

# GridView: A Dynamic and Visual Grid Monitoring System

Ni Guangbao, Ma Jie, Li Bo

*Institute of Computing Technology, Chinese Academy of Sciences*

*P.O. Box 2704, Beijing 100080, P.R. China*

*Graduate School of the Chinese Academy of Sciences, Beijing 100039*

*Emails: ngb@ncic.ac.cn, majie@ncic.ac.cn, libo@ncic.ac.cn*

## Abstract

*Computational and Data Grids consist of the coordinated utilization of large sets of diverse, geographically distributed resources for high performance computation. In order to make better use of these computing entities, a substantial amount of monitoring data is collected for a variety of tasks. The large number of heterogeneous resources available in Grids makes the task challenging. The processing of mass data is another problem with large amounts of monitoring data increasing. In this paper, a monitoring system in Grid environment called GridView is presented. It gathers much important information from a large computing facility. For giving more efficient performance evaluation, comprehensive computation capacity (C3) model is set up. GridView also gives mass data management mechanism and mass data visualization technique. Due to the Grid environment, GridView must consider dynamic changes and heterogeneous characteristic of different platforms.*

## 1. Introduction

Grid will integrate large sets of diverse, geographically distributed resources into a huge virtual supercomputer, and implement resources sharing. The goal of Grid is to eliminate isolated island of information, and make users transparently utilize diverse resources and easily and efficiently get desired information. Therefore various Grids such as computational Grid, data Grid and information Grid are widely used.

The success of the Grid depends greatly on efficient utilization of these resources [1]. Monitoring is necessary for Grid. Some monitoring systems have already achieved information collecting, and they are

scalable, efficient and low-impact. But most of them don't consider about the dynamic and large-scaled characteristics in the Grid. The research in these aspects becomes a key problem.

GridView is introduced in this paper, which can monitor and acquire the information and status of the resources joined it. GridView is a scalable, heterogeneous and dynamic Grid monitoring system. It still has more features such as the comprehensive computation capacity model, mass data management mechanism and mass data visualization technique.

Some key issues are discussed as following in this paper:

- Four-tier structure. In GridView, in order to monitor the heterogeneous resources and dynamic changes in Grid environment, four-tier structure is designed. Four-tier structure consists of detecting interface layer, cluster information gateway layer, Grid information management layer, and Grid view layer. Detecting interface layer gathers monitoring data. It can also acquire monitoring data from other cluster monitoring systems through its API. Cluster information gateway layer hides heterogeneous information, which makes GridView monitor heterogeneous resources easily. Grid information management layer monitors dynamic changes of Grid nodes.
- Comprehensive computation capacity model (C3 model). In some Grid or cluster monitoring systems, they simply sum up or average every node remaining capacity in calculating CPU processing capacity. But this algorithm is imprecise, even sometimes incorrect. In order to get more precise performance evaluation of the entire Grid and clusters, C3 model is presented. It provides the precise computation capacity model in calculating CPU processing capacity. Furthermore, C3 model can be extent to calculating memory and swap

partition. Moreover, C3 model also reflects the mutual influence of different resources.

- Mass data management mechanism. With increasing of the monitoring time and growing of Grid resources, mass data will be collected. In order to efficiently utilize the storage space, GridView adopts the redundancy compression technology that greatly decreases the required space and has not the loss of the precision. This compression technology doesn't contradict the database operation. When GridView uses these data, these data must be recovered firstly.
- Mass data visualization technique. Most of Grid monitoring systems just directly display the data gathered. They don't give overall performance characteristic for mass data. GridView gives the some kinds of visualization representations in this aspect such as the load-distributed view, status-distributed view and geographic view.

The rest of this paper is structured as follows. Section 2 briefly summarizes recent work on Grid monitoring and Grid information visualization representation. Section 3 describes some design issues. Then the design and implementation provides in section 4. In the final section, the paper is summarized briefly.

## 2. Related Work

Monitoring computer system has had a long history in the literature, and recent interest in computing Grid has led to work on specialized monitors for it [2]

**Ganglia:** Ganglia is a scalable distributed monitoring system designed for high performance computing systems. [1,2]. Ganglia use recursive languages to represent monitoring data, namely XML. The protocol is similar to our wide-area monitoring data transport protocol. Ganglia uses RRDtool [4, 7] for data storage and visualization. Although the RRD time-series database system is efficient, Gmeta's use of them leads to a performance bottleneck and the form of visualization representation is simple. But we provide a more visualization representation about information.

**Grid Monitoring Architecture (GMA)** [5]: The Grid Monitoring Architecture consists of three components: directory service, producer and consumer. Grid Monitoring Architecture supports both a streaming publish/subscribe model, and query/response model. Producers publish their existence, description and type of performance data to the directory service. Consumers query the directory service and subscribe to the selected producer. The time-stamped performance data, called events, are directly sent from the producers to consumers based on

subscription entries stored at the directory service. Communication of control messages and transfer of performance data occurs directly between each consumer/producer pair without further involvement of the directory service.

## 3. Design Issues

GridView is designed to collect information of heterogeneous resources utilizing and configuration information of distributed resources in the Grid environment. How to monitor these heterogeneous resources and find their dynamic changes is a considerable problem. With monitoring time and resources increasing, the mass data management and visualization must be solved. These factors result in several design issues in the GridView.

### 3.1. Four-Tier Structure

The system for monitoring should be flexible because the target to be monitored is likely to change over time. The dynamic characteristics of the Grid also allow the computing resources to participate and withdraw from the resource pool frequently. The four-tier structure that we designed is beneficial to implement the function of dynamic and heterogeneous characteristics. GridView provides the API in the Grid information manager layer for this purpose. When a Grid node wants to join GridView, its basic information will be transmit to Grid Information manager through the API. Then GridView will perform the monitoring to the Grid node. Similarly, when a Grid node wants to withdraw from the Grid monitoring system, it can notify an administrator, and the administrator can remove it through the API.

Due to the probable heterogeneous environment, the implementation of the cluster information gateway layer must consider the problem that the data transmitting meets with heterogeneous platforms. Another consideration is diversity of clusters so that data length and uncertainty of monitored items of different clusters vary greatly with them, so the implementation can't use the fixed data structure. GridView adopts the text transmission conformed to the standard of XML between the cluster information gateway layer and the Grid information manager layer.

### 3.2. Comprehensive Computation Capacity Model

Comprehensive computation capacity model (C<sup>3</sup> model) is used to provide the more precise performance evaluation of the entire Grid and clusters.

For combinations of different scale resources, there are different formulae to calculate  $C^3$ .

For a single resource, its  $C^3$  formula is:

$$R_{C^3} = \begin{cases} (1 - R), & R \leq R_1 \\ (1 - R) \times \alpha, & R_1 < R \leq R_2 \\ 0, & R > R_2 \end{cases}$$

Where  $R_{C^3} = C^3$  of some resources (including CPU, memory, swap partition),  $R$  = a certain resource usage,  $R_1$  = a lower threshold of a certain resource usage,  $R_2$  = an upper threshold of a certain resource usage. When the alphabet  $R$  is replaced by  $P$ ,  $M$  and  $S$ , it represents  $C^3$  of CPU,  $C^3$  of memory and  $C^3$  of swap partition respectively.

For the single node of a cluster,  $C^3$  model adapts the following formula:

$$N_{C^3} = \begin{cases} P_{C^3}, & \text{when } M_{C^3} \geq \text{freepages.high} \\ P_{C^3} * \beta, & \text{when } \text{freepages.low} \leq M_{C^3} < \text{freepages.high} \\ 0, & \text{when } M_{C^3} < \text{freepages.low} \end{cases}$$

where  $N_{C^3} = C^3$  of cluster nodes,  $P_{C^3} = \text{CPU } C^3$ ,  $M_{C^3} = \text{memory } C^3$ .

The principle of divided function is based on the memory management of Linux. According to it, Linux starts background swapping if free pages of memory fall below the attribute high of data structure freepages, and Linux begins intensive swapping below the attribute low of data structure freepages.

For cabinets, logical groups defined by users or the entire clusters, the  $C^3$  formula is shown as following:

$$C_{C^3} = (\sum N_{C^3}) / C$$

Where  $C_{C^3} = C^3$  of a certain scale,  $N_{C^3} = C^3$  of cluster nodes,  $C$  = the node count in the scale.

### 3.3. Mass Data Management Mechanism

Owing to the GridView monitoring many Grid nodes which each maybe be a large-scale cluster, this will lead to production of mass data. How to store and retrieve all of these mass data is a considerable problem. Before storing data to database, GridView compares the current collecting data with the previous a frame data in the memory. Only if the changed data with time-stamped save into the database. At the same time, to avoid the data too much in a table, which is harmful to query the data, GridView stores monitoring data into different tables on day.

When GridView uses these data, it must recover them firstly. According to the algorithm of data

compression that stores a integrated data frame at the first frame data and at the beginning of every hour, when the data is compressed that the GridView wants, GridView only forwardly searches data at an exact o'clock, and then can get the first data at the appointed time span. The following data can recover from the first data according to the data collecting cycle.

### 3.4. Mass Data Visualization Technique

GridView adopts the mechanism of three-layer view model. The three-layer view model is Data Extraction Layer (DEL), Data Mapping Layer (DML) and Data Visualization Layer (DVL).

Data Extraction Layer derived data from database. Data Mapping Layer filters and recovers data transmitted from DEL according to the demand of different views of DVL. Data Visualization Layer is to visualize the information of resource utilizing and configuration information of Grid nodes, which can be greatly convenient to an administrator. DVL provides the function of zooming in and out, so you can get discretionary granular information of the whole Grid.

## 4. Design and Implementation

GridView has been used to monitor the part of DAWNING 4000A [10] high-performance Grid-enabling computer, which will consist of 640 SMP nodes based on the AMD 64-bit processor, more than 2 terabytes memory capacity and 30 terabytes disk space and connected with the 1024-port Myrinet. It has a four-tier architecture as shown in figure 1: status and information detecting interface layer, cluster information gateway layer, Grid information management layer, Grid view layer.

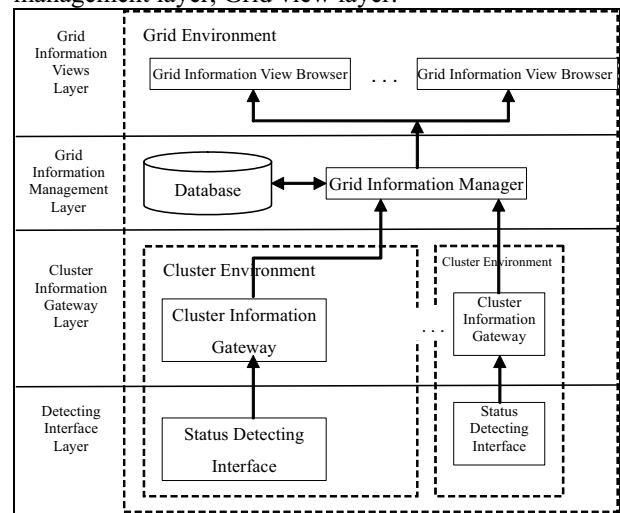


Figure 1. System Architecture

#### 4.1. Detecting Interface Layer

Status and information detecting interface layer communicates with low-level collecting daemon, which measures the characteristics of a target system and generates a time-stamped performance statistics such as CPU usage, memory usage and network traffic, even monitoring and capturing abnormal system status, some application programs status and so on. Status detecting interface can get the desired data according to predefined format through it. In our experience system, nodes in a cluster are divided into physical partitions, which avoid all monitoring data flowing through a central component that represents a potential bottleneck. Thus, it is easily scalable for a large-scale cluster. At the same time, by detecting interface layer, GridView can acquire monitoring data from other cluster monitoring system. The low-level data collecting system is beyond the scope of this paper.

#### 4.2. Cluster Information Gateway Layer

Cluster information gateway layer acquires the information of resources utilizing and configuration information of the corresponding cluster by low-level detecting service and transmits the information to the Grid information manager layer. It periodically gathers the current statuses of all of the nodes in the cluster, and continually refreshes the previous frame data in the memory. Cluster information gateway layer will mask heterogeneous information and transform various data into a standard format.

#### 4.3. Grid Information Management Layer

Grid information management layer locates in Grid environment, and it consists of Grid information manager and a database that stores the data of the information of resources utilizing and configuration information of clusters. Grid information management layer monitors dynamic changes of Grid nodes and monitors all of the clusters that participate.

#### 4.4. Grid View Layer

The Grid information view provides homogeneous views of the underlying heterogeneous data [6]. It provides a visual tool for an administrator to get knowledge of resources utilizing and configuration information of clusters. GridView implements four-category views: static views, dynamic views, distributed views and data comparison views. Of them, Distributed views are emphasized. It want to supply overall information of load balance of all nodes in a cluster. It uses gradient colors to represent the resource usage. The darker color used is, the higher the resource usage is.

## 5. Summary

GridView is a Grid monitoring system that collects information of resources utilizing and configuration information of different Grid nodes from low-level data collecting system. This paper describes its architecture and implementation. Several design issues are presented, including four-tier structure, comprehensive computation capacity model (C3 model), mass data management mechanism and mass data visualization technique. Now the project is under way. We expect to perfect its function later.

The design of GridView is not intended to interact with application; rather it is designed to monitor the resources that an application may use. In order to provide the visual information and make an administrator to get desired information easily, GridView presents visualization representation technique of mass data and provides four kinds views: static views, dynamic views, distributed views and data comparison views

## References

- [1] Rich Baker, Dantong Yu, et al. GridMonitor: Integration of Large Scale Facility Fabric Monitoring with Meta Data Service in Grid Environment. Computing in High Energy and Nuclear Physics, California, March, 2003.
- [2] Federico D. Sacerdoti, Mason J. katz, and Matthew L. Massie, David E. Culler. Wide Area Cluster Monitoring with Ganglia.
- [3] Ganglia, <http://ganglia.sourceforge.net/>.
- [4] The RRDtool time-series data archiving system. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool>
- [5] Ruth Aydt, Dan Gunter et al. A Grid Monitoring Architecture. December, 2001.
- [6] Mark Baker, Garry Smith. GridRM: An Extensible Resource Monitoring System. June 2003.
- [7] Massie, M. L., Chun, B. N., and Culler, D. E. "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience", submitted for publication, February 2003.
- [8] <http://java.sun.com/products/jdbc/>, June 2003
- [9] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. Intl. Journal of Supercomputing Applications, 2001. <http://www.globus.org/research/papers/anatomy.pdf>.
- [10] Zhiwei Xu, Ninghui Sun, Dan Meng et al. "Cluster and Grid Superservers: The Dawning Experience in China", Proc of the 3rd IEEE Int'l Conf on Cluster Computing, 2003.