

*GridMonitor: Integration of Massive Facility Fabric Monitoring with Meta Data Service in Grid Environment**

RHIC/USATLAS Computing Facility
Department of Physics
Brookhaven National Lab
Upton, NY 11973, USA

Abstract

Grid computing consists of the coordinated use of large sets of diverse, geographically distributed resources for high performance computation. To monitor these Grid computing resources and provide Grid information service becomes extremely important for efficiently using Grid Computing resource. The large numbers of computing entities with great diversities make the task extremely challenging. In this work, we describe a Grid Monitoring Architecture which captures the most important characterization from large facility monitoring. The GMA consists of four tiers: local monitoring, archiving, publishing, harnessing. This architecture was applied to large scale of linux farm and network infrastructure. It can also be used by other higher-level grid services, i.e. scheduling service, resource brokering.

1 Introduction

Grid computing consists of large sets of diverse, geographically distributed resources that are collected into a virtual computer for high performance computation. The success of Grid depends greatly on efficient utilization of the resource. PPDG is a primary user of Data Grid. PPDG is a collaboration of computer scientists in distributed computing and Grid technology, and physicists who work on major high-energy and nuclear experiments. These experiments include ATLAS, Star, CMS, D0, and Babar. PPDG. There are tremendous computing resources involved in PPDG physics experiments. For example, BNL computing facility consists of computing resource, disk storage system and taper storage system. The computing resource includes 700 dual processor PCs which come from six different vendors. The Linux farms at the BNL RHIC/USATLAS provide the 1.082 Tera-Flops computation power. The storage system provider 66 Tera-Bytes disk space and 1.2 Peta-Bytes robotic tape storage space. The diversity of these computing resource and their large number of users make the grid environment vulnerable to faults and excessive loads. This seriously affects the utilization of grid resources. Therefore, it is crucial to get knowledge about the status and performance of the computing resource to enhance the performance and avoids faults. The following example illustrates that an application relies on Grid information server:

*This work is supported by PPDG/ATLAS grants

A job scheduler needs information about available CPU resources in order to plan the efficient execution of tasks. A *Compute Farm* consists of a set of one or more CPUs available for scheduling via Grid protocols. If required by the exact nature of the interrelationship between the farm monitor and the job scheduler, the CPU at a given site may be broken down into multiple clusters that consist of homogeneous nodes, such that the local job manager can assume that any queued job can be run on any available node within the farm, Grid information service should provide the system status about each cluster, i.e. cluster configuration, associated storage system, and so on.

Many applications, fault detection, performance analysis, performance tune, prediction, and schedule needs information about the Grid environment. Good methods should be designed to monitor resource usage, get the performance information and detect the potential failures. Due to the complexity of the Grid, to design monitoring architecture for such a large scale of computing resource is not a trivial work. The targets to be monitored in grid resource include CPU usage, disk usage, and network performance among grid nodes. The ability to monitor and manage distributed computing components is critical for enabling high-performance distributed computing. Monitoring data is needed to determine the source of performance problems and to tune the system and application for better performance. Fault detecting and recovery mechanisms need monitoring data to determine if a server has a problem. A performance prediction service might use monitoring data as inputs for a prediction model, which would in turn be used by a scheduler to determine which resources to use. When more people use Grid, the more instrumentation requirements will be discovered, and the more facility information needs to be monitored. Many researches have been conducted on monitoring computer facility in a relatively small scale. There are many people working on the Grid monitoring systems. The proposed systems are Autopilot [2], Network Weather services [5], Netlogger [4], Grid Monitoring Architecture (GMA) [3] and Grid Information Service (MDS) [1].

Due to diversity of the computing resource and applications in Grid computing, existing monitoring architecture can not monitor all of the computing resource belonging to the grid. When the size of computing facility grows, existing monitoring strategy will significantly increase system overhead. The dynamic characteristics of grid resources allows the computing resource participate and withdraw from the resource pool constantly. Only a few existing monitoring system address this characteristics. In this work, we present a grid monitor system which is adapt to the Grid environment, It includes:

- Local monitoring: The local monitoring system monitor the facility which consists of computing, storage and network resources. The monitor information will be provided to different types of application with different requirements.
- Grid Monitoring: the grid monitoring uses MDS [1] to publish the selected monitoring information into Grid environment.

The proposed architecture can separate the facility monitoring from the grid environment. By using the MDS, it provides a well-designed interface between Grid and facility. It can provide monitoring information

for different grid applications as long they use protocols provided in [1]. When new hardwares are added in the local facility, the local monitoring structure can easily add the new software to monitoring the system. These change of hardware and monitoring tools can be hidden from the Grid computing environment.

The rest of work is organized as follows: Section 2 introduces the definitions of grid information service. Section 3 provide the architecture of GridMonitoring. Section 4 evaluates the performance the GridMonitor. Section 5 summaries recent work on Grid monitoring and Grid information service. Section 6 gives the conclusion and the future work.

2 Terminology

- **Sensors:** A sensor can measure the characteristics of a target system. It generates a time stamped performance statistics. A sensor typically execute one of UNIX utilities, such as top, ps, ping, iperf, ndd or read constantly from system files, such as /proc/* to extract sensor-specific measurements. Typical sensors are used to monitor CPU usage, memory usage and network weather. Some sensors can monitor and capture system abnormal status. We call the type of measurement provided by a sensor subject.
- **Information Provider (IP):** Information provider provides detailed, dynamic statistics about instrumentation. Information provider either invokes and stops a set of sensors to do active probing or interacts with running sensors to obtain the current status of resource. An information provider can also query database to get historical information. In our grid environment, the information providers (GRIS) need to talk two protocols to make information about resource available to users: one is to specify how to access information from this information provide (GRIP), the other one is to register the information provider's existence and the availability of the information associated with this information provider to a directory service (GRRP).
- **Sensor-Information Provider communication:** The GRIS communicates with a sensor via a well-defined API. Some sensors can also be information provider. In order to support easy extensibility, we do not discuss sensors that can measure the target system and provide instrumentation information through a pre-defined protocol.
- **Aggregate Directory:** the directory service is used to publish the location of information provider and its associated sensors. This allows the users to discover which sensors are currently active and which information provider they should contact to obtain information. In grid environment (GIIS), the directory accepts GRRP request from GRIS or other GIIS instances and merges these information services into a unified information space.
- **Archival System:** the storage system is used to hold historical data that can be used later for prediction and analysis. There are three components in Archival System: Data Importer, telemetry database

and information provider to publish the data. An importer can subscribe to an information provider and input the received sensor data into database. Database consists mostly of telemetry gathered by different sensors of different subjects. Some database might include derived parameters, statistics, and any other data element required by the Grid application. A Telemetry Database also acts as a server to answer all types of telemetry queries. An information provider is needed to read the data from this database per users' requests.

3 Grid Monitoring Architecture

In this section, we specify the system requirement for Grid monitoring, provide the Grid monitoring infrastructure, and describe the designing of each component in the system.

3.1 System Requirement

Due to the complexity and dynamics of grid computing model, the monitoring toolkits built on top of this computing model are also complex. To build an efficient and effective monitoring model, the designers and developers need to keep the following requirements in mind.

- The Monitoring toolkits can make use of existing monitoring tools. The efforts for incorporating a monitor tool should be minimum.
- Scalability: the system for monitoring and fault management should be scalable. The number of grid nodes will increase every year in order to satisfy the requirement of HENP. The monitoring system should be scalable for the growing grid system.
- Flexibility: the system for monitoring should be flexible because the target to be monitored and the grid architecture are likely to change over time.
- Extensibility and Modularity should be implemented, which allows users to include those components easily that they wish to use. All Communication flows should not flow through a single central component. Having a single, centralized repository for dynamic data causes two performance problems. The centralized repository for information represents a single-point-failure for the entire system. The centralized server can form a performance bottleneck.
- Non-intrusiveness: the grid monitoring system should incur as small system overhead as possible. It should not disrupt the normal running of the monitored system. This is extremely important if the monitoring system monitors a large number of target systems.
- Security: typically, an organization defines policies controlling who can access information about their resource. The monitoring system must comply with these policies.

- Ability of logging: Some important data should be archived.
- Inter-operability.

3.2 A Grid Monitoring Architecture Based on MDS

Grid monitoring infrastructure includes four basic components: Sensors, Archive System, Information Providers and Grid Information Browser. Figure 1 describes a typical monitoring scenario built on top

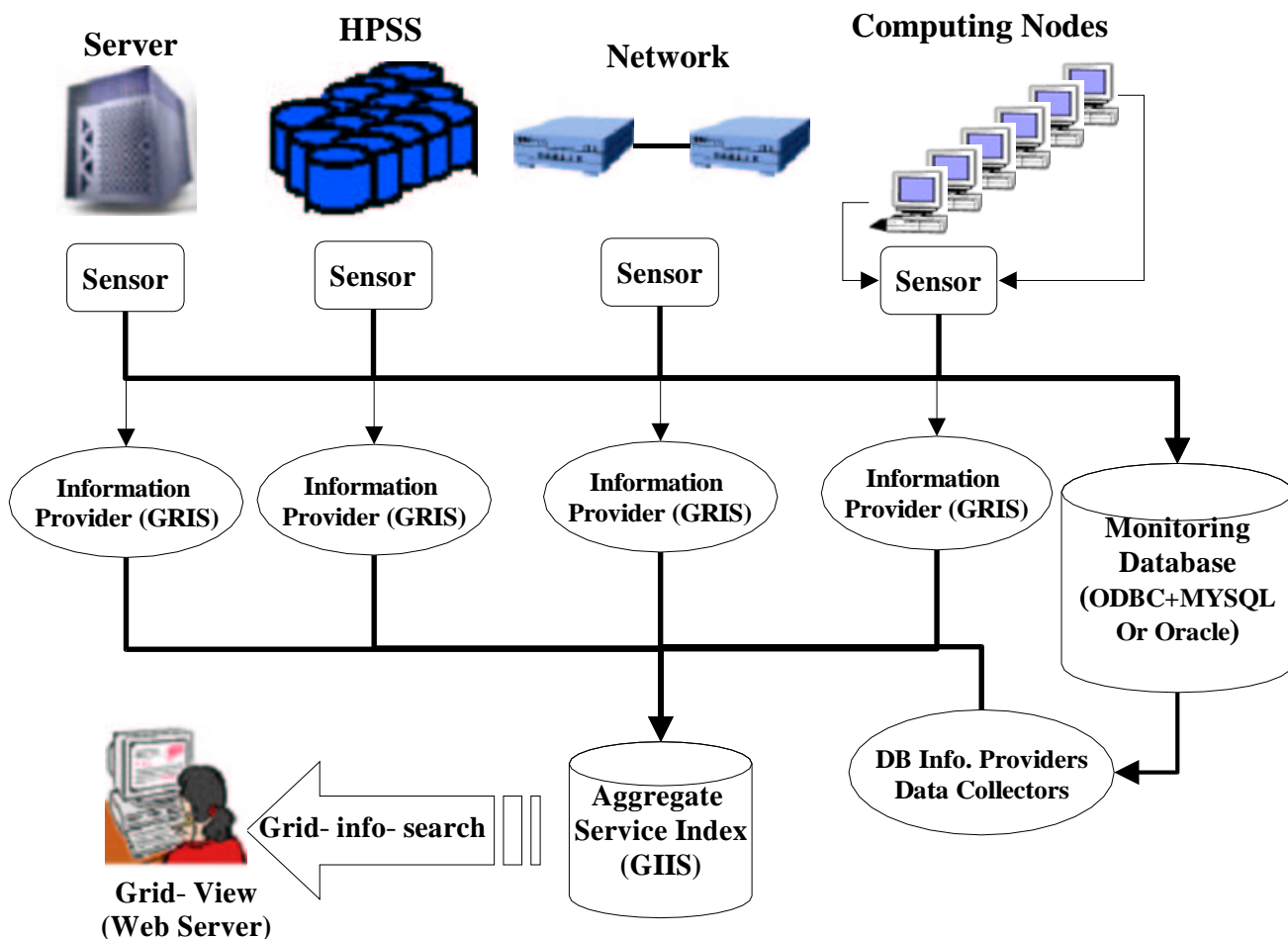


Figure 1: Grid Monitoring Framework

these basic components. In this architecture, the sensors are deployed in any computing facility that is monitored. These sensors might not locate in the same host as the information providers. For example, most of hosts in LSF cluster, condor pool do not have Globus or GRIS installed on them. Therefore, the sensors need to interact with the remote information providers, which usually locate in the globus-gatekeeper.

3.2.1 Sensor

3.2.2 Archive System

3.2.3 Information Provider

3.2.4 Grid Information Browser

GridView¹ is being developed at the University of Texas at Arlington (UTA) to monitor the U.S. ATLAS Grid. It was the first application software developed for the U.S. ATLAS Grid Testbed, released in March, 2001, as a demonstration of the Globus 1.1.3 toolkit. GridView provides a snapshot of dynamic parameters like cpu load, up time, and idle time for all Testbed sites. GridView has gone through two subsequent releases. First, in summer 2001, MDS information from GRIS/GIIS servers were added. Not all Testbed nodes run a MDS server. Therefore, the front page continues to be filled using basic Globus tools. MDS information is provided in additional pages linked from this front page, where available. Recently, a new version of GridView was released after the beta release of Globus 2.0 in November 2001. The U.S. ATLAS Testbed incorporates a few test servers running Globus 2.0 as well as every Testbed site running the stable 1.1.x version. GridView provides information about both types of systems integrated in a single page. Globus has changed the schema for MDS information with the new release. GridView can query and display either type. In addition, a MySQL server is used to store archived monitoring information. This historical information is also available through GridView. We will continue to develop GridView to match the needs of the U.S. ATLAS testbed. In the first quarter of 2002, we plan to set up a hierarchical GIIS server based on Globus 2.0 for the Testbed. The primary server will be at UTA which will collect and publish monitoring data for all participant nodes through MDS services. This GIIS server will also store historical data which can be used for resource allocation and scheduling decisions. Information will be provided for visualization through GridView and the Grappa portal. In the second quarter of 2002, it was planned to develop graphical tools for better organization of monitored information. Performance optimization of the monitoring scheme will be undertaken after the first experience from DC0 and DC1. Integration of various Grid services will be an important goal. Figure 2 shows the status of USATLAS grid testbed.

3.2.5 Desired Features of the System Architecture

Based on this system architecture, the sensors can be plugged into the monitoring architecture with minimum effort. We can simplify the sensor design because the sensors only need to talk to information provider to get their monitoring subject published to the authorized customers. The sensors do not need to know who wants to subscribe to this subject and the number of the subscribers. The subscribers (the consumers) can also be simplified because they just need to tell information provider what subject they are interested in. It is up to the information provider to pass the system subjects to the subscribers. By distributing the telemetry databases and MDS servers in different location, we can avoid the problems caused by a centralized server.

¹<http://heppc1.uta.edu/atlas/grid-status/index.html>

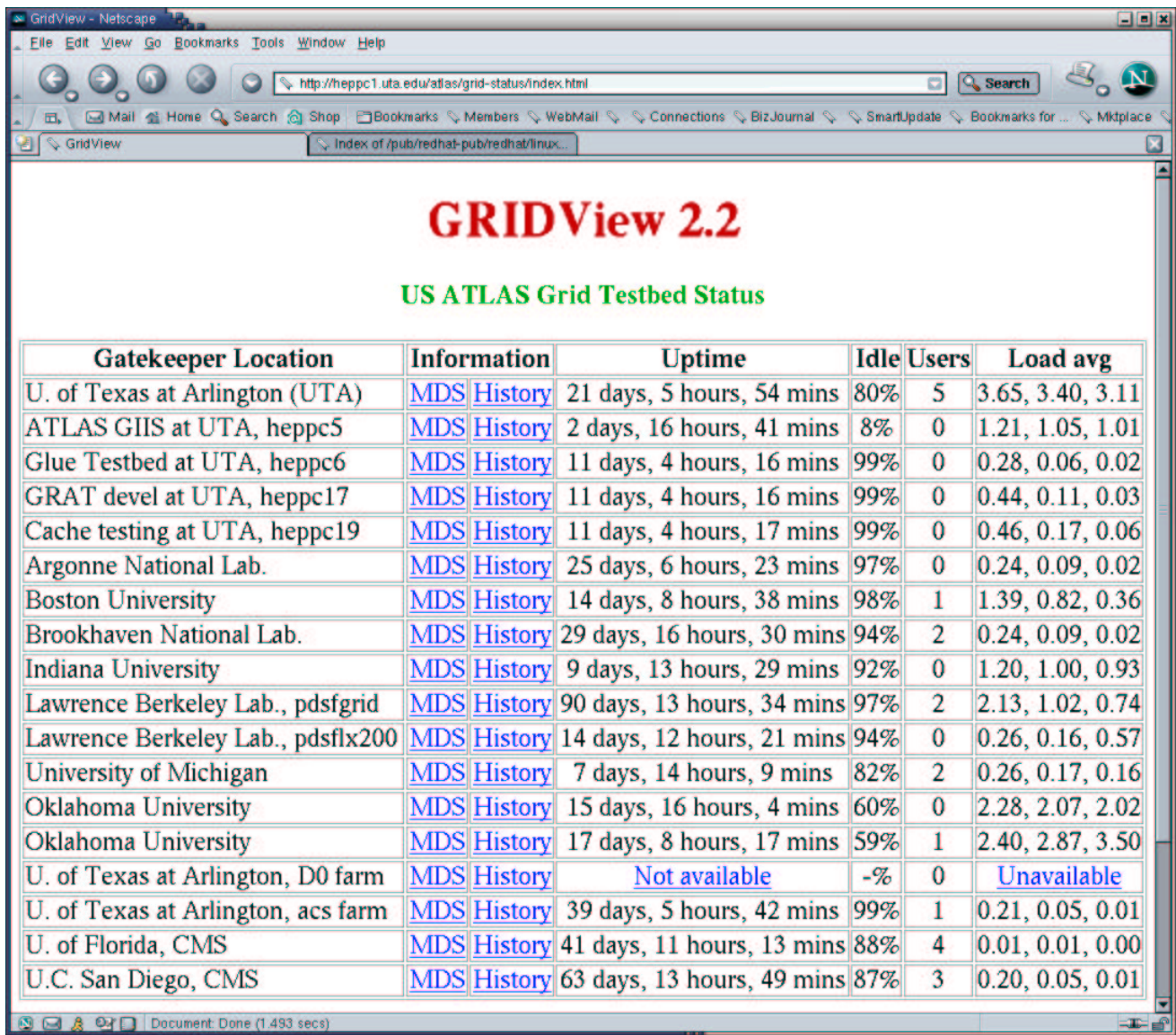


Figure 2: Grid View Interface

- Information Provider provides cache for the newest value from the mysql database.
- Non-intrusiveness: Information provider can eliminate the user random accesses to the database server.
- Scalability can be significantly increased. 800 linux nodes are being monitored Network connectivity of eight USATLAS testbeds.
- Flexibility: Independent on Sensors. Many sensors can be easily plugged as long it has well defined protocol and API. Archive system is independent to underlying database. They can be RDBMS, Oracle, MySql, Sybase, Informix, flat files, objectivity as long the ODBC drivers is available.

4 Performance Evaluation

5 Related Works

Network Weather Service (NSW): The goal of the Network Weather Service is to provide accurate forecasts of dynamically changing performance characteristics from a distributed set of meta-computing resources. It can produce short-term performance forecast based on historical performance measurement. The Network Weather Service attempts to use both extant performance monitoring utilities and active resource occupancy to measure the performance. It can measure the fraction of CPU time available for new processes, TCP connection time, end-to-end TCP network latency, and end-to-end TCP network bandwidth. It has NWS sensors, CPU sensors, network sensors. It also has predictors that forecast the system performance. We will incorporate the NSW in our Grid monitor toolkits. We can pull out the sensor modules and predicting modules and put them into the monitor architecture. We also need to design the interface that can bridge the communication between the sensors and the Telemetry Database.

Simple Network Management Protocol (SNMP): Since SNMP was developed in 1988, the Simple Network Management Protocol has become the standard for inter-network management. Because it is a simple solution, requiring little code to implement, we can easily build SNMP sensors for our monitoring architecture. SNMP is extensible, allowing us to easily add network management functions to the monitoring system. SNMP also separates the management architecture from the architecture of the hardware devices, which broadens the arena of our monitoring architecture. SNMP is widely available today and has extensive support from academic, vendors and other research institutes. Therefore, SNMP based tools are widely used for network monitoring and management. SNMP based tools and sensors should be evaluated for grid monitoring architecture.

Grid Monitoring Architecture (GMA):

Metacomputing Directory Service, Monitoring and Discovery Service (MDS):

JINI

6 Conclusion

References

- [1] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proceedings of 10th IEEE International Symposium on High Performance Distributed Computing (HDPC-10)*, IEEE Press, San Francisco, California, August 2001.
- [2] R.L. Ribler, J.S. Vetter, H. Simitci, and D. A. Reed. Autopilot: Adaptive Control of distributed applications. In *Proceedings of 7th IEEE International Symposium on High Performance Distributed Computing (HDPC-10)*, IEEE Press, 1998.

- [3] B. Tierney, B. Crowley, D. Gunter, J. Lee, and M. Thompson. A Monitoring Sensor Management System for Grid Environments. *Cluster Computing Journal*, 4(1), 2001.
- [4] B. L. Tierney. The NetLogger Toolkit: End-to-End Monitoring and Analysis of Distributed Systems. <http://www-didc.lbl.gov/NetLogger/>.
- [5] R. Wolski, N. Spring, and J. Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. *Journal of Future Generation Computing Systems*, 15, October 1999.