7

Administration and Monitoring Service for Storage Virtualization in Grid Environments

Salam Traboulsi
Traboulsi.salam@gmail.com

1. Introduction

A grid is collection of computers and storage resources maintained to serve the needs of some community (Foster et al., 2002). It addresses collaboration, data sharing, cycle sharing and other patterns of interaction that involve distributed resources.

Actually, the de facto building block is for high performance storage systems. In grid context, the scale and the reliability have key issues as many independently failing and unreliable components need to be continuously accounted for and managed over time (Porter & Katz, 2006). Manageability also becomes of paramount importance, since nowadays the grid commonly consists of hundred or even thousands of storage and computing nodes (Foster et al., 2002). One of the key challenges faced by high performance storage system is scalable administration and monitoring of system state.

A monitoring system captures subset of interactions amongst the myriad of computational nodes, links, and storage devices. These interactions are interpreted in order to improve performance in grid environment.

On a grid scale basis, ViSaGe aims at providing to the grid users a transparent, reliable and powerful storage system underpinned by a storage virtualization layer. ViSaGe is based on three services, namely as: administration and monitoring service, storage virtualization service and distributed file system. The virtualization service incorporates storage resources distributed on the grid in virtual spaces. These virtual spaces will be attributed by the distributed file system various qualities of service and data placement policies.

In this paper, we present our scalable distributed system: Admon. Admon consists of an administration module that manages virtual storage resources according to their workloads based on the information collected by a monitoring module. It is known that the performance of a system depends deeply on the characteristics of its workload. Usually, the node's workload is associated to the service response time of a storage application. As the workload increases, the service response time becomes longer. However, the utilization percentage of system resources (CPU load, the Disk load and Network load) must be taken into more consideration. Therefore, Admon traces ViSaGe's applications and collects system resources (CPU, Disk, Network…) percentage of utilization. It is characterized by an automatic instrumentation. It is an auto-manager monitoring system. In this paper, we demonstrate Admon efficiency due to nodes workload and user constraint like CPU usage workload.

The remainder of this paper is composed as follows: Section 2 presents the existing monitoring systems and motivates the need of Admon. Section 3 illustrates ViSaGe and grid environment, followed by Admon architecture, in Section 4, where we delve into details of Admon functionalities and API. In Section 5, we show a performance experiment tests for an Admon evaluation in a storage virtualization system. Finally, we conclude the paper in Section 6 and some future work is also presented.

2. Related work

In a distributed system such as ViSaGe, the storage management is principally funded on automatic decisions to improve performance. These decisions identify nodes which can be accessed efficiently. Therefore, analyzing node's workload is primary to make an adequate decision. Moreover, this workload is mainly supported by the monitoring systems. Several existing monitoring systems are available for monitoring grid resources (computing resources, storage resources, networks) and grid applications. Examples of existing monitoring systems are: Network Weather Service (NWS) (Wolski et al., 1999), Relational grid Monitoring Architecture (R-GMA) (Cooke et al., 2004), NetLogger (Gunter et al., 2000), etc.

Network Weather Service (NWS) is a popular distributed system for producing short term performance forecasts based on historical performance measurements. NWS is specific to monitor grid computing and network resources. It provides CPU percentage availability rather than CPU usage of a particular application. It supposes that nodes are available for the entire application. This conclusion is very limited to be used to achieve high throughput in a storage virtualization system like ViSaGe.

Relational grid Monitoring Architecture (R-GMA) is based on a relational model. It is an implementation of GGF Grid Monitoring Architecture (GMA). It is being used both for information about the grid and for application monitoring. R-GMA collected information was used only to find out about what services are available at any time.

Netlogger is a monitoring and a performance analysis tool for grids. It provides tracing services for distributed applications to detect bottlenecks. However, performance analysis is carried out in post mortem. In ViSaGe, we don't need a system for studying the application's state, but a system that analyzes the availability of grid resources.

The aforementioned monitoring systems are designed to produce and deliver measurements on grid resources and application efficiently. Nevertheless, none of these systems presents the whole of the pertinent characteristics for a virtualization system like ViSaGe. ViSaGe needs a monitoring system providing a full view of node's workload in order to choose better nodes according to its target (replicating data, distributing workload …) and during nodes workload variations. Therefore, our monitoring system, Admon, traces applications, and collects information about storage resources, grid resources and networks. It is considered as an intersection point between all the aforementioned monitoring systems. It should use its monitoring knowledge for choosing the accurate node. The choice will be according to a prediction model in order to place data, efficiently, during runtime execution.

3. ViSaGe Architecture

A grid consists of three levels: node level represents storage node and computing node, site level represents site's administrator of the grid, and grid's level represents grid's administrator. The components of our storage virtualization system are distributed on this architecture (Fig.1) which brought an additional service for improving storage quality, and the access to the data.

- At the node level, we have :
  1. The virtualization service (Vrt).
  2. The distributed file system service (Visagefs).
  3. The administration and monitoring service (Admon).
- At the upper levels (the site level and the grid level), we found the tools of the administration and monitoring service (Admon).

ViSaGe aggregates distributed storage resources. It proposes for providing a self adaptive storage management system. The ViSaGe storage virtualization layer allows simplifying the management of sharing storage resources: flexible and transparent access to the data, high performance (data replication, links configuration, etc).

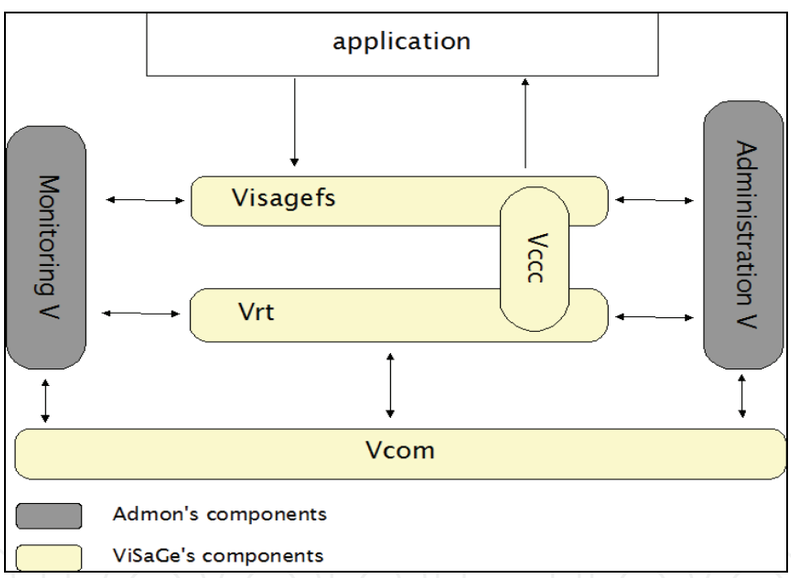

Fig. 1. ViSaGe's Architecture

The traditional application (like creation file, reading/writing data) uses the virtual storage resource through our file system, and thus reached data by means of virtualization layer. Furthermore, to handle the dynamic and the inherent nature of the grid, and to control the virtualization (storage resources attribution, virtual storage resources creation/destruction or modification), our storage virtualization system proposes the administration and the monitoring layer, Admon. Therefore, Admon's design must be strongly related to grid environment. Next, we present the hierarchical architecture and the fundamental concept of this system.

4. Administration and Monitoring Service

The Admon's components (Fig.2) are distributed throughout the grid. These components are:

1. An intelligent administration and monitoring agent, distributed among the storage and computing resources.
2. The administration and monitoring tools (server types), installed on site and grid controllers.

These components rely on a send/request protocol to monitor and manage the state of grid nodes (Foster et al., 2002).

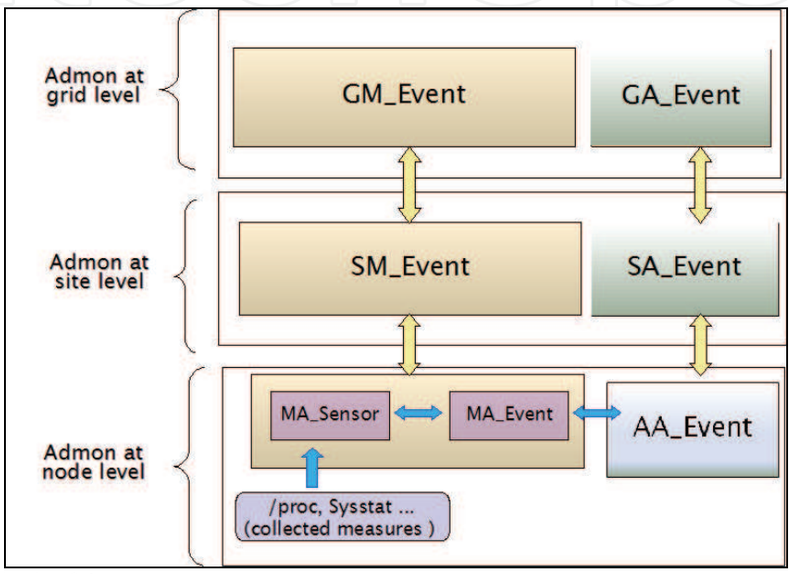

Fig. 2. Admon's Architecture

4.1 The administration components

The administration service manages the grid resources. It consists of (Fig.2): administration agent (AA_event), site's administration tool (SA_Event) and grid's administration tool (GA_event).

a) The administration agent:

In order to facilitate the deployment, the configuration and the update of the resources, an administration agent runs at the storage and computing elements. It constitutes a means of communication with the site's administration tool, and the other layers' interfaces of our storage virtualization system. It receives or sends events for instance when a node asks to be integrated into the site, its administration agent contacts the administration tool of its site to send its identity.

b) The site's administration tool:

It is a centralized tool for the site management, able to send to the grid's administration tool the information concerning the site that it manages. For example:

1. The storage elements configuration within the site, to allow the resource sharing.
2. The management of the available physical resources to be shared.
3. Sending and receiving messages from the grid's administration tool.

c) The grid's administration tool:
At the grid's level, the Admon's administration part is represented by the administration tool which runs in front of the grid owner. It manages grid user requests, and the information concerning the grid components in order to make decisions according to the nodes state.

4.2 The monitoring components

The Monitoring service collects information and sends notifications to the administrator. It consists of (Fig.2): monitoring agent (MA_sensor and MA_event), site's monitoring tool (SM_event) and grid's monitoring tool (GM_event).
a) The monitoring agent:
The intelligent monitoring agent is launched on all single machines where the processes of the application are executed. Distributed among the computing and storage elements, it allows collecting relevant information by the mean of MA_sensor and observing the behaviour of its node by the mean of MA_event. This information is represented by: CPU usage percent, Memory usage percent, Input/Output throughput to storage devices, Network throughput and event log.
The monitoring information is collected by the system command as "top". We usually poll at 100 ms intervals. Since the kernel events are not time-stamped, the data obtained this way represents all events in this interval. We use a simple statistical method to send the collected information when a significant change occurred. In order to no overload the grid, our agent monitoring is autonomous. It changes its interval of polling.
Event log analysis: for studying the node stability it is very important to collect the times-tamping of interest events to the logical volume of this node. This information will be sent to Admon grid's tool, by the mean of site's tool, in order to analyze the storage virtual state.
The adopted method for collecting relevant information is no disrupted of the running application, which means that our system is a non intrusive system.
b) The site's monitoring tool:
It is introduced as an intermediate monitoring process between the grid's monitoring tool and the monitoring agent. These monitoring tools are used to forward data from the monitoring agent to the grid's monitoring tool. The introspection mechanism, adopted by the monitoring agent to the node level, leverages the collected and optimized information of each observed node. Thus, it will have a global vision concerning the state of the components of the corresponding site.
c) The grid's monitoring tool:
The grid's monitoring tool incorporates the synthesized information by the monitoring tools of each site into the grid information system. This information describes the state of each site in the grid and represents the significant behaviours of the various components.
The synthesized information consists of:

- Site's state: overloaded or failing sites.
- Network's state: overloaded or broken links.

This collected information makes it possible to notify the grid's administration tool in order to make appropriate management decisions, and helps to have a global vision concerning the state of virtual storage resources created.

The collection of traces and logs creates a complete view of what the system is being ordered to be achieved. The next step should allow combining the system's knowledge with the statistically gathered data, for evaluating the stability state of the virtual storage resources.

5. Experimental results

In a data storage distributed system, the major concern is the data access performance. Many works focus on the data access performance has been concentrated on a static value and is not related to nodes load. However, whole system's performance is affected not only by the behavior of each single application, but also by the resulting execution of several different applications combined together. We have implemented Admon to study the service response time of ViSaGe's application influencing the system resources utilization. It traces applications and collects system resources' percentages (CPU, Disk, Network…) of utilization. The contribution of our work is to uncover which nodes are dedicated or not to ViSaGe. The Admon's performance metrics are represented by the CPU, Disk and Network. Admon sets the maximum value of performance metrics; constraint for achieving an experiment and making an adequate decision. This solution is used by Admon to identify grid dedicated node.

Here we present a use case study in order to show the Admon automatic instrumentation due to its decision. This decision improves data storage management and performance in our data storage system. Before on going on our experiment, we choose the CPU load (CPU < 70) like a constraint. This constraint is like a user demand to improve experiment performance.

5.1 Experimental setup

In our experimental, we choose the meshing AMIBE 3D. The European consortium EADS-CCR makes use of AMIBE drive electromagnetic simulations. We used grid platform, which gathered 4 sites distributed in France (IRIT, SEANODES, EADS-CCR, and CS-SI). For our preliminary experiments, we used 4 nodes (node1, node2, node3 and node4) distributed in these sites, where nodes are connected through gigabit Ethernet network.

The meshing full process encompasses three phases, each one being data dependent from its elder. First step is the 1D meshing which is started with a structural description. This job will run on a single node and generates one directory filled with one file per face to mesh. After completion, ViSaGe administration tools replicate data from current logical volume to all of the involved sites with at least one replica per site. Having 1D stage data replicated, we can launch the 2D stage on the reserved ViSaGe nodes for our application. We use a job scheduler (torque) to dispatch AMIBE 2D jobs on a face per face basis. On each run, a new directory is filled with an intermediate files. The final AMIBE 3D job is to collect and compound all those intermediate 2D files in one final set of 3D files. Thereby, although 2D data sets now span over a wide range of local storage resources. In AMIBE, before launching, an edge length must be determined. We used an AMIBE executable version where this value is set to 50. Modifying this value will modify performance result. However, in our experiment, we fix this value to 3. The reason from that is to show the administration and monitoring efficiency and utility by using AMIBE.

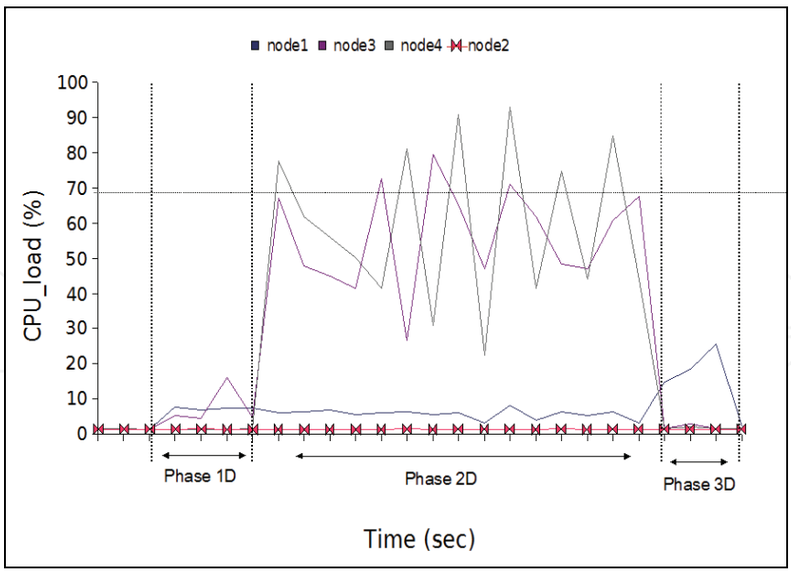

Fig. 3. nodes dedicated to ViSaGe

5.2 Results and discussion

We installed on node1 the Admon grid tools and site tools. On the three other nodes, we installed the Admon agents. The site monitoring tools know the nodes of its site where ViSaGe turns. For AMIBE, we used 3 nodes: one node for 1D or 3D and two other nodes for 2D phase. The CPU load worse case is fixed to 70 %.

We started by mounting the ViSaGe file system (Visagefs) on node1.

Before starting 2D stage, the Admon administration contacts the monitoring part to ask about computing nodes state reserved for the task. The grid scheduler has not all control over nodes tasks will be delegated. In our case study we have three nodes. The Admon administration chooses nodes according to their workload.

The first test (Fig.3) showed the CPU's nodes state in our experiment. This figure (Fig.3) shows the normal case of our test: all the nodes are dedicated to ViSaGe. We conclude that CPU usage percentage, in the 2D phase reaches highest percentage. Here, I must mention that the 2D phase is computing of 2D faces followed by storage of information in order to initialize the next step: we had CPU usage and I/O usage. That's mean, if the node2 is loaded node and it was choose in the 2D phase. We will have an experiment failure or increasing of service time.

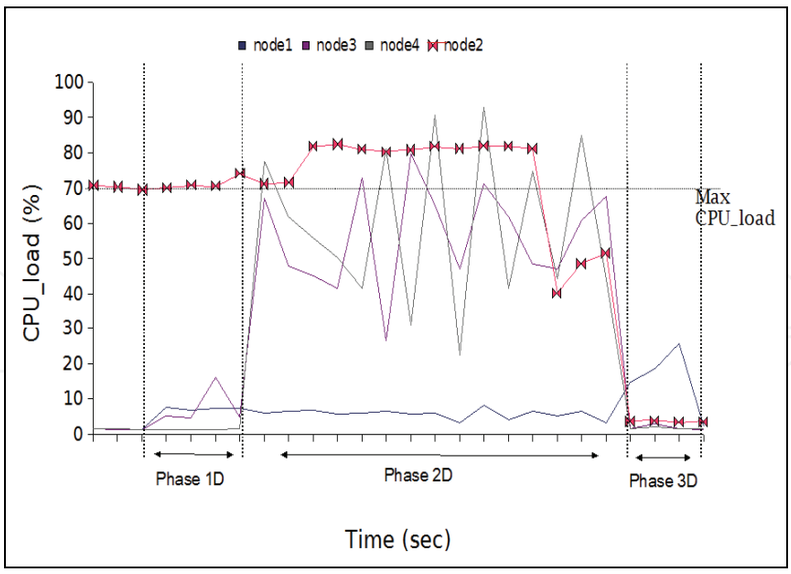

Fig. 4. node2 is dedicated to the grid

In the other hand, the second test, represented by the figure 4, shows from the workload of node2 (in terms of CPU load), that node2 is dedicated to the grid- that mean that this node is used by another applications (local or on the grid).
Before the launching the 2D phase, we launch on the node2 applications simulating the use of this node by a virtual user. The application of this user uses the nodes CPU, stores and sends requests to the disk (read or write data). The goal of this operation is to increase the load of the CPU in a progressive way in order to stabilize it on a maximum value. The monitoring agent collects the percentages of CPU node and sends this significant change to the monitoring tools. The monitoring tools, starting from the information collected by each monitoring agent, test the node's state. For ongoing on the 2D phase, Admon administration doesn't choose this node – "node2" for the 2D stage.
The other two nodes, (node3 and node4), aren't detected in term of dedicated node to the grid, Admon launches commands to mount the Visagefs. For adding these nodes in the same virtual space, Admon contacts the Vrt (the visage virtualization layer). The Vrt replicates data in order to launch the 2D phase.
We finalize our experiment by the 3D phase that synthesizes information on the node1.

6. Conclusion

In this paper, we have described a non-intrusive, scalable administration and monitoring system for high performance grid environment. It is a fundamental building block for achieving and analyzing the performance of the storage virtualization system in a huge and heterogeneous grid storage environment. It offers a very flexible and simple model that collects nodes state information and requirements needed by the other services of our virtualization storage system improving storage distributed data performance. It is based on a multi-tiered hierarchical architecture the start-up of the monitoring and administration system.

Future work will focus on developing interactive jobs with the storage virtualization system, in order to study the relation between different system resources (CPU, disk, networks) and tracing events, where Admon allows changing the storage distributed protocols (for instance, replication) during runtime execution. Therefore, this work will be difficult since we have many parameters that must be taken into consideration. However, this mechanism will allow generalizing I/O workload model in order to improve applications throughput during workload variations in grid environments.

7. References

Foster and al. (2002), *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, Intl J. High-Performance Computing Applications, vol. 15, no. 3, 2001, pp. 200-222, June 2002.

Porter, G. & Katz, R.H. (2006). Effective Web Service Load Balancing Through Statistical Monitoring. *Communication of the ACM, March 2006.*

Wolski, R; Spring, N.T. & Hayes, J. (1999). The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. *Future Generation Computing Systems, Metacomputing Issue*, 15(5-6): 157-768, Oct. 1999.

Gunter, D.; Tierney, B.; Crowley, B.; Holding, M. & Lee, J. (2000). NetLogger: A Toolkit for Distributed System Performance Analysis, *In Proceedings of the IEEE Mascots 2000*, Aug. 2000.

Cooke, A. et al. (2004). *The Relational Grid Monitoring Architecture: Mediating information about the grid*, 2004

Data Storage

Edited by Florin Balasa

ISBN 978-953-307-063-6

Hard cover, 226 pages

Publisher InTech

Published online 01, April, 2010

Published in print edition April, 2010

The book presents several advances in different research areas related to data storage, from the design of a hierarchical memory subsystem in embedded signal processing systems for data-intensive applications, through data representation in flash memories, data recording and retrieval in conventional optical data storage systems and the more recent holographic systems, to applications in medicine requiring massive image databases.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Salam Traboulsi (2010). Administration and Monitoring Service for Storage Virtualization in Grid Environments, Data Storage, Florin Balasa (Ed.), ISBN: 978-953-307-063-6, InTech, Available from: http://www.intechopen.com/books/data-storage/administration-and-monitoring-service-for-storage-virtualization-in-grid-environments

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821