Abstract. Grid is distributed, heterogeneous environment comprised of various resources. Monitoring of grid resources is essential for realizing functionalities such as failure detection, application performance analysis, debugging, execution management, etc. Today many mature monitoring systems exist, but there are no complete solutions and functionalities of existing systems often overlap.

In this article, we propose methodology of integration of various existing systems in order to provide rich information base for applications, users and administrators. Furthermore, we extend this system with following functionalities: additional sensors, data archiving, integrated web interface and tools for installation, configuration and maintenance of monitoring components.

Keywords. Grid, monitoring, information systems, CRO-GRID.

1. Introduction

Grid is a large-scale, distributed environment, which enables execution of computationally demanding scientific and engineering applications. Resources used for application execution can dynamically join and leave grid. For efficient utilization of these resources, monitoring is needed. Monitoring allows system to detect anomalies, enables performance analysis and application tuning, provides information to other parts of the grid (e.g. execution or data management systems), etc.

Grid monitoring is defined as the process of measurement and publication of the state of a grid resource at a particular point in time [8]. In the grid context, systems that perform monitoring are also called information systems. Components that comprise monitoring systems are sensors, sensor management, information management, data archiving and presentation.

Sensor is a process that performs actual monitoring of resources. Data that sensor generates in one measurement is called event. There are several types of sensors: host, network, application, general and custom-made. Based on provided type of sensors, some systems are more appropriate for particular use. Another component is sensor management, which allows user to remotely configure and control sensors. Information management component enables sensors to publish events, as well as consumers to discover data of interest. Information management uses some sort of a database to store information about available sensors and monitoring events. Data archiving is necessary for postmortem performance analysis, accounting and resource utilization statistics. Presentation component enables data analysis, graphical view of monitored data, etc.

Up to date, there is no system that implements all described components. Some systems are oriented towards specific type of sensors and sensor management (e.g. Ganglia, NWS), while others focus on information management (e.g. MDS, MonALISA). Most of the existing grids today combine several monitoring systems.

The remainder of the paper is structured as follows. Grid standards related to monitoring are listed in section 2. In section 3, we describe requirements for monitoring system and overview of existing systems. CRO-GRID grid environment and monitoring needs are described in section 4. In section 5, we depict our proposed systems in detail. Conclusions are given in section 6 and future work in section 7.

2. Grid standards

Grid Monitoring Architecture (GMA) is architecture for grid monitoring systems proposed by Grid Global Forum's (GGF) [5] GMA group. Fig 1 shows three components that comprise GMA and interaction between them.

Consumer is service representing the entity (e.g. user, grid subsystem, application) that demands monitoring of certain resources. Producer is service that represents sensor and
publishes events. Directory service stores information about Producers and Consumers. Both Consumers and Producers can register to the Directory. While Producers register information about monitored events, Consumers can register events of interest. In the same way, both of these components use Directory to discover interested entities.

3. Grid monitoring systems

Basic requirements ([8], [15]) for grid monitoring system are described below.

Sensor management. System has to provide mechanisms for configuration and control of remote sensors.

Publish and discovery. Monitoring system has to enable mechanisms for sensors to publish information and for users to discover information about resources.

Data archiving. Monitored data repository should be implemented, which can be used for application performance analysis and resource utilization statistics.

Non-intrusiveness. Monitoring system has to be designed in a way that measurement process does not affect monitored system.

Extensibility. Ability to extend set of basic functionalities is necessary. For example, it should be possible to design sensors for specific resources or implement alternative message passing protocols.

Interoperability. Monitoring system should be able to cooperate with other grid middleware systems, such as execution, data and security management. In addition, it should be able to cooperate with other monitoring systems.

Scalability. System architecture has to be able to support large-scale, dynamic environment, comprised of numerous monitored resources.

Security. Grid monitoring system has to provide basic security mechanisms, such as authentication, authorization and secure message transport. System can implement security internally or use external security system.

In the rest of this section, we shortly describe existing grid monitoring systems. Systems which are used in our monitoring architecture, are described in details.

3.1. Ganglia

Ganglia [9] is Java based system that performs host monitoring. It is widely used for cluster monitoring. Main Ganglia features are: multicast based communication, hierarchical information management, web interface, extensibility, XML data format usage and interoperability with Globus [6] MDS. Ganglia maintains data archive in Round Robin Database (RRD).

Ganglia comprises of three components: Gmond, Gmetad and graphical web portal. Gmond is component that performs hosts...
monitoring. Gmetad is used for aggregation of data from Gmond or Gmetad services.

3.2. NWS

Network Weather Service (NWS) [14] is a system for host and network monitoring. Beside for monitoring, NWS has the ability to predict resources behavior in close future time. Information is stored in LDAP-like database. In addition, it is possible to integrate NWS with MDS and it is used by various execution management systems (e.g. APST, Gridbus, etc). Furthermore, NWS enables remote sensor management. However, NWS lacks support for security mechanisms.

3.3. INCA

INCA Test Harness and Reporting Framework [11] performs automated testing, verification, benchmarking and information management of grid middleware services and other software installed on resources. It is mainly used to verify that all grid software components are properly installed and operational. INCA can be integrated with MDS system and uses RRD for data archiving. It is still under development.

3.4. NetLogger

The Network Application Logger Toolkit (NetLogger) [13] is system for application monitoring. NetLogger provides an API that grid application developer uses to add sensors in application source code. Once the monitoring event take place, system adds timestamp and some additional host and network information. This data is then transferred using XML based protocol to remote computer (e.g. computer from which the job was submitted). NetLogger provides basic network and host sensors and graphical interface for presentation of events, but lacks security mechanisms.

3.5. MonaALISA

MonALISA [10] is grid monitoring system based on Jini technology. MonALISA doesn’t provide sensors, but only information management, archiving and presentation components. Due to the Jini based implementation, it is very scalable, autonomous, and easy to install. Integration with various monitoring systems and protocols is possible. MonALISA lacks security mechanisms and interoperability with grid middleware systems.

3.6. Globus MDS

Monitoring and Discovery System (MDS) [4] is monitoring system developed as a part of Globus Toolkit. MDS's main goal is information management, although it provides simple host monitoring sensors. Usually, sensors are provided by other monitoring systems, such as Ganglia and NWS. There are several versions of MDS. MDS2 is LDAP based implementation which is part of Globus version 2.x. MDS3 and MDS4 are OGSA based implementations which are part of Globus versions 3.x and 4.x.

MDS2 system is comprised of Information Provider (IP), Grid Resource Information Service (GRIS) and Grid Index Information Service (GIIS) services. GRIS service gathers information from sensors (IPs) on a host. GIIS is sort of a Directory service, which enables GRIS and GIIS services to register monitoring events.

4. CRO-GRID grid environment

CRO-GRID [2] is Croatian grid initiative, which consists of three projects and aims to build grid for science and enterprise needs. CRO-GRID Infrastructure [3] focuses on building and maintaining grid infrastructure. CRO-GRID Mediator is developing service-oriented grid middleware based on WSRF specification. Three scientific applications and grid portal are being developed within CRO-GRID Application.

At this moment, CRO-GRID infrastructure consists of five clusters placed in institutes in four cities. Network used for CRO-GRID infrastructure is shown on Fig 2. Gigabit links between sites are provided by Giga CARNet project. Cluster job management systems (JMS) Sun Grid Engine (SGE) and Portable Batch System (PBS) are installed on clusters. Globus version 3.2 pre-WS services are used as basic grid middleware.
4.1. Monitored data

Analysis of our monitoring demands has shown that we need following information:

- Information about hosts, such as availability, free memory, cpu, etc. Information are used for fault detection, accounting and performance analysis. Data should be used both real-time and from archive.
- Extensive information from cluster JMS are needed. Examples are: information about jobs, available reservations, nodes, etc. Since all resources are controlled by JMSs this data is crucial for execution management system.
- Information about network links, namely network bandwidth and latency.
- Application monitoring is not a priority at this moment. However, basic infrastructure which will enable developers to add sensors to applications, needs to be deployed.

5. CRO-GRID Monitoring Architecture

After analysis of CRO-GRID monitoring needs and existing systems we decided to combine several existing solutions and provide some additional functionalities. Main functionalities we plan to add are monitoring components management, data archiving and web interface.

We choose to use existing systems that provide sensors and information management, because many of them have already been used in production environment and satisfy our requirements.

5.1. External monitoring systems

After detailed review, we decided to use following grid monitoring systems: Ganglia, NWS, INCA and MDS. All these systems are already deployed on resources as described in following section. Here we describe rationale for choosing these systems.

Major benefits of Ganglia are: it is already deployed on all clusters as a part of cluster distribution, easy maintenance, intuitive web interface and interoperability with MDS. It lacks sensor management and security. Currently, we are using Ganglia both as standalone system and integrated with MDS. We have deployed central Gmetad so that we utilize Ganglia’s web interface and data archiving.

NWS benefits are sensor management, network monitoring, behavior predictions and integration with execution management systems. Currently we have deployed NWS as standalone system because some execution management cannot use it as a part of MDS. In future, we are planning to integrate it completely with MDS.

INCA allows us to check integrity of software stack installation and grid middleware services. Problem is that INCA is still under development, but we find it stable enough for usage.

MDS is de facto standard system for information management. MDS’s advantages are: hierarchical information management, extensibility and integration with numerous monitoring systems and Globus GSI security system. However, MDS has efficiency issues, e.g. it needs to be configured cautiously in order to provide as fresh data as possible. We are planning to use GLUE information schema for describing data in MDS.
5.2. Proposed architecture

Description of the proposed monitoring architecture is divided in two parts. In first part, we describe deployment of monitoring components on individual clusters. In second part, we describe integration of cluster monitoring in grid monitoring architecture.

![Figure 3. CRO-GRID cluster monitoring](image)

Deployment of monitoring components on cluster is shown on Fig 3. Ganglia’s Gmond services are installed on all nodes and they perform host monitoring. Ganglia’s Gmetad, NWS, INCA and MDS’s GRIS and GIIS services are deployed on cluster’s frontend. Gmetad service gathers information from nodes, stores it in RRD database, reports information to GRIS and to the central Gmond service. Currently, both GRIS and site GIIS are deployed on frontend, since there are no other resources on site. Once site adds new resources, GIIS will be placed on dedicated server.

Components that we propose (colored in dark gray) are Monitoring Manager (MM) and JMS Monitor. MM is responsible for control over all the monitoring components installed on frontend. JMS Monitor will extract information from JMS and report it to GRIS.

Deployment of monitoring components on entire grid system is shown on Fig 4. At this moment, we propose that single server is dedicated for placement of central monitoring services. However, all these systems can be deployed on separate servers. Also, it is possible to introduce another level of hierarchy for all of them. Central NWS and Gmetad are marked with dashed line because they are optional.

![Figure 4. CRO-GRID grid monitoring](image)

Two components that we propose in this part of the architecture are Data Archive and Web Interface. Data Archive component will provide archiving functionalities to MDS system and integrate archives from other systems. Web interface will allow users to analyze real-time monitored data or archived data, as well as administrators to manage monitoring components on resources.

5.3. JMS Monitoring

Selected external systems provide quality host, network and application sensors. However, most of them do not provide sensors for cluster JMS, which is critical for execution management system. We are planning to extend MDS system with JMS Monitor, which will provide sensors for SGE and PBS systems. Currently, sensors for PBS exist, but they do not provide data in depth (e.g. they lack information about reservations). JMS Monitor will use GLUE schema for description of published data.

5.4. Monitoring components management

As mentioned before, sensor management is one of the required functionalities of monitoring system. We find that it is necessary to enable remote management of monitoring components placed on clusters. NWS is currently the only system which allows such control.

In order to enable remote management of other three systems and any future systems we propose Monitoring Manager (MM) component. MM is placed on all resources. Once deployed, MM will pull information from central server and deploy selected monitoring components. Furthermore, it will monitor and update components and allow administrators to manage them remotely. We are planning to implement it
as a web service in order to achieve interoperability.

MM component is in early state of development. At this point, we use Grid Packaging Tool (GPT) for packaging monitoring tools. GPT is tightly integrated with Globus Toolkit and enables installation, configuration and update of tools as non-privileged user. Temporarily, central component is used to manage monitoring software on resources. This component uses Globus Toolkit tools for reliable access to remote resources.

5.5. Data archiving

We find that the one of the big issues of monitoring systems integration is data archiving. Some of the selected systems implement data archiving, but they all use different databases. Ganglia and INCA use RRD for storing data, NWS uses flat files, while MDS do not have archiving capability at all.

We are planning to design our own Data Archive component, which will enable archiving for systems that don’t have that capability. In addition, component will be able to integrate archived data from other systems, such as Ganglia and NWS (b on Fig 4). Our first step is to implement MDS information archiving (a on Fig 4).

5.6. Web interface

Information gathered by monitoring system are used by applications, grid middleware components and users. For users it is necessary to implement graphical interface. Interface will enable users to access data from used monitoring systems (1 on Fig 4) and retrieve archived data from Data Archive component (3 on Fig 4).

Special part of web interface will be provided for grid administrators. This part will be linked with MM components and allow remote management of monitoring components (2 on Fig 4).

![Figure 5. CRO-GRID Ganglia web interface](image-url)

At this point, we use Ganglia (show on Fig. 5) and INCA web interfaces. We developed simple web interface for network bandwidth and latency reports extracted from NWS. Our next steps will be to use Ganglia interface for data provided by NWS. Furthermore, we will extend Ganglia web interface. In current version, interface can only show information for last hour, day, week or year. We will implement the ability to browse stored data in more details.

Currently, we are also investigating approaches for implementation of JMS monitored data interface.
6. Conclusion

Resource monitoring is necessity for dynamic environment such as grid. Many mature systems exist today. However, none of them provides complete solution, which covers all requirements needed for efficient grid monitoring.

In this article, we propose grid monitoring architecture based on CRO-GRID demands. We focus on following issues: integrating several existing systems, providing component for management of monitoring components, adding sensors for monitoring of cluster JMS, integrating and extending data archiving capability and designing a web interface for management. First step, deployment of described monitoring systems has already been successfully accomplished.

7. Future Work

In future, we are planning to extend system in following directions:
- Enabling management of other grid services deployed on resources.
- Integrating NetLogger application monitoring system in proposed architecture.
- Integrating with grid middleware systems other then Globus (e.g. UNICORE, Condor).
- Integrating GGF’s standards related to monitoring systems.
- Linking monitoring system with CRO-GRID SEcurity Management system. This approach will allow us to achieve optimum security.
- Integrating monitoring web interface with grid portal, which is being developed by CRO-GRID Application project.

We also plan to investigate approaches for achieving autonomous system. Such system would be able to self-recover, self-configure, automatically expand on new resources, etc. This functionality will be achieved by extending MM component with ability to recover from resource failure and seek for configuration on known remote computers. This component can also be used to manage other grid services, such as execution or data management services.

At this point of development, we are focused on achieving set of functionalities needed by our users. Once we finish this stage, we will concentrate more on performance of whole system. We intend to conduct precise benchmarking and measurement of network and resource overhead introduced by monitoring components.

8. Acknowledgements

All given results were achieved through the work on the TEST program – Technological research-development projects with the support of the Ministry of science, education and sports.

9. References


