University of Victoria
Faculty of Engineering
Fall 2008 Work Term Report

Monitoring Grid Virtual Machine deployments

Department of Physics
University of Victoria
Victoria, BC

Michael Paterson
0031209
Work Term 2
Software Engineering
mhp@uvic.ca

January 8, 2009

In partial fulfillment of the requirements of the
Bachelor of Software Engineering Degree

Supervisor's Approval: To be completed by Co-op Employer

I approve the release of this report to the University of Victoria for evaluation purposes only.
The report is to be considered (**select one**): ☐ NOT CONFIDENTIAL ☐ CONFIDENTIAL

Signature: _____ Position: _____ Date: _____

Name (print): _____ E-Mail: _____ Fax #: _____

If a report is deemed CONFIDENTIAL, a non-disclosure form signed by an evaluator will be faxed to the employer. The report will be destroyed following evaluation. If the report is NOT CONFIDENTIAL, it will be returned to the student following evaluation.

Contents

1 Report Specification **4**
- 1.1 Audience . . . 4
- 1.2 Prerequisites . . . 4
- 1.3 Purpose . . . 4

2 Introduction **4**
- 2.1 Grid Computing Systems . . . 4
- 2.2 Globus Toolkit . . . 4
- 2.3 Nimbus . . . 5
- 2.4 System and Network Monitoring . . . 5

3 Nimbus Nagios Integration Strategy **5**
- 3.1 Nagios . . . 5
- 3.2 Monitoring and Discovery System - MDS . . . 5
- 3.3 Monitoring Nimbus . . . 5
  - 3.3.1 Monitoring with nagios . . . 5
  - 3.3.2 Monitoring with MDS . . . 5
- 3.4 Monitoring VMs . . . 6

4 Implementation **6**
- 4.1 Monitoring Nimbus . . . 6
  - 4.1.1 Monitoring Nimbus using nagios . . . 6
  - 4.1.2 Simple Plug-ins to Monitor a Metric . . . 6
  - 4.1.3 The Nimbus Consistent Plug-in . . . 7
  - 4.1.4 Monitoring Nimbus using the MDS . . . 7
- 4.2 Monitoring VMs . . . 7

5 Conclusion **9**

6 Future Work **10**

7 Acknowledgments **10**

8 Glossary **11**

List of Figures

1 Nimbus plug-ins in Nagios. 6
2 The front display of Web MDS showing the Nimbus aggregator. 7
3 Details of the Nimbus Aggregator Plug-in. 8
4 Sequence diagram showing Nimbus Nagios interaction with Client. 8
5 Nimbus hosts being monitored by Nagios. 9
6 Nimbus hosts with services being monitored by Nagios. 9

Monitoring Grid Virtual Machine deployments

Michael Paterson
mhp@uvic.ca

January 8, 2009

Abstract

Nimbus allows the creation of virtual workspaces on grid systems, making computing resources more accessible to users who require it, while still allowing the grid to accept other jobs in the usual manner. In order to monitor both the Nimbus service and the virtual workspaces it creates, plug-ins were created for both the Nagios monitoring platform, and for Globus' Monitoring and Discovery Service(MDS). Nimbus was then modified to communicate with Nagios in order to provide monitoring capabilities for virtual workspaces.

1 Report Specification

1.1 Audience

This report is intended for members of the High Energy Physics Grid Computing Group at University of Victoria, in addition to anyone working to monitor Nimbus created virtual machines.

1.2 Prerequisites

A general understanding of virtualization, distributed computing, and clusters is assumed.

1.3 Purpose

The report provides overviews of Nimbus, and Nagios. With a discussion of the challenges associated with their integration to provide monitoring capabilities of virtual workspaces.

2 Introduction

2.1 Grid Computing Systems

Grid systems are often composed of heterogeneous resources, with each portion of the grid under control of a different organization. Grid systems are often used for solving scientific or technical problems where large amounts of data needs to be processed.

2.2 Globus Toolkit

The de facto standard for building grid solutions. The Globus Toolkit [1] is middle ware that contains components for creating and managing aspects of a grid system.

2.3 Nimbus

Formerly known as Globus Virtual Workspaces (GVW), Nimbus [2] is built on the Globus Toolkit and uses a Virtual Machine Manager (VMM) in order to create and manage Virtual Machines (VMs) on grids so users have a known computing environment to use without having to deal with different hardware systems that may be present on the grid. Nimbus can operate in two modes, resource pool mode, in which resources are requested directly from the Nimbus service, or pilot mode, where Nimbus works with a Local Resource Management System (LRMS) so that workspace requests as well as normal computing jobs can be accommodated by the system. The ability to viritualize operating systems on a grid can greatly increase the availability of computing resources to users.

2.4 System and Network Monitoring

A common administration task for any computer system or network is monitoring the health of the system and services it provides. The nature of grids creates challenges in monitoring their health, with the inclusion of virtual workspaces there has been a lack of tools available to monitor these resources. The goal of my work term was to help address the lack of monitoring options available for virtual workspaces.

3 Nimbus Nagios Integration Strategy

In order to monitor Nimbus and workspaces there were two obvious choices, a well known and widely used monitoring system called Nagios [3], and Globus' own monitoring service MDS [4]. Both of these options were investigated for use in developing monitoring solutions. It is also possible to use both these systems by parsing results from Nagios plug-ins and posting them to the MDS as shown by the Nagios Information Provider [5].

3.1 Nagios

Nagios is a widely used platform for system, network, and application monitoring. It is highly configurable and has a plug-in architecture for customizing monitoring needs. The use of Nagios plug-ins allows the creation of monitoring tools tailored to specific applications and services. Information about the Nimbus service was gathered and monitored using Nagios plug-ins, also Nimbus was modified to interact with Nagios to provide monitoring services to virtual workspaces.

3.2 Monitoring and Discovery System - MDS

Globus has a Monitoring and Discovery System (MDS) where information about services can be gathered using a Web Services Resource Framework (WSRF). The MDS provides information about services on the grid and their status. This information can be queried using WSRF. Information about the current workspaces and the user who created them was gathered from the Nimbus service and published to the MDS.

3.3 Monitoring Nimbus

3.3.1 Monitoring with nagios

Plug-ins following the Nagios plug-in standard were created to monitor various aspects of the Nimbus service. The aim of these plug ins was to show that it was possible to monitor the service and provide information that would assist in tracking down what problems were occurring without having to resort to log files.

3.3.2 Monitoring with MDS

A plug-in was developed to gather information about the current Nimbus users and post that information to the MDS. A request was also made to add a method to the Nimbus service API to obtain this information, but due to time constraints only preliminary testing was done using the new API method.

Nimbus_Consistent	WARNING	10-17-2008 11:13:20	0d 0h 0m 21s	1/4	STATUS WARNING: mismatch on: gsn-wn1 IDs: 12 gsn-wn3 IDs: 10
Nimbus_VM_Running	OK	10-17-2008 11:13:10	0d 0h 40m 31s	1/4	VMRUN_STATUS: 5 workspaces running
Nimbus_VM_Slots	WARNING	10-17-2008 11:11:16	0d 0h 5m 25s	4/4	VMSLOT_STATUS: WARNING 1 slots remaining

Figure 1: Nimbus plug-ins in Nagios.

3.4 Monitoring VMs

Now that it was possible to monitor the Nimbus service, we began investigating ways to monitor the VMs directly. Modifications were made to the Nimbus service to allow it to interact with Nagios in order to enable and disable monitoring on VMs. These VMs could then contain their own Nagios plug-ins to be monitored if desired.

4 Implementation

4.1 Monitoring Nimbus

Nimbus keeps a persistence database that stores information about the state of workspaces, resources, networks,and other details about the state of the service. In order to monitor Nimbus, querying the persistence database was the most straightforward solution. The Nimbus persistence database is where Nimbus stores all information regarding its available and used resources, running workspaces, and the state these resources are in. The default database used by Nimbus is an embedded Apache Derby database [6]. A feature of this database is that it using file locks to prevent synchronization problems. This posed an obstacle since the Nimbus service does not release the database as it was never intended by the developers to be a source of information about the service. We worked around this by taking a copy of the database and removing the locks in order to perform queries, a proposal was made to the Nimbus developers to include the Derby Network Server in Nimbus that would allow for external read only access to the persistence database. The inclusion of the Derby Network Server would remove the need for making a copy and improve performance.

4.1.1 Monitoring Nimbus using nagios

Nagios plug-ins can be Perl, python, Unix like shell scripts, and other languages. So long as the Nagios plug-in specification is adhered to and the plug in supports the expected inputs and outputs anything that goes on in between is up to the plug-in developer. There is a timing requirement that the plug-in can complete execution during Nagios' 10 second window before it determines the plug-in has timed out.

To show the possibilities of monitoring the Nimbus service two simple plug-ins were created and one additional more involved plug in.

All of the plug-ins follow the same procedure to work.

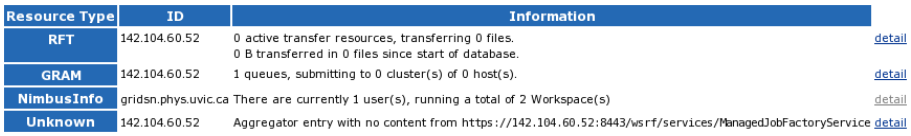
1. A copy of the Nimbus Persistence Database is made and the locks are removed to enable access.
2. A query to the database is made to gather information of interest.
3. The plug-in interprets the query results and reports status and information back to Nagios.

Figure 1 shows the plug-ins status as shown in Nagios. If the plug-ins are not actively being checked, the output from the last check is displayed.

4.1.2 Simple Plug-ins to Monitor a Metric

Two simple plug-ins were developed for monitoring a metric of Nimbus.

NimbusWSRunning Provides of a count of all running workspaces



Resource Type	ID	Information	
RFT	142.104.60.52	0 active transfer resources, transferring 0 files. 0 B transferred in 0 files since start of database.	detail
GRAM	142.104.60.52	1 queues, submitting to 0 cluster(s) of 0 host(s).	detail
NimbusInfo	gridsn.phys.uvic.ca	There are currently 1 user(s), running a total of 2 Workspace(s)	detail
Unknown	142.104.60.52	Aggregator entry with no content from https://142.104.60.52:8443/wsrf/services/ManagedJobFactoryService	detail

Figure 2: The front display of Web MDS showing the Nimbus aggregator.

NimbusWSSlots Monitors the default public network pool and reports how many remaining entries are available for workspace creation.

Nimbus Creates workspaces using network pools, if Nimbus runs out of entries to use in the requested pool the error returned to the user was fairly verbose and could be difficult to determine that this was being caused by a lack of available entries.
These two plug-ins illustrate monitoring a fairly simple metric of the Nimbus service. As other metrics of the service become of interest to administrators additional plug-ins could be developed to monitor them.

4.1.3 The Nimbus Consistent Plug-in

After making the two earlier plug-ins, a plug-in was developed that would not just look at and report values, but interpret them and compare retrieved information against Nimbus. The result was the Nimbus Consistent plug-in.

NimbusConsistent Check for inconsistencies between what Nimbus thinks is running and what is actually running.

During the initial testing of the Nimbus service it was found that occasionally if a workspace was started with a bad configuration script, or if the VM was unexpectedly terminated, the workspace would not know the VM was gone or would persist after the workspaces intended termination time.

This error was difficult to see as it was often unclear which workspace was still running when it should not be and where the VM should be located. Using data queried from the persistence database a list of VMs that should be running based on their shutdown times was created by the plug-in. The first version of the plug-in then used ssh to check the xen manager listing of running domains and parsed the results. If the correct VMs were found the service was working fine, if an expected VM was missing from the list the plug-in reported the workspace ID and the node where the VM should be running. This made it possible to clean up the workspaces and investigate nodes that were failing to start VMs. The second iteration of this plug-in replaced the ssh calls with use of the libvirt [7] library in order to avoid parsing based issues. This required the libvirtd daemon to be running on the nodes but provides a consistent and programmatic way of obtaining the information instead of parsing output that may change and more prone to error.

4.1.4 Monitoring Nimbus using the MDS

To publish data about Nimbus to the MDS an aggregator plug-in was created that queries the persistence database then formats the results into XML. This aggregator operates in the same manner as the Nagios plug-ins. An XSLT was added to have some summary information able to be displayed on the front page of Web MDS as seen in Figure 2. The detailed output can be seen in Figure 3.

A second iteration of this plug in is being investigated to use a new method in the Nimbus API to acquire the same information without having to query the database and parse output. Preliminary testing was done on the method but no implementation was done due to time constraints.

4.2 Monitoring VMs

In order to make Nimbus and Nagios communicate in some way, the Nagios external command file was used. This file is checked by Nagios periodically and the commands within are executed. Nimbus was modified

- AggregatorData:
  - creatorsInfo:
    - creatorName: /C=CA/O=Grid/OU=phys.uvic.ca/CN=Michael Paterson
    - workspaces:
      - workspaceInfo:
        - id: 20
        - node: gsn-wn3
        - name: http://example1/localhost/image
        - image: file://ttylinux-xen
        - IPAddress: 192.168.42.111
      - workspaceInfo:
        - id: 19
        - node: gsn-wn1
        - name: http://example1/localhost/image
        - image: file://ttylinux-xen
        - IPAddress: 192.168.42.113

Figure 3: Details of the Nimbus Aggregator Plug-in.

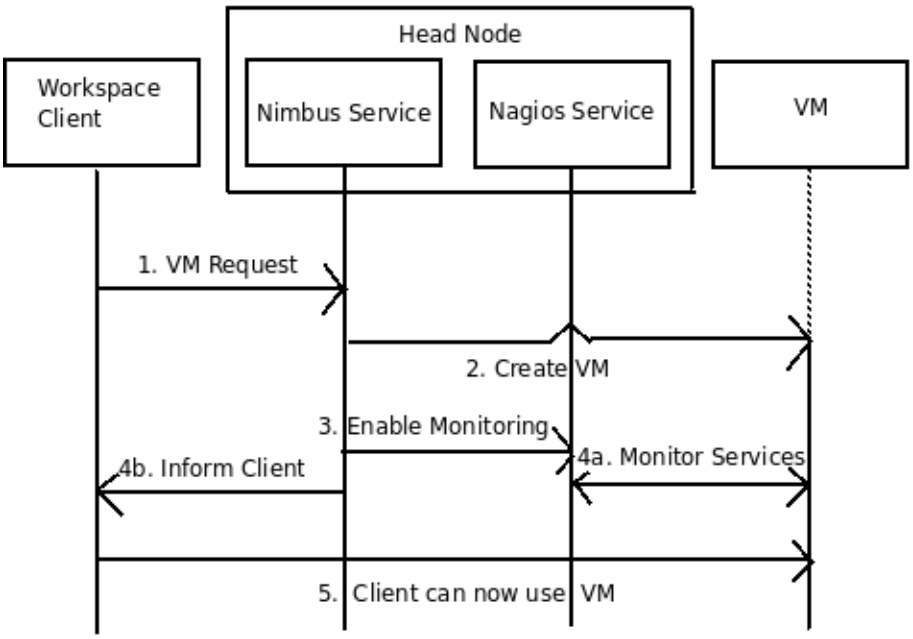


Figure 4: Sequence diagram showing Nimbus Nagios interaction with Client.

Host	Status	Last Check	Duration	Status Information
gsn-wn1	UP	01-07-2009 21:34:53	58d 10h 16m 35s	PING OK - Packet loss = 0%, RTA = 0.21 ms
gsn-wn2	UP	01-07-2009 21:35:13	47d 7h 57m 30s	PING OK - Packet loss = 0%, RTA = 0.23 ms
gsn-wn3	UP	01-07-2009 21:35:13	58d 10h 15m 35s	PING OK - Packet loss = 0%, RTA = 2.50 ms
localhost	UP	01-07-2009 21:37:33	168d 7h 31m 21s	PING OK - Packet loss = 0%, RTA = 0.06 ms
pub01	UP	12-17-2008 15:52:23	21d 5h 46m 35s	PING OK - Packet loss = 0%, RTA = 0.30 ms
pub02	DOWN	12-17-2008 14:02:33	34d 4h 55m 58s	CRITICAL - Host Unreachable (192.168.42.112)
pub03	DOWN	12-17-2008 14:01:53	22d 5h 6m 55s	(Host Check Timed Out)
pub04	DOWN	12-04-2008 16:24:36	34d 5h 14m 28s	CRITICAL - Host Unreachable (192.168.42.114)
pub05	DOWN	12-11-2008 16:11:34	27d 5h 27m 28s	CRITICAL - Host Unreachable (192.168.42.115)
pub06	PENDING	N/A	26d 8h 16m 55s+	Host is not scheduled to be checked...

Figure 5: Nimbus hosts being monitored by Nagios.

pub01	Test VM Plugin	OK	12-17-2008 15:54:41	21d 5h 45m 26s	1/4	successfuly excuted /home/nagios/test_plugin.sh on image!
pub02	Test VM Plugin	UNKNOWN	12-17-2008 14:02:14	26d 9h 28m 10s	1/4	Remote command execution failed: ssh: connect to host 192.168.42.112 port 22: No route to host

Figure 6: Nimbus hosts with services being monitored by Nagios.

with additional options and knowledge of this file with some specific commands to use. The interaction between Nimbus and Nagios is illustrated in Figure 4

1. User makes a request for VM to Nimbus.

2. Nimbus checks for available resources and creates VM

3. Nimbus notifies Nagios that the VM has started.

4a. Nagios begins monitoring the VM

4b. Nimbus informs User that VM has started

5. User can now connect to VM

Nagios will continue to monitor the VM until the shutdown time is reached, at which point Nimbus will notify Nagios to stop monitoring on that VM.

There are configuration options to monitor hosts only see Figure 5, or both hosts and services see Figure 6. The red X marks and pass down arrows indicate that hosts and service checks are currently disabled, so the status shown is from the last run check.

The interaction between Nimbus and Nagios works because the way the Nimbus network pool resources are configured is similar to how Nagios defines host machines. Both use a host name matched with an IP address, these are the two pieces of information that are passed from Nimbus to Nagios to enable and disable monitoring of the VM.

5 Conclusion

With the addition of monitoring options to Nimbus, Nagios is able to monitor virtual workspaces like any other machine. Using Nagios or MDS, it is possible to monitor the Nimbus service to aid in administrative tasks and trouble shooting. These features should help mitigate the lack of monitoring solutions for virtual workspaces and provide a basis for further work in this area.

6 Future Work

One of the current drawbacks of the plug-ins is the need to copy the database, A feature request to implement the Derby Network Server into Nimbus has been made [8].

Additional work with the getGlobalAll method, only preliminary testing has been done to date. The use of this method could replace some database queries, performance analysis of using the method versus querying should be done. The lead Nimbus developer has stated that the GlobalGetAll method is fairly process intensive.

Modeled after the Nagios Information Provider, the current plug ins could be adapted to be read into MDS to make that information available there as well as to Nagios.

Additional options to control the granularity of monitoring on VM images could be included.

Results from the monitoring plug-ins could be read back into Nimbus from the MDS and used in some way.

7 Acknowledgments

I would like to thank Dr. Randall Sobie for this work term opportunity. Thanks to Ian Gable for guidance on this work term. Additional thanks Ron Desmarais, Patrick Armstrong, and Nimbus developers: Tim Freeman and Kate Keahey at Argonne National Laboratory for technical help and advice.

8 Glossary

GT4 Globus Toolkit 4. The *de facto* grid middle ware.

GVW Globus Virtual Workspaces. Now known as Nimbus.

LRMS Local Resource Management System. Manages jobs on local clusters.

MDS Monitoring and Discovery System. Information services component of the Globus Toolkit and provides information about the available resources on the Grid and their status.

VM Virtual Machine, an instance of a machine(computer) running in software.

VMM Virtual Machine Monitor, used for managing virtual machines.

Xen Open-source VMM used by Nimbus.

XSLT eXtensible Stylesheet Language Transform, for transforming XML between XML documents.

References

[1] Globus Toolkit http://www.globus.org/toolkit/

[2] Nimbus http://workspace.globus.org/index.html

[3] Nagios http://www.nagios.org/

[4] MDS http://www.globus.org/toolkit/mds/

[5] Nagios Information Provider http://www.globus.org/toolkit/docs/4.2/4.2.1/info/providers/nagios/

[6] Apache Derby http://db.apache.org/derby/

[7] libvirt - Virtualization API http://libvirt.org/

[8] Bugzilla request to add Derby Network Server support http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=6516