems: file/replica catalog which keep names of our files, data moving over WAN, job submission in multicluster environment.
PHENIX is a running experiment and this fact narrowed our ability to test new software on the collaboration computer facilities. We are experimenting with system prototypes at State University of New York at Stony Brook (SUNYSB) where we run midrange computing cluster for physics analysis [21]. The talk is dedicated to discuss some experience with Grid software and achieved results.

1. Introduction

PHENIX [1] is a large, widely spread collaboration where many organizations from many countries participate (12 nations, 57 institutions, about 400 collaborators, about 1/4 PB is planned to be produced in the current year). The collaboration is in third year of data taking. Distributed data (experimental and simulated), distributed computing facilities and distributed users is our reality.

Before discussing technical decisions we have to emphasize a range of common things and features in Grid computing for HEP which are important for further discussion.

First of all we paid attention that many systems for physics analysis are already in developing or prototyping stage [3, 4, 5, 6, 8, 15]. We recognized that using the Globus tools [17] is now standard de facto for distributed systems in HEP. The last thing was emphasized many times on CHEP2003 [22].

In more detail we could see the following common components (on all projects):

- using Globus Security Infrastructure (GSI);
- using the replica/file catalog for files distributed around the globe, however different projects use different cataloging engines: Globus Replica Catalog [10], other types of file catalogs;
- using the GridFTP over WAN in conjunction with other data moving protocols [23, 24];
- job submission with Globus tools;
- using the concept of virtual organization.

That is not surprising because every collaboration (PHENIX is not exclusion) needs several common tools:

- data moving over WAN;
- job submission from distributed users to distributed computing clusters;
- monitoring tools.

On other hand the character of using Globus middleware is different in different projects. If someone tries to see in deeper details - a lot of differences between collaborations could be discovered. Those differences are rooted in the style of physics measurements which in turn depends on the details of the physics experiment, style of collaboration work, existing computing resources, prehistory of computing infrastructure and many other details. In other words there is no evidence that concrete large and sophisticated system built on top of Grid middleware might be used in a different collaboration without reasonable adaptation or redesigning.

That means we have to discuss briefly PHENIX computing environment for physics analysis.

2. PHENIX computing environment for physics analysis

PHENIX has several computing facilities for physics analysis:

- RHIC Computing Facility - RCF [26] - main computing facility for PHENIX;
- CC-J [27];
- midrange cluster at SUNYSB [21];
- also there is a range of other computing facilities which is used for PHENIX physics analysis at several member institutes of PHENIX.

It is assumed PHENIX will have more computing facilities in future.

By taking a look at PHENIX member list it is clear that data distribution inside collaboration is not trivial. Even limited task to know the files are (in which site and location) is not possible without file/replica catalog (or file catalog).

3. File cataloging

In general file catalog is required to keep locations of all data files in the collaboration. There was a range of various decisions [2, 15] concerning file catalog.

Architecturally we tested two level *file cataloging engine*: distributed PHENIX file catalog [13, 14] is located at BNL and at remote universities and local SUNYSB file catalog based on MAGDA [15].

All instances of the catalog are interconnected by special replication process.

Technical description for central PHENIX file catalog was available in a different presentation on CHEP-2003 [25].

At the same time it was recognized that remote universities need some cataloging facilities for internal use.

4. Local cataloging and data moving at SUNYSB

An adapted version of MAGDA [15] was implemented and used at SUNYSB as local (at SUNYSB) cataloging facility [16]. With time, it became clear that it is suitable and important to keep the information about many files (physics data and other type of files: scripts, papers, etc.). Part of this information (information about files with physics data) is propagated to the central PHENIX file catalog.

There are several tools on top of adapted MAGDA at SUNYSB which are most interesting for end users. First of all there are web pages [16] with detailed information where and which files are available. Another tool is the set of scripts to link locally available data files to the local directory.

As described before, SUNYSB MAGDA catalog has replicated subset of central PHENIX file catalog. Periodically cronjob starts the scripts to transfer the information about part of the files from central PHENIX file catalog to MAGDA catalog and back, part of information from MAGDA catalog is to be copied to central PHENIX file catalog.

In this way it is possible to keep detailed information about files which are interesting for SUNYSB.

4.1. To link local data files

In this context we mean that commands below create the soft link for locally available files. Also information about linked files can be seen at the web pages [16] if the panel *Used Files* will be clicked.

- `ma_LinkLocalFile` *lfn* - to link locally available file *lfn* otherwise special completion code will be returned;
- `ma_LinkFileList` *list* - to link locally available files from the *list*;
- `ma_LinkFileSubstr` *substring* - to link all files which names are containing the *substring*;
- `ma_ShowLinkedFiles` - to display all files linked by current user;
- `ma_ReleaseFile` *lfn* - to release the file *lfn*;
- `ma_ReleaseHereFiles` - to release all files in current directory;
- `ma_ReleaseAllFiles` - to release all files earlier linked by current user.

When data file names are released the following steps are performed for every file:

- the appropriate soft link is deleted;
- the appropriate record is deleted from the MAGDA database; that means this record will not appear anymore in output of the command `ma_ShowLinkedFiles` and in statistics delivered on the web pages.

The information to the catalog MAGDA is coming from special spider scripts which are running on required *sites* (in our case there are 3 sites where spiders are running). On most sites a spider is started once a day or even once a week if the information is not changed often.

4.2. Data moving

File moving over WAN is done at SUNYSB through use of adapted MAGDA and through an alternative way - by a script. With MAGDA user could go to the web site [16] and describe required *site, host, location* (actually the exact directory path) and *collection* (collection of files). After that it is possible to describe the *task* for data transfer (with web pages on [16]). Real data moving is possible after activating the *task* by using the web pages [16]. At night cronjob will perform all the activated *tasks*.

For the user convenience the script **gcopy** was developed to copy the files over WAN. The script uses cluster descriptions which is discussed in chapter *Job submission* of this paper. The usage of the script:

gcopy *FromSite*:*FromLocation ToSite*:[*ToLocation*] \
[*substring*]

The parameters *FromSite* and *ToSite* are names of Globus gateways. *FromLocation* and *ToLocation* are exact directory paths. The parameter *substring* with wild-cards may be used to select file names in directory *FromLocation* to be copied.

Technically the file transfer is performed with GridFTP protocol (command **globus-url-copy**). The feature *third party transfer* is used the feature because the host where the file transfers are started is not the part of any computing cluster. Default number of threads for data transfer is 5. Due to a range of network routers (6 or so) between SUNYSB and BNL and due to other causes we see a significant difference in network connectivity speed during a day (a factor of 2). The best throughput we saw was about 7 MBytes/sec. This maximum throughput could be reached with different number of transfer threads and different size of network window at different time.

Taking into account those facts we conclude that it is difficult to predict what time the data transfer between BNL and SUNYSB will take. This is true for relatively large portion of data (0.2 TB and more).

Finally, it is much better to be sure that your data are available locally on the computing cluster where you plan to submit the jobs before job submission.

5. Job submission

In a distributed environment, it is often effective to use several computing clusters in different sites to get enough computing power or to load available clusters more evenly.

We have to emphasize that nobody in the collaboration needs computing power as it is. Physicists have a need to use *qualified* computing power. That means such a computing cluster where all PHENIX software is already installed and ready to be used. In further discussion we will assume the following:

- all required application software has been installed;
- required physics data are already locally available or if you plan to do a simulation you have enough disk space for the output simulated data;
- all Globus tools have been deployed;
- users have already the certificates to use Globus Secure Infrastructure (GSI) as well.

As already mentioned, the use of Globus infrastructure for job submission is common place now. At the same time till last autumn (2002) we had some difficulties with Globus toolkit (GT) (especially with data transfer). It was decided to create a light weight testbed with minimum functionality of GT, with minimum efforts, and with minimum time for implementation which could be tested in real environment where conditions are close to production reality. To do that a simple set of scripts was developed.

Our script set (set of wrappers on top of GT 2.2.3) for job submission and job output retrieval is deployed at client side. About 30 scripts were developed with total number of lines about 2000. Several of them are most significant for users.

- **GPARAM** - script to describe the configuration: number of computing clusters, names of Globus gateways, other information; in addition the script $HOME/.gsunyrc (same meaning as **GPARAM**) is used to keep local values for current account;
- **gproxw** - to create Globus proxy for a week;
- **gping** - to test availability of all clusters described in **GPARAM**;
- **gping**-*M* to test availability of a desired cluster (here *M* is suffix to denote a cluster: **s** - for cluster at SUNYSB [21], **p** - for PHENIX at BNL [26], **unm** for cluster at University of New Mexico [20]);
- **grun**-*M* to perform one command (script) on a desired cluster (see remark to **gping**-*M*);
- **gsub**-*M job-script* - to submit the *job-script* to a desired computing cluster (see remark to **gping**-*M*);
- **gjobs**-*M* [*parameter*] - to get the output of command **qstat** on a desired cluster adn *parameter* is parameter to **qstat** (see remark to **gping**-*M*);
- **gsub** *job-script* - to submit the *job-script* to less loaded computing cluster;
- **gsub-data** *job-script file* - to submit the *job-script* to the cluster where file *file* is located; if the file *file* has replica on all clusters - to submit to less loaded cluster. First of all the file location will be tested through local (on site) file catalog.
- **gget** *job-id* - to get output from accomplished job *job-id*, if parameter is missing then last submitted job will by taken into account.
- **gstat** *job-id* - to get status of the job *job-id*, if parameter is missing then last submitted job will be taken into account.

If submitted job generates some files they will be left on the cluster where the job was performed. The files could be copied to a different location by data moving facilities described in previous section of the paper.

The meaning of *less loaded cluster* is important.

5.1. Less loaded cluster

We need to know on which cluster the expected execution time for the job is minimum. Unfortunately this task in an unstable environment has no simple and precise solution. The estimate becomes worse if the job runs long time (many hours for instance). All estimates might be done only on some level of probability.

That was the reason why we took for a prototype a simple algorithm to determine *less loaded cluster*. In principle that choice reflects our hope that situation with job queues will

not be changed fast. Values from queuing system (the answer from the command **qstat**) are used in algorithm. The estimate algorithm uses also a priori information about the average relative power of a node in the cluster. In our configuration we use two clusters: average computing power for the node at SUNY was determined as 1, average computing power at BNL was determined as 2 (the machine at BNL was twice faster). Another parameter that is used the maximum number of jobs which may be in run stage at SUNY and at BNL. All those parameters are assigned in the cluster description script **PARAM**.

Just before starting the job the scripts **gsub, gsub-data** will gather the information about real status of queues on every cluster described in the configuration (it is done with Globus command *globus-job-run* which gets the answer from **qstat**). After that the following value for each cluster is calculated:

L = [(number of jobs in run stage) + (number of jobs in wait stage) - (maximum jobs in run stage)] / (relative computing power)

The cluster with a minimum value of L is considered as *less loaded cluster*. Of course it is only an estimate of the reality. Some peculiarities of a dispatching policies in local job manager (LSF, PBS, other) could make the above estimate wrong. However in most simple cases it gives the right direction.

More sophisticated algorithms might be discussed separately.

The current client part is really simple.

5.2. How to use described scripts on client side

We assume the client side (usually desktop or laptop) has a stable IP address and can be seen in the Internet. This feature is mandatory to use all client Globus stuff (we used GT-2.2.3 and GT-2.2.4 on client side).

- Deploy the Globus toolkit.
- Copy all scripts from [18]. After that please become root and do

```
tar zxvf gsuny.tar.gz
cd GSUNY
./setup
```

  All the scripts will be at the directory /usr/local/GSUNY/. To make them available please add this directory to the $PATH environment variable.

Now the client is ready to use almost all the commands (excluding **gsub-data** which requires an additional package [19]).

First command has to be **gproxw**, it creates the Globus proxy for a week. You could start **gping** after that. The command **gping** will show existing configuration (number of clusters, Globus gateways, other parameters).

Now you could submit the jobs to described clusters. If you submit many jobs (several tens or hundred) it may happen that they will run on different clusters. The job submission scripts use special logs to remember where the jobs were submitted and which Globus commands in multicluster environment were performed.

5.3. User log files in multicluster environment

In order to keep the trace of user commands in Globus context and job submission several log files have been created and used:

- `$USER/.globus/CLUSTERS/commands` - file contains list of performed Globus commands in format *date/time command parameters* ;
- `$USER/.globus/CLUSTERS/jobs` - file contains job ID (with name of globus gateway) in format *date/time jobID*
- `$USER/.globus/CLUSTERS/DataTransLogs` - directory contains several files to keep trace of data moving over WAN.

Almost all of mentioned scripts add some records to the above log files.

The logs are very valuable for many reasons: debugging, statistics, etc. They are also important because the set of clusters which is used by a user may be different.

5.4. Changing the set of involved clusters

The technical adding of a new computing cluster consists of several simple steps for every/only client computer:

- to change the parameters you can edit the script /usr/local/GSUNY/GPARAM (for system wide parameters) or to edit the script $USER/.gsunyrc (for current account);
- to talk to cluster authority to add your account on new cluster to the file /etc/grid-security/grid-mapfile on new cluster Globus gateway.
- to prepare three scripts and put them into the directory /usr/local/GSUNY/
  - **{** script to ping the new cluster (please see **gping-p** as an example);
  - **{** script to submit the job (please see **gsub-p** as an example);
  - **{** script to get the new cluster load level (please see **gchk-p** as an example).

Table I Measurement results

Command	Execution time	Remarks
gping	6 secs	
gsub	42 secs	
gsub-p	26 secs	
gsub-s	5 secs	
qsub-unm	11 secs	
gsub-data	17 secs	including looking at the catalog

After that you can use the cluster under your own account from the desktop where you changed the scripts.

If the cluster has to be from the configuration, you have to edit the script /usr/local/GSUNY/GPARAM accordingly: it is possible to delete the description of the cluster and change the number of clusters.

If you have to change the cluster (to use new cluster instead old one), you have to edit script /usr/local/GSUNY/GPARAM (or $USER/.gsunyrc) as well.

6. Results and discussion

An execution time for different scripts is shown in table I. Here we have to emphasize that this time is required only to submit the job to the standard job queue. We use LSF at BNL and Open PBS at SUNYSB as local job managers.

By looking at the table it is easy to realize that job submission takes time. It is not surprising because we used Globus command *globus-job-run* to get current load of the cluster and we did not use Globus Index Information Service (GIIS) [11]. Also we use the parameter *-stage* for the command *globus-job-submit* to copy a job script from local desktop to remote cluster (it takes time as well). Somebody may think that there is no reason to submit the job with expected execution time 1 minute because the overheads for job submission might be more than execution time. On the other hand if you do not know exactly the situation on clusters your short 1 minute job may stay in input queue many hours due to high load or some problem on the computing cluster.

To decrease the delays we plan to use Monitoring and Discovery Service (MDS) [11] in nearest time.

7. CONCLUSION

The first working prototype includes central PHENIX (BNL), SUNYSB, Vanderbilt University (only replication subsystem for PHENIX file catalog was deployed), University of New Mexico [20] (only simple job submission scripts were deployed). All components are working and results look promising:

- The distributed information about location of significant collaboration physics data is delivered in uniform and consistent manner;
- Job submission from remote universities could be directed to less loaded cluster where required data are. It helps to use computing resources more effectively in two different scenarios.
  - In first scenario it is needed to collect all available distributed computing power to do an analysis. In this situation you need to distribute the data around computing clusters before starting the job chain. Described tools will help to load all available clusters evenly.
  - In second scenario there are already distributed over several clusters data (experimental and simulated). In this case the using of described tools will help to minimize data moving over WAN.

Apparently it was done the first step in implementing the flexible distributed computing infrastructure. Some additional work is required on robust interfaces in between cataloging engine, job submission tools, job accounting tools, data moving, and trace information about the jobs.

Finally, during the implementation we learned several lessons:

- Deployment of the Grid infrastructure in collaboration scale may not be a business for one person.
- Grid architecture has to be supported at central computing facilities.
- Better understanding of our needs in Grid comes with real deployment of the components.

Acknowledgments

Authors have to mention the people who gave us valuable information and spent discussion time: Rich Baker, Predrag Buncic, Gabriele Carcassi, Wensheng Deng, Jerome Lauret, Pablo Saiz, Timothy L. Thomas, Torre Wenaus, Dantong Yu. Special thanks for our colleagues Nuggehalli N. Ajitanand, Michael Issah, Wolf Gerrit Holzmann who asked many questions and helped to formulate things more clearly.

The work was done with support of the grant NSF-01-149 (award number 0219210).

References

[1] PHENIX home page
http://www.phenix.bnl.gov/

[2] Distributed File and Tape Management System (FATMEN)
http://wwwinfo.cern.ch/asdoc/fatmen_html3/fatmain.html

[3] SAM GRID INFORMATION and MONITORING SYSTEM
http://samadams.fnal.gov:8080/prototype/

[4] Alice Environment - AliEn
http://alien.cern.ch/

[5] The Chimera Virtual Data System
http://www-unix.griphyn.org/chimera/

[6] GRAPPA - Grid Access Portal for Physics Applications
http://iuatlas.physics.indiana.edu/grappa/

[7] iVDGL - International Virtual Data Grid Laboratory
http://www.ivdgl.org/index.php

[8] NORDUGRID
http://www.nordugrid.org/

[9] Grid Data Mirroring Package (GDMP)
http://project-gdmp.web.cern.ch/project-gdmp/

[10] Globus Replica Catalog
http://www-fp.globus.org/datagrid/deliverables/replicaGettingStarted.pdf

[11] Monitoring and Discovery Service
http://www.globus.org/mds/mdstechnologybrief_draft4.pdf

[12] VAMPIRE - Grid Computing
http://www.vampire.vanderbilt.edu/grid_detailed.php

[13] API to central PHENIX file catalog
http://www.phenix.bnl.gov/phenix/WWW/offline/tutorials/frogDoc.html

[14] Web interface to central PHENIX file catalog
http://replicator.phenix.bnl.gov/replicator/fileCatalog.html

[15] MAnager for Grid distributed DAta - MAGDA
http://www.atlasgrid.bnl.gov/magda/info

[16] MAGDA, adapted for SUNYSB
http://ram3.chem.sunysb.edu/magdaf

[17] The Globus project
www.globus.org

[18] Client scripts
ftp://ram3.chem.sunysb.edu/pub/suny-gt-1/gsuny.tar.gz

[19] Magda-suny (SUNY distribution)
ftp://ram3.chem.sunysb.edu/pub/suny-gt-1/magda-client.tar.gz

[20] Computing Facility at University of New Mexico
http://www.hpc.unm.edu/

[21] SUNYSB computing cluster
http://nucwww.chem.sunysb.edu/ramdata/

[22] 2003 Conference for Computing in High Energy and Nuclear Physics
http://www-conf.slac.stanford.edu/chep03/

[23] bbftp
http://doc.in2p3.fr/bbftp/

[24] bbcp
http://www.slac.stanford.edu/~abh/bbcp/

[25] Relational Data Bases for data management in PHENIX
http://www-conf.slac.stanford.edu/chep03/register/report/abstract.asp?aid=461

[26] RHIC Computing Facility
http://www.rhic.bnl.gov/RCF/

[27] CC-J System Information
http://ccjsun.riken.go.jp/ccj/

[28] World Grid demonstration
http://www.ivdgl.org/demo/worldgrid/

[29] Atlas Map Center
http://www.atlasgrid.bnl.gov/mapcenter/

[30] Brookhaven National LAboratory
http://www.bnl.gov