GRID2003 MONITORING, METRICS, AND GRID CATALOGING SYSTEM

M. Mambelli* (University of Chicago), B. Kim# (University of Florida),
I. Legrand (CERN)
I. Fisk, J. Weigand (Fermi National Laboratory)
L. Grundhoefer, R. Quick, J. Hicks (Indiana University)
M. Green (University at Buffalo)
R. Gardner, A. Zahn (University of Chicago)
C. Prescott, J. Rodriguez, P. Avery (University of Florida)

Abstract

Grid computing involves the close coordination of many different sites which offer distinct computational and storage resources to the Grid user community. The resources at each site need to be monitored continuously. Static and dynamic site information needs to be presented to the user community in a simple and efficient manner.

This paper will present both the design and implementation of the Grid3 monitoring infrastructure and the design details and the functionalities of a new application called the GridCat.

The Grid3 monitoring architecture follows a user-oriented design that specifies standard metrics and utilizes underlying monitoring tools to collect them into a diversified framework. Then existing tools can be integrated, their functionality extended and new tools developed. The primary tools used include the ACDC Job Monitoring system from University at Buffalo, Ganglia, a preliminary version of GridCat, Globus MDS, the University of Chicago Grid telemetry MDViewer, and US CMS MonALISA. Information of interest for a particular VO can be extracted from archives of collected data. For example, sufficient information necessary to associate resources provided and used for each VOs, and job statistics for each VO.

GridCat provides a service giving status information through a web interface with a database backend that collects and stores a site's configuration and dynamic information. It is simple and powerful enough to determine the site's readiness to accept grid applications. The status information displayed by the prototype GridCat was used extensively by the Grid2003 project as a coordination point for the iVDGL Grid Operations Center (iGOC).

INTRODUCTION

Many large science collaborations have computing requirements that exceed the capabilities of most computing facilities. The amount of data involved may range to the Petabyte scale, requiring the combined resources of several institutions to provide scientists with the necessary data and compute capability to process their data, Computations may require millions of CPU-days to complete which in many cases is made more challenging by the distributed nature of the organizations. Common, shared Grid infrastructures are being developed (ref grid projects LCG, EGEE, TeraGrid, PPDG, GriPhyN, and iVDGL [6]) in order to overcome many of the practical obstacles standing in the way of enablement of applications requiring high performance, distributed computing platforms.

The Grid2003 project was organized as a collaboration to build an application-driven Grid laboratory, called Grid3, to run production scale applications driven pincipally by two of the LHC experiments, ATLAS [1] and CMS [2], as well as experiments from astronomy and astrophysics, LIGO [4] and SDSS [3], and the Fermilab collider experiment, BteV [5]. In addition, computer science demonstrators and biology applications were deployed.

The architecture of the Grid3 environment is quite simple. A Site is defined as a set of resources accessible from outside using only the protocols provided by the Grid middleware (it may be internally connected in other ways). Each Site conforms to a set of accessibility conventions that allows usage of its resources according to public policies. A Virtual Organization (VO) is an organization providing resources and making active usage of the Grid. It is the administrative unit coordinating the admission of new sites or new users and therefore allowing a decentralized management of the Grid. Grid3 itself is a dynamic collection of Sites, and other operation related resources, like the Globus MDS indexes (GIIS) [18], the VOMS servers [12], VO servers for user management, and the server to archive monitoring information. Multiple VO services were located at the iVDGL Grid Operation Center (iGOC), co-located with the Abilene NOC in Indianapolis [21]. Globus MDS allows the dynamic discovery of all the resources in Grid3. VOMS delegates to one server per VO the registration of the users, allowing the Sites to manage authentication and authorization with a VO level granularity. More details about the project may be found in [12]. The remainder of this paper will focus on the Grid3 monitoring system, with emphasis on resource

*marco@hep.uchicago.edu
#bockjoo@phys.ufl.edu

discovery, availability, load measurement, profiling and debugging.

THE GRID3 MONITORING SYSTEM

In August 2003 Grid3 and its core services were designed by identifying principles derived by immediate application needs. The collected data was supposed to serve three main purposes:

- Document and measure the activity of the project; allow comparisons suitable for accounting and for comparing grid production with classical batch productions.
- Provide scientists using the grid tools to efficiently submit and track that execution of jobs, without requiring extensive expertise in the Grid technology.
- Provide to developers of the grid middleware and site administrators effective troubleshooting tools to discover problems, investigate the causes in order to improve system reliability and performance.

The monitoring system needed to be flexible and non-invasive in order to take advantage of software installed at different sites and thus reduce the burden on site administrators. Efforts were made to reduce the number of packages installed. Finally, the system needed to be in place on a time-scale of approximately three months from the project's start in order to be ready for experiment data challenges.

At the beginning, performance metrics were identified with an eye towards re-using and integrating with existing tools. Following principles set forth in [19], we utilized different levels of granularity and aggregation of the provided information, and where possible made choices that avoided single points of failure while increasing scalability. Different paths for information allowed for redundancy which made the system more reliable and easier to debug. We considered ease of integration and in some cases the anticipated level of support given by developers.

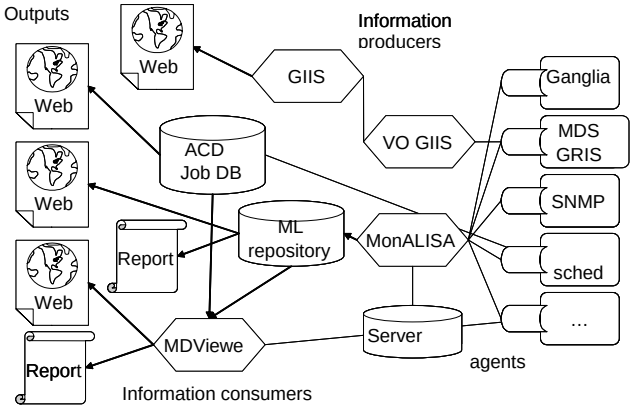

Figure 1: Architecture of the Grid3 monitoring system.

Figure 1 shows the resulting system architecture. The main components involved are: the Globus MDS, MonALISA, Ganglia, ACDC, GridCat and MDViewer. Contributions in these tools range from deployment to full development. For information providers installed on sites, we utilized the metapackaging tool Pacman.

Services requiring central collection servers (JINI, the top-level MDS GIIS, and the ML repository) were hosted at the iGOC, with backup servers for ML at CERN and CalTech.

The Globus Meta Directory System (MDS) is a distributed information service based on LDAP and deployed in Grid3 as a three level architecture with Site GRIS, VO servers and a central directory (GIISes) [ref. G3ID_System]. The base Globus MDS was extended by deploying the GLUE information providers for grid interoperability; new information providers (Grid3 IP) were developed to provide users with attributes describing installed application locations, available disk storage and limited monitoring measures for I/O activity.

The MonALISA [15] framework was used to collect and aggregate monitoring information. The system, based on a web service publish/subscribe model and a relational database backend, requires deployment of "station servers" at the sites, a central repository, and comes with an attractive client GUI which could be used by Grid3 users. Since MonALISA was designed to easily acquire information from disparate systems, we extended the system by adding additional information producers (MonALISA agents) for site performance and job monitoring. The central server acts as display and historical archive of summarized information.

Ganglia [14] is an efficient tool to collect usage and load information about the local cluster from sources like /proc filesystem, kvm (kernel memory interface) and kstat (kernel statistics). In Grid3 it is used to feed information into MonALISA and also its hierarchy with a central server [14] at iGOC. In both cases Ganglia is used to collect cluster information.

The Job Monitoring System at the Advanced Computational Data Center, in Buffalo NY (ACDC Job Mon)[16], provides near real-time snap shots of critical computational job metrics, which are stored in a database and utilized by dynamic web pages that it generates for the user. Jobs are segmented into several classes (running, queued, historical, etc.) and statistics for each class are created on-the-fly. The ACDC Job Mon uses MDS to discover the characteristics of the Sites, and then it sends appropriate Grid jobs to collect the desired information.

GridCat [17], the Grid3 Grid Cataloging System, provides summarized system and availability information about the whole Grid3. It has been developed within Grid3 and is the primary tool for the Grid Operation Center; it is the subject of the next section.

The Metrics Data Viewer (MDViewer) is a Web application developed in Grid3 to analyze the data collected and to produce plots out of them. It works with different data sources within the Gri3 monitoring system, provides some predefined plots and a library easy to extend to add new plots or data sources.

All these elements are integrated and together provide both real time snapshots or the Grid3 status and historical archives of its behavior and use.

Parallel to the software framework there is also an organizational structure covering coordination, development, deployment and troubleshooting activity. A working group is coordinating the monitoring system and planning the introduction of new features, while the activity on the single tools is coordinated with their main developers, often external to the collaboration. A trouble ticket system at iGOC allows the users to submit problems via Web or via email and have these ticket managed internally in the monitoring team. This is important, since iGOC together with the monitoring team provide to scientists and system administrators a central reference for the whole monitoring system, managed as a whole, and they don't have to worry about the single monitoring components.

GRIDCAT, GRID CATALOGING SYSTEM

Most of the Grid3 monitoring functionalities are derived from existing tools have been customized, extended and integrated . In contrast, GridCat is a new project designed and built from scratch. Although GITS[20] was some inspiration, it had to be extensively tailored to Grid3 and made efficient and simple enough to use for the iGOC. Functioning like a 'Control room' display, the catalog and status of the Grid is how the Grid itself is introduced, visually represented and known. The initial prototype, developed and extensively used during the Grid2003 project [13], is evolving now in a tool still with a simple front-end, but with added features and a more complete and modular back-end.

What is GridCat

GridCat is a Web application that presents high level status information on geographic maps about resources in a Grid. It can be characterized as:

- An information collector because it collects static and dynamic site information using remote Globus commands
- An information presenter, because it displays all the collected information in a dynamic Web page with the status map, where you can choose to view summaries or details
- A tool to check the grid availability, the available CPU slots and the disk space
- An archival information database, because it stores the newly updated database periodically

The information stored in the database can be analyzed and, if necessary, the results of the analysis of the database content can be presented in a separate Web page. All Web pages are generated by PHP scripts using HTML templates and a few java-scripts.

Components and Operational Flow

There are three GridCat components: configuration, front-end Web presentation, and back-end scripts. GridCat configuration is provided with *autoconf*. Beyond the initial configuration and installation, a process of manual site population must be carried out to fill in a list of network connected hosts. The front-end provides a view of the site status map and catalog. Green and red status dots represent "healthy" and "unhealthy" sites, respectively. The back-end is coded using unix shell scripts and PERL scripts; these scripts are mostly wrapper scripts for Globus commands with system queries of remote sites. The back-end scripts perform a various grid tests and query the sites about available CPU slots and disk space. The grid tests include an authentication, an interactive command execution over the grid, a batch job submission, a batch job status query, a batch job clearing, and a GSI file transfer test. It also checks static information, e.g., the check on the directory information for application binary programs.

Almost all back-end scripts are run in parallel to reduce the information query interval among sites. The load per site due to the back-end scripts is fairly small. The information retrieved by the back-end scripts is stored in a MySQL database. Finally, the collected information is retrieved and a status map is generated by the back-end, so that the front-end can query and display the status. The information that is not related with the status map is retrieved directly by the front-end while generating the dynamic page. Figure 2 shows an operational flow of GridCat.

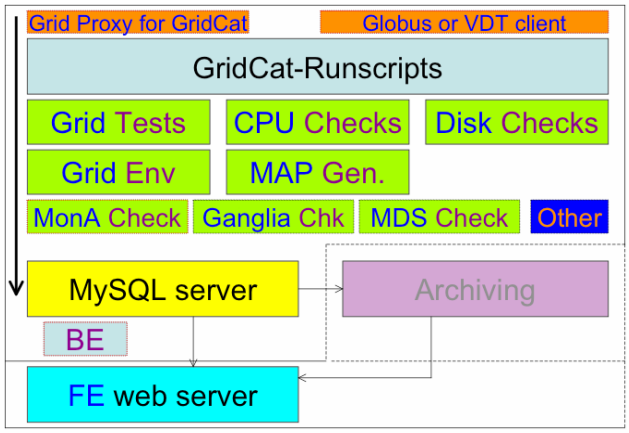

Figure 2: Operational Flow of GridCat

Features, Use-Cases, and Database Analysis

GridCat gathers and presents static and dynamic resource information in a simple Web presentation using a geographic map and catalog of information. Map views help any collaborator quickly identify resource location, availability and problems on grid. Catalog information is easy to consult. Database archiving can further be utilized in more complex failure analyses.

The most important information for application users is the possibility to know the number of available CPU slots for job submissions. This is especially important with the current grid computing environment, because most of job scheduling is done manually by the application users. Information could also be used by those who design schedulers as a quick reference.

The available disk space and the directory structure are additional information important to the application users,

so they determine whether there is enough space for the input and output data for the application they want to run. Because the Grid3 model includes a series of predefined spaces for data and applications, their path is left to the site administrator to decide. So, knowing these paths is important because the user has to know where to install the application and store the data.

GridCat is increasing its flexibility. The prototype system was quite static. The current version allows dynamic lists of sites and a tool to facilitate the editing has been developed and added. Under development is the possibility to remotely modify some attributes. For example, site administrators will be able to authenticate using their grid proxy and directly modify entries like email contact, URL and other site specific information not available through the monitoring interface.

The site status information should be maintained as up-to-date as possible for accurate and timely site status information query. Primarily, there are three types of scripts that need to be running to efficiently gather different types of dynamic information: Grid status tests, available CPU slots and site's disk and directory information. For 24 grid sites, the completion time ranges between 30 and 40 minutes.

The status map should be updated synchronously with the information gathered by the three scripts. The status map is rendered visually using a 2-dimensional template PNG file and adding colors and text labels. Since all the back-end scripts run as a single complex daemon, a daemon script is provided to start, stop, or check the status of the back-end scripts. In addition to the daemon script, the db analyzer daemon can be run simultaneously.

The historical information stored in the MySQL database during the operation of GridCat is a valuable source for analysis of complex failure modes and helps a grid operation center identify chronic problems. An example of such database analysis is illustrated in Figure 3. In Figure 3, the site availability history is presented using the database accumulated between Aug. 19 and Sep. 22, 2004. When the bar in the 'maptogram' is green the site was fully functional during that period of time, when red it was not.

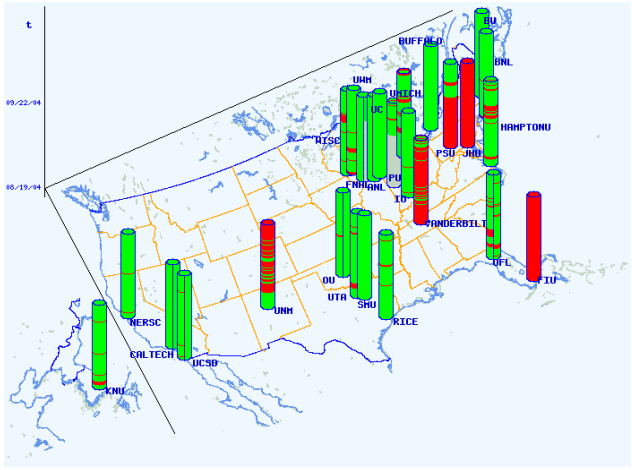

Figure 3: GridCat Site Status History Analysis

CONCLUSIONS

The Grid3 monitoring system successfully integrated pre-existing and internally developed heterogeneous tools, in a coherent and reliable infrastructure. It extended and tested thoroughly several of the tools. A set of user defined metrics are collected and are used to present the real time status of the grid, to interact with it, and to document its past activity. This data has been used to show how multiple VOs are able to take advantage of the resources shared in Grid3 and to analyze usage patterns and improve user behaviors.

GridCat has proved to be a very effective application for users, site administrators and especially the Grid Operation Center. Thanks to its power and simplicity the USATLAS Tier1 at BNL has decided to adopt it for monitoring its resources. For the same reason GridCat could be extended as a Web application for other uses.

Thanks to all the staff and personnel of the participating institutions of the Grid2003 project for their help. This research was supported in part by the National Science Foundation, Grant ITR-0122557 and ITR-0113343.

REFERENCES

[1] A Toroidal LHC ApparatuS, http://atlas.web.cern.ch/
[2] The Compact Muon Solenoid, http://cmsinfo.cern.ch/
[3] Sloan Digital Sky Survey, http://www.sdss.org/
[4] Laser Interferometer Gravitational Wave Observatory http://www.ligo.caltech.edu/
[5] BTeV experiment, http://www-btev.fnal.gov/
[6] International Virtual Data Grid Laboratory, http://www.ivdgl.org/
[7] Grid Physics Network, http://www.griphyn.org/
[8] Particle Physics Data Grid, http://www.ppdg.net/
[9] Globus, http://globus.org/
[10] Virtual Data Toolkit, http://www.cs.wisc.edu/vdt/
[11] Condor, http://www.cs.wisc.edu/condor/
[12] The Grid2003 Project "The Grid2003 Production Grid: Principles and Practice", *HPDC13*, 2004, Honol., HI, Grid2003, http://www.ivdgl.org/grid2003
[13] European Data Grid, http://eu-datagrid.web.cern.ch/
[14] F. Sacerdoti, M. Katz, M. Massie, D. Culler, "Wide Area Cluster Monitoring with Ganglia", *IEEE Cluster 2003 Conference*, 2003, Hong Kong http://gocmon.uits.iupui.edu/ganglia-webfrontend/
[15] I. Legrand, H. Newman, P. Galvez, R. Voicu, C. Cirstoiu "MonALISA: A Distributed Monitoring Service Architecture", *CHEP2003*, La Jola, CA http://gocmon.uits.iupui.edu:8080/index.html
[16] ACDC Job monitor, http://acdc.ccr.buffalo.edu/
[17] GridCat, http://www.ivdgl.org/grid2003/catalog
[18] MDS, http://www-unix.globus.org/toolkit/mds/
[19] B. Tierney, R. Aydt, et al. "A Grid Monitoring Service Architecture", *GGF*, 2001
[20] D.J. Baker, et al. "Building the e-Science Grid in the UK: Providing a Software Toolkit to Enable Operational Monitoring and Grid Integration", AHM, 2003, Nottingham
[21] Abilene/Internet2 http://abilene.internet2.edu/