Framework for Hypervisor in Grid

G.Senthil Kumar
Dept of EEE, Karunya Univeristy,Ciombatore

ABSTRACT

In the recent years, the role of virtualization has become vital in the grid computing community to bridge the gap between computation resource configuration and job requirements. The venture of virtualization in the grid computing environment has set another challenge to monitor and manage the virtual machines in the grid**.** To solve this problem of monitoring the virtual machines we propose a Virtual Information System which will provide the complete information about the virtual parameters of the machine. This information would be better useful for Grid management system to enhance the management of virtual resources.

Keywords: Grid, Virtual machine, Xen, GT4, MDS4, Virtual organization.

1. INTRODUCTION

Grid computing is a form of distributed computing whereby a cluster of loosely coupled computers, acting in concert to perform very large tasks, which exploits the underutilized resources and also provides balanced resource utilization along with parallel CPU utilization which leads to parallel processing. But sharing the resources among the clusters is not so easy, as each cluster has its unique configurations. Virtualization [11] technology has a long history in computer science. It allows the partitioning of computational resources (hardware, software, networking, etc.) into virtual entities, i.e. virtual machines. This technique abstracts the internal details of physical resources, thus provides the isolation and common interface for virtual elements to share the physical entities. To maximize the resources in grid computing, the use of the virtual machines has been employed. These virtual machines offer the ability to instantiate a new, independently configured guest environment on a physical resource; multiple, different such guest environments can be deployed on one resource at the same time. On providing convenience to the resource usage, these virtual elements can also increase the utilization of the resources. Furthermore, virtual machines make software easier to migrate, thus aiding application and system mobility, and can be used to consolidate the workloads of several under-utilized servers to fewer machines, perhaps a single machine (server consolidation). The virtualization technique simplifies the assignment of jobs to resources either by discovering a statically created environment or by expressing parameters of a dynamically created one, because the user's requirements specified in an abstract language may specify abstract features of the environment. Modern hypervisor implementations, such as Xen [13,19] and VMware [14] also provide outstanding isolation and enforcement properties, as well as excellent performance making the use of virtual machines cost-effective.

Virtualization of the grid resources has created an easier platform to work on. Also, the cost of deploying virtual clusters is cheaper than configuring a whole physical cluster for executing a particular job. These virtual machines that are created all over the grid has to be monitored like any other physical resource for scheduling, management purpose, reporting etc. So there is a need for a information system that monitors the virtual machines that exists in the grid, the Virtual Information System that we have proposed is capable of monitoring all the virtual machines in the grid which would be useful for the efficient management of resources.The rest of the paper is organized as follows.Section 2 describes the related work and the architecture of Virtual information system is described in section 3. The implementation details is explained in section 4 and finally we add our conclusion in section 5.

2. RELATED WORK

Various monitoring systems have been proposed in grid environment which monitors the resource information but there has been only a few proposal for monitoring the virtual information. Cluster monitoring systems like Ganglia[2], Hawkeye[15] provides the complete information about the physical resources and are commonly used in Grid systems. Ganglia provides scalable monitoring of distributed systems at various points in the architectural design space including large-scale clusters. In Ganglia , the Ganglia monitoring daemon (gmond) monitors the clusters using the listen/announce protocol. gmond runs in every node, and is organized as a collection of threads, each performing specific tasks, as a whole monitor the single cluster. It responds to the client requests via XML representation of its monitored data. The Ganglia Meta Daemon (gmetad) provides the federation of multiple clusters. The TCP connections between the gmetad daemons allow the aggregation of monitoring information between various clusters, and the application-specific data is published by the applications via the gmetrics, while command line programs provide programmatic access to the Ganglia features.

Hawkeye is a monitoring tool designed for grid and distributed applications which provide automated detection and timely alerts of problems. Hawkeye can be used for monitoring the system load, I/O, usage of the system, and also watching for run-away process. Hawkeye uses push data model and is built on Condor [15] technology. The Hawkeye Information Provider gathers Hawkeye data about Condor pool resources using the XML mapping of the GLUE schema and reports it to a GRAM4 service, which publishes it as resource properties The information includes basic host data (name, ID), processor information, memory size, OS name and version, file system, data processor load data, other basic Condor host data. This information provider is included in the GT4 toolkit and used for reporting GLUE Computing Element (CE) information.

MonALISA is a robust distributed monitoring service, which uses an architecture designed to as one of autonomous multi-threaded, self-describing agent-based subsystems which are registered as dynamic services, and are able to collaborate and cooperate in performing a wide range of information gathering and processing tasks. It provides the ability to invest the system with increasing degrees of intelligence, to reduce complexity and make global systems manageable in real time The system is designed to easily integrate existing monitoring tools and procedures and to provide this information in a dynamic, customized, self describing way to any other services or clients.

The literature ANG[10] proposes a information system which runs a grid gateway at all sites, has a number of virtual machines being used to deploy the hosts running other then GT4.The MDS4 Index Service uses resource information provided by the MIP (Modular Information Provider) and uses PBS(Portable Batch System) as their local resource manager.

MIP is a modular architecture which gives the administrator to perform the remote queries to remote information source like queuing system that are not running on the GT4 host. It uses XML implementation of GLUE1.2 and provides generic interfaces to all middlewares. MIP is integrated to MDS4 through RPProvider framework. However these functionalities provided information system to every middlewares, It does not provides information about the virtual machine and its neither supports software package information by default.

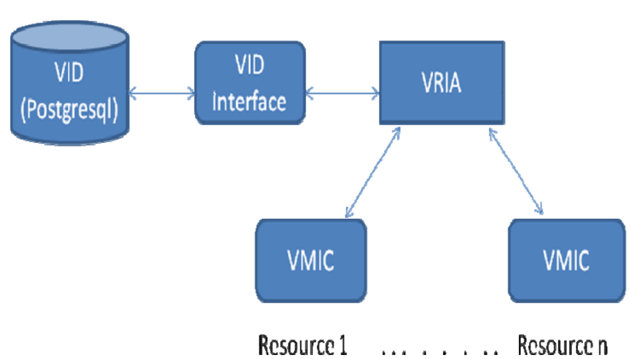
Another proposal which is more in line with our implementation is Magrathea [3]. This project aims at monitoring the virtual machine informations. Monitoring scheme employed in Magrathea is similar to Ganglia. It deviates from Ganglia from the point of implementing the master daemon. The master daemon manages the virtual machines and also implements a scheduler to schedule the jobs accordingly, as which virtual machine suits the job. The slave daemon in each virtual machine monitors the virtual machine, runs the job when submitted to it and intercepts the results to the master daemon, which then recomputes status of all virtual machines. The status cache is a primary information source, which maintains the information about the virtual machines to increase the performance and scalability. Although, this system provide the architecture for scheduling the jobs across virtual machines, it fails to address the common scheme for information management. We propose a Virtual Information System to monitor the virtual machines in individual nodes and across cluster nodes in a grid. The virtual machine informations obtained are very useful for the management of virtual resources in a grid.

3. ARCHITECTURE OF VIRTUAL INFORMATION SYSTEM

VIS is based on a hierarchical design, aims for the better management of grid resources. The Virtual Information System has 4 main components viz., Virtual Resource Information Aggregator (VRIA), Virtual Machine Information Collector (VMIC), Virtual Information Database (VID), VID Interface.

VRIA:

Virtual Machine Information Collector monitors the complete information about the virtual machines available in the resource and regularly updates the information to VRIA. VRIA collects the information obtained from VMIC and process the data to VID interface for the storage. VID interface acts as a bridge for communication between the database and VRIA. The information thus obtained by VRIA is transformed and stored in the VID. The VID we included in our architecture is of open source database, PostgreSQL. The virtual information system (VIS) can be configured in a centralized/distributed fashion depending upon the requirement.



.Fig 1. Architecture of Virtual Information System

VMIC:

VMIC is designed in such a way that the information can also be monitored for cluster resources also. It has various subcomponents for obtaining the virtual machine information. VMIC obtains the virtual machine information using its major component Host Monitor. The architecture of Host Monitor is represented in figure 2.

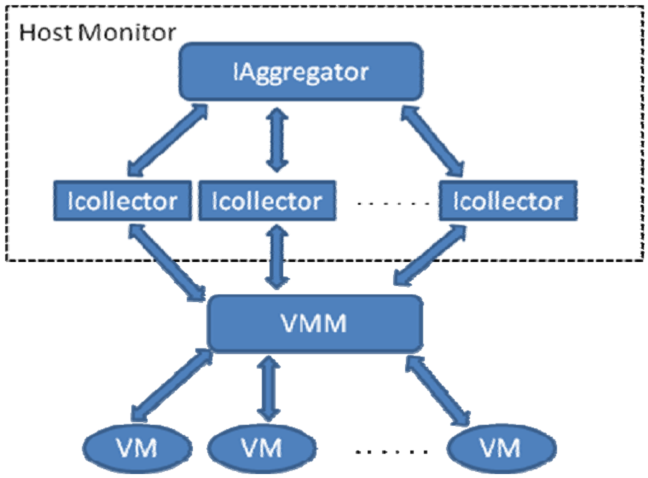


Fig 2. Architecture of Host Monitor

The Host Monitor is configured per each compute node of a cluster. The host monitor component is responsible monitoring the virtual machines information and sending the monitored data to VMIC. HM has two main components viz., Information Collector (ICollector) and Information Aggregator (IAggregator). ICollector monitors all the virtual machine available using the Hypervisor APIS and operating system calls. It also collects the information in accordance with the virtual information schema proposed. The virtual parameters obtained by the Host monitor viz**.,** Universally Unique Identifier provides opaque reference to VM, Power state, Caption of VM, Description about VM, Host of VM currently Resident on, Obsolute memory size (Minh/Max), Vibrant memory size (Min/Max), Hypervisor version, Number of virtual CPUs running in host, Virtual CPUs utilization, etc. The API used to repossess these information in XEN is represented in Table 1. IAggregator obtains the information from the ICollector and periodically updates to the VMIC.

NARRATION	XEN
VM UUID	Get_uuid()
VM Power State	Get_power_state()
VM Name	Get_name_label()
VM Description	Get_name_description()
VM Resident	Get_resident_on()
Static Memory Maximum	Get_memory_static_max()
Static Memory Minimum	Get_memory_static_min()
Dynamic Memory Maximium	Get_memory_dynamic_max()

Dynamic Memory Minimum	Get_memory_dynamic_min()
Virtual CPU Number	Get_VCPUs_number()
Virtual CPU Utilization	Get_VCPUs_utilisation()

Table 1. XEN API

4. IMPLEMENTATION

VIS system model is completely realized using Java. For testing the virtual machine information we have used the Hypervisor XEN-3.0.4 in Scientific Linux . The related Hypervisor API has been used along with Java for the information retrieval about the virtual machines. Information extracted using XEN-API is stored in database like PostgreSQL. In our System Virtual information about the VMs is integerated to the Globus Toolkit - MDS4 through the ResourceProvider. This cause changes in default index services of Globus by adding the extracted VMs information. Using WSRF query the virtual information added to Index services of GT4 is viewed it is given below.


```
<info>
<time> Thu Mar 19 18:34:59 IST 2009 </time>
<host> g12.grid </host>
<vminfo>

XEN PRESENT
<COLLECTED_INFORMATION_FROM_VIRTUAL_MACHINE>00000000-0000-0000-
0000-000000000000 </COLLECTED_INFORMATION_FROM_VIRTUAL_MACHINE>

<UUID>00000000-0000-0000-0000-000000000000</UUID>
<State>Running</State>
<Name>Domain-0</Name>
<Description>null</Description>
<Resident>177a1f3a-0dab-cfac-97bb-02e6cd50255e </Resident>
<StaticmemoryMax>256MB</StaticmemoryMax>
<StaticmemoryMin>224MB</StaticmemoryMin>
<DynamicmemoryMax>0MB</DynamicmemoryMax>
<DynamicmemoryMin>224MB</DynamicmemoryMin>
<VCPUNo>1</VCPUNo>
<VCPUUtilize>123.720400485</VCPUUtilize>
<VMPlatformSerial>100_61760</VMPlatformSerial>
<VMPlatformLocaltime>18.19.531</VMPlatformLocaltime>
<VMPciBus>DataBus</VMPciBus>
</vminfo>
</info>
```

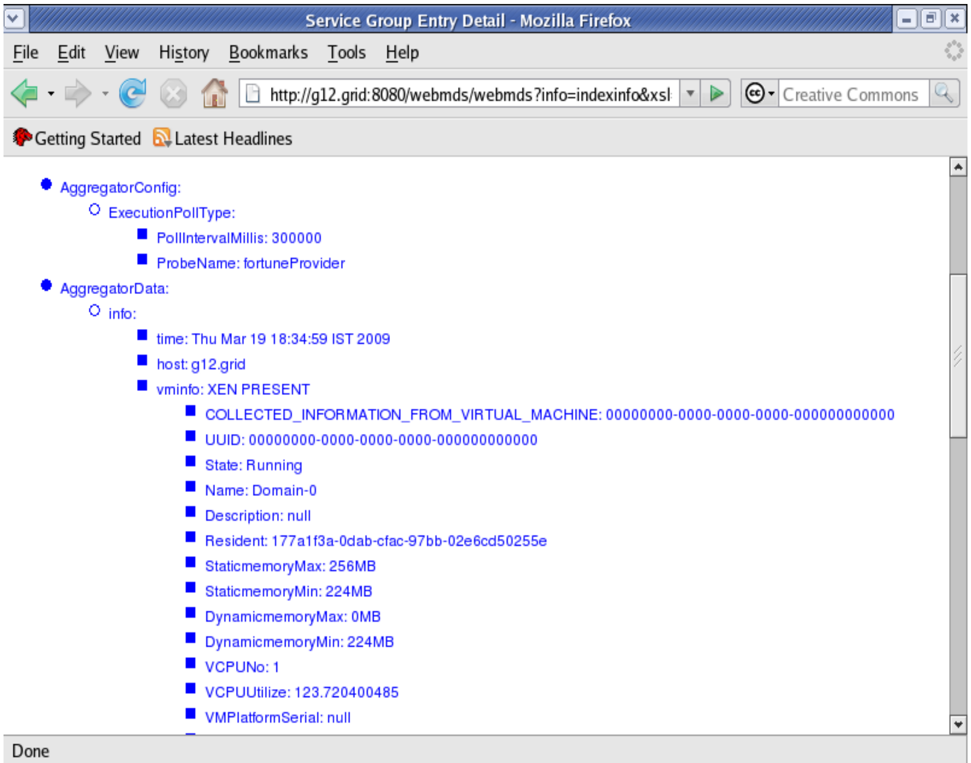


Fig 3. MDS4 With XEN Hypervisor Information

5. CONCLUSION

In this paper we propose a Information system for grid environment. The architecture which we proposed is of hierarchical fashion to monitor cluster grid resources.VIS tracks the complete information about the virtual resources available in the grid. This information could be useful for any grid management system to access the virtual resources and monitor them. VID interface which act as ResourceProvider is integrated to MDS4 and leads to effective usage of resource in Virtual Organization.

REFERENCES

[1] Jiri Denemark, Michał Jankowski, Paweł Wolniewicz ,Ludek Matyska,Norbert Meyer, "Virtual Environments - Framework for Virtualized Resource Access in the Grid", In CoreGRID Technical Report Number TR-0066 , December 27, 2006

[2] Matthew L. Massie , Brent N. Chun , David E. Culler ,"The Ganglia distributed monitoring system:design, implementation, and experience", Lecture notes in University of California, Berkeley, Computer Science Division, Berkerley, CA 94720-1776 , 11 February 2003

[3] Jiri Denemark et al., "Virtualizing META Center Resources Using Magrathea", In CESNET technical report number25/2007, 2007

[4] Marcel Kunze and Lizhe Wang, "Information services of virtual Machines pools in grid computing ", Lecture Notes in Computer Science Volume 4854/2008 Euro-Par 2007 Workshops: Parallel Processing ISBN978-3-540-78472-2 Pages174-184, March 08, 2008

[5] Ghazala Shaheen et al., "Grid Visualizer: A Monitoring Tool for Grid Environment", In Proceedings of the 16th International Workshop on Database and Expert Systems Applications pp297-301, August 2005.

[6] Andrea Ceccanti and Fabio Panzieri et al., "Content-Based Monitoring in Grid Environments", Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterpriss(WETICE'04) pp255-259, 2004.

[7] Resource Schema Description and Definition for GridX1 Services Project", Is the Schema Defintion Document (SDD) in HEP Computational Grid Services Version 1.2 ,June 2006

[8] Thomas Naughton et al., "Dynamic Adaptation using Xen -Thoughts and Ideas on Loadable Hypervisor Modules". In proceedings of the 1st workshop on system-level virtualization for high Performance computing (HPCVirt'07) held in conjunction with the ACM EuroSys'07 Lisbon ,March 2007

[9] Ann Chervenak, Jennifer M.Schopf ., "Monitoring the Earth System Grid with MDS4",e-science,p.69, In second IEEE international conference on e-science and grid computing(e-Science'06); 2006

[10] Gerson Galang, Paul Coddington , "Experiences in Deploying an MDS4 Grid Information System on the Australian National Grid", In APAC conference on Advanced computing ,grid applications and eResearch , Australia, October 2007.

[11] J. E. Smith and Ravi Nair, "An Overview of Virtual Machines Architectures", Excerpt from "Virtual Machines: Architectures, Implementations and Applications," published by Morgan Kaufmann Publishers, 2004, copyright 2004 by Elsevier Science (USA), November 1, 2003

[12] Xuehai Zhang, Katarzyna Keahey, Ian Foster, Timothy Freeman, "Virtual Cluster Workspaces for Grid Applications", ANL Tech Report ANL/MCS-P1246-0405.

[13] Xen: Enterprise Grade Open Source Virtualization A XenSource White Paper V06012006

[14] Xen Management API, Release notes on API Version 1.0.0, 27th April 2007

[15] Thomas Naughton, Geoffroy Vallee, Stephen L. Scott, "Dynamic Adaptation using Xen -Thoughts and Ideas on Loadable Hypervisor Modules", In First Workshop on System-level Virtualization for High Performance Computing (HPCVirt 2007), Lisbon, Portugal, March 20, 2007