University of Victoria
Faculty of Engineering
Spring 2008 Work Term Report

Integration of Globus Virtual Workspaces and LRMS using the Workspace Pilot

Department of Physics
University of Victoria
Victoria, BC

David Bartle
0429068
Work Term 3
Computer Science/Mathematics
dwbartle@uvic.ca

September 8, 2008

In partial fulfillment of the requirements of the
Bachelor of Science 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

List of Figures

Integration of Globus Virtual Workspaces and LRMS using the Workspace Pilot

David Bartle
dwbartle@uvic.ca

September 8, 2008

Abstract

The Workspace Pilot provides integration of Globus Virtual Workspaces (GVW) and LRMSs. Integration allows both systems to share cluster nodes, providing increased flexibility and resource utilization, while maintaining the autonomy of the LRMS.

GVW TP1.3.1 and TORQUE 2.0.0p11 were successfully installed on a test cluster and GVW was configured to use the Workspace Pilot. Trials of continuous prepropagated virtual workspace deployments were conducted to analyze fluctuations in node memory, to verify the functionality of the Xen balloon driver and independent management of the LRMS. Test results indicate that the Workspace Pilot is promising, although additional progress needs to be made to improve reliability.

1 Report Specification

1.1 Audience

This report is intended for future Coop students working in Grid computing.

1.2 Prerequisites

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

1.3 Purpose

The report provides overviews of Globus Virtual Workspaces and TORQUE and challenges associated with their integration.

2 Introduction

In High Energy Physics, large data sets and extensive calculations require immense resources to produce results. Computational grids are employed to fulfil these requirements.

Grids are composed of heterogeneous resources, spanning organizational, institutional, and geographical boundaries. Heterogeneous resources introduce complexity as they provide an inconsistent and unstable programming environment. As clusters within individual organizations are managed independently, it is cumbersome and time-consuming to maintain conformance to specific software prerequisites.

Globus Virtual Workspaces (GVW) [1], a component for the *de facto* grid middleware, Globus Toolkit (GT) [2], mitigates the development challenges of heterogeneous resources by using virtual machines. Instead of targeting specific hardware, programs can be written for any OS and configuration that can be launched as a virtual machine; this provides a software development environment that is independent of the hardware on which it runs, and allows the developer to select a preferred OS. Using virtual machines also allows legacy software to be used on new hardware. The Virtual Machine Monitor (VMM), not the programmer or system administrator, is tasked with environment uniformity; this drastically reduces the amount of time spent on cross-compatibility and cluster administration, while increasing the availability of compatible resources.

Local Resource Management Systems (LRMSs) provide submission, scheduling, and dispatch services for local clusters. Until release TP1.3.1, GVW had no capacity to integrate with existing LRMSs. Each cluster node was reserved exclusively for the LRMS or GVW. Unoccupied nodes in one system could not be utilized to assist the other.

This report concerns the feasibility of integrating two independent distributed computing systems, GVW and TORQUE [3] (a popular Open-source LRMS), using the newly-released Workspace Pilot [4]. The environment and configuration of the two systems is described. Results from testing are discussed. Security, networking, and OS image management are beyond the scope of this report.

3 Systems

3.1 Environment

A test cluster was setup using one head node and three worker nodes. Scientific Linux 5.0 was installed on each node. Password-less SSH and NFS were configured within the cluster to allow automated file transfers. The nodes were time-synchronized using NTP, as required by GVW.

Hostname	Physical Memory (MiB)	GT4/GVW TP	TORQUE	Xen
gridsn	2048	4.0.5	2.0.0p11	
gsn-wn1	1024	1.3.1	2.0.0p11	3.1
gsn-wn2	512	1.3.1	2.0.0p11	3.1
gsn-wn3	512	1.3.1	2.0.0p11	3.1

3.2 TORQUE

3.2.1 Introduction

TORQUE is an Open-source implementation of the Portable Batch System (PBS) [6] Local Resource Management System (LRMS) and is currently the only LRMS supported by the Workspace Pilot. With TORQUE, jobs are submitted to a queue on a cluster head node, scheduled, then delegated to worker nodes. Typically, jobs are in the form of shell scripts that will execute commands and programs that are assumed to exist on the worker node. If the worker node software prerequisites are not met, the job will fail. Environments vary between clusters and with time; they are moving targets, usually outside the control of the grid user.

3.2.2 Scheduling

TORQUE uses a dedicated scheduler to determine the order of job dispatching. The scheduler component is modular and may be replaced with other commercial options such as the Maui Cluster Scheduler™ [7]. Using the default configuration and scheduler, jobs in the queue are scheduled as FIFO.

3.2.3 Configuration

A stable point release of TORQUE, 2.0.0p11, was installed on each worker node. SCP was selected as the default transfer mechanism.

3.3 Globus Virtual Workspaces

3.3.1 Introduction

Globus Virtual Workspaces is a component for GT4 that introduces resource abstraction via virtual workspaces. Individual workspaces are implemented using virtual machines. Currently, Xen [8] is the only VMM available to use with GVW; support for additional VMMs is planned for the future.

3.3.2 Configuration

The latest release of GVW, TP1.3.1, was installed on each worker node; GT 4.05 was installed on the head node.

3.3.3 Interaction

GVW functionality is exposed via web services implemented in the GT4 container. Providing essential functionality are the Workspace Factory Service, for generating new workspaces; and the Workspace Service, for managing existing workspaces. The Workspace Client communicates with these services, to authenticate users and control virtual workspaces. Additional services exist for homogeneous group deployments, heterogeneous ensemble deployments, and status information.

GVW maintains an interal resource list of worker node hosts and IP addresses, stored in the workspace persistence database. Each available node is identified as a *resource slot*. The persistence database must be properly updated when there is a state change, otherwise it will become inconsistent and may incorrectly report that free resources are not available.

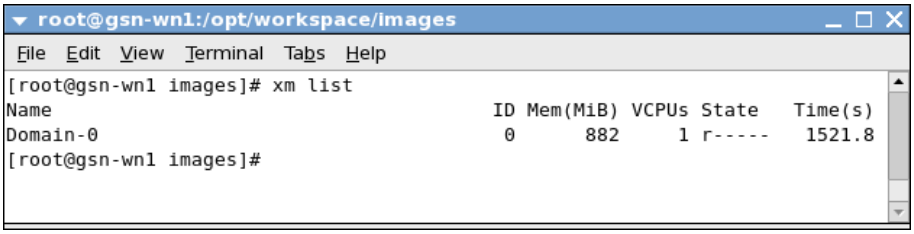

Figure 1: Domain 0's total memory is 882MiB with no guest operating systems running.

3.4 Xen

3.4.1 Introduction

Xen is a VMM that is ideal for scientific computing for multiple reasons. First, it is freely available Open-source software. Second, Xen has performance that is much greater than that of a traditional VMM through paravirtualization; this contrasts with the typically large performance losses of fully virtualized environments [9]. Instead of attempting to duplicate the identical host machine hardware, paravirtualization presents guest operating systems with a simplified interface.

In Xen terminology, the host operating system is referred to as the dom0, or Root Domain. Guest operating systems that are run in the Xen hypervisor belong to domU, or User Domain.

3.4.2 Requirements

Xen generally requires source modifications to guest operating systems to run them in a hypervisor. Open-source operating systems, such as the Ubuntu [10] and Scientific Linux [5] distributions, have been successfully modified to both support the Xen hypervisor and to run as guest operating systems.

Unmodifiable guests, such as the proprietary Microsoft Windows™, may be run in a Xen hypervisor if there is hardware virtualization support present; Intel's VT [12] (codename "Vanderpool") and AMD's AMD-V [13] (codename "Pacifica") virtualization hardware are supported by Xen [11]. Intel VT and AMD-V are becoming increasingly common, especially in server markets where there is high demand for virtualization solutions.

3.4.3 Memory Ballooning

Xen allocates memory using the Xen balloon driver. To facilitate guest operating systems, the host operating system (dom0) memory is "ballooned down", reducing it by the amount required for the guest operating system. From the dom0 user's perspective, the total memory decreases; this contrasts with the used memory increasing, as would be the case when loading a program in dom0, for example.

3.4.4 Configuration

Xen 3.1, The latest stable version of Xen in February 2008, was installed on all of the worker nodes.

3.5 Workspace Pilot

3.5.1 Introduction

The Workspace Pilot achieves integration of TORQUE and GVW by submitting a special job to the LRMS that, when run, allocates the nodes' resources using the Xen balloon driver, and communicates with the VWS

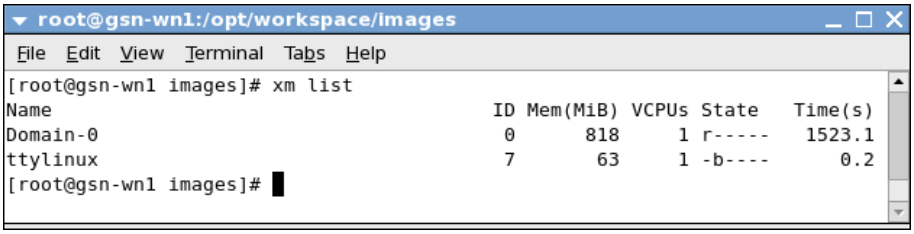

Figure 2: Domain 0's total memory has been ballooned down to 818MiB; a guest operating system ("ttylinux") is now running.

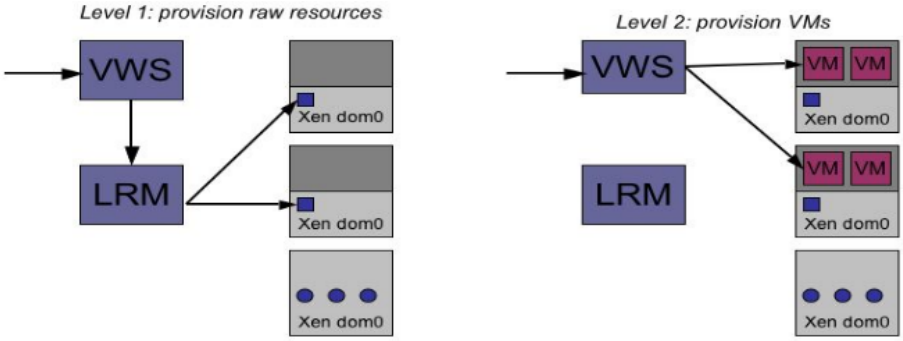

Figure 3: Two level scheduling: (1) the pilot scheduled by the LRM adjusts the memory, and (2) the obtained slots are used to schedule VMs [4].

using HTTP; SSH can be configured as a secondary communication option. The Workspace Pilot allocates node resource in two levels. First, the pilot manually balloons down the memory using Xen. Second, the GVW server sub-allocates the node's CPUs and alloted memory for individual virtual workspaces as seen in Figures 3, 4, and 5.

3.5.2 Configuration

The Workspace Pilot was configured to use HTTP and not SSH to communicate with the Workspace Service.

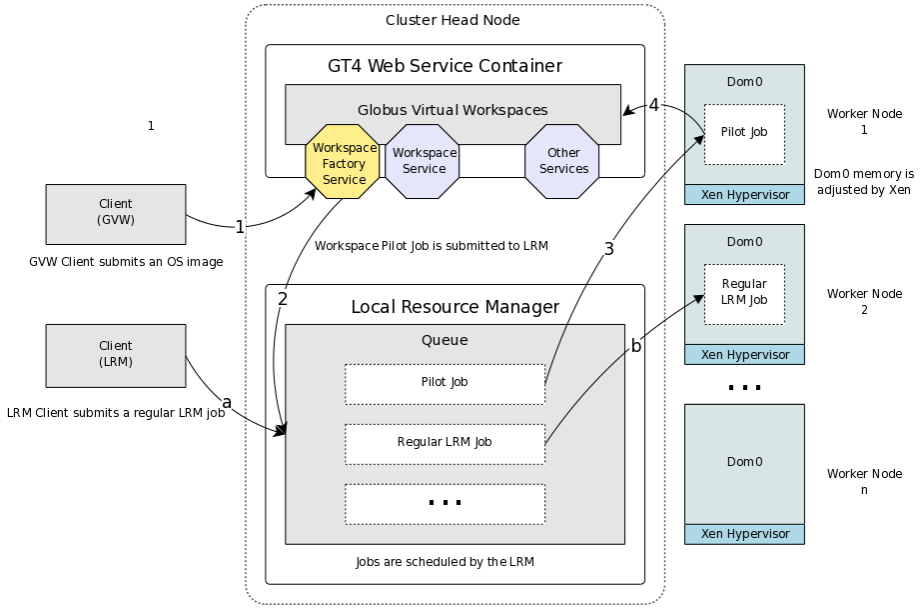

Figure 4: A GVW Pilot job is submitted, as is a regular LRM job. The Pilot job reserves a resource slot while the regular LRM job simply runs in dom0.

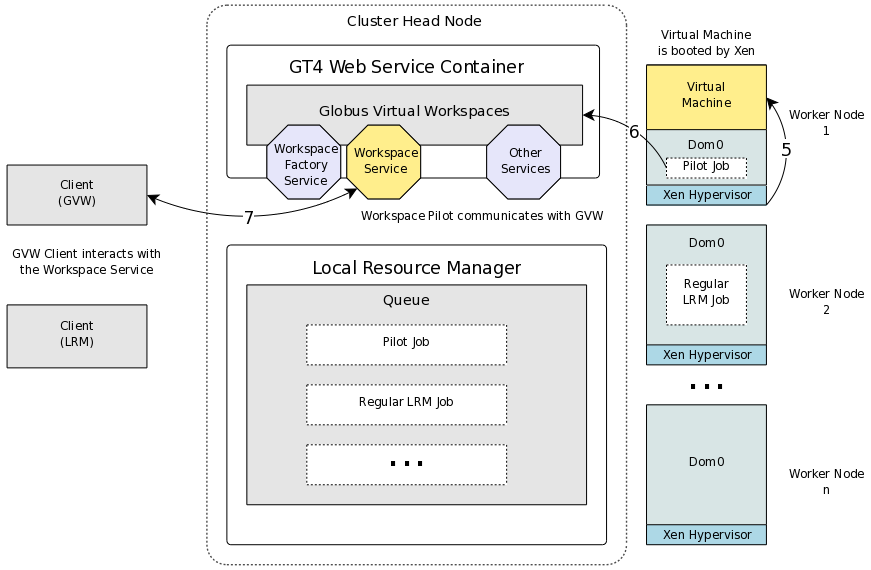

Figure 5: The GVW Pilot job uses the ballooned down memory and launches a VM.

4 Testing

4.1 Motivation

Of primary concern was ensuring that the Workspace Pilot operated as expected when using the Workspace Client. As the Workspace Pilot further sub-allocates resources independently of the LRMS, a new point of failure was added to the workspace deployment cycle. Of secondary concern was verifying the ability for the LRMS to operate independently of GVW. To the LRMS, virtual workspaces should not necessarily be distinguishable from traditional LRM jobs.

4.2 Trials

The trials were constructed to demonstrate successful prepropagated OS image deployments, and node resource allocation of the Workspace Pilot. Specifically, memory allocation by the Workspace Pilot via the Xen balloon driver was monitored.

In each test, 100 total workspaces were deployed per node, with 45 seconds allowed for the three workspaces to instantiate, and 20 seconds allowed for removal. Worker node memory was recorded at 1-second intervals for the duration of each trial.

Test A

Test A was a deployment and destruction of workspaces. This simulated interacting with the Workspace Client and no intervention from the LRMS. The LRMS queue contained only Workspace Pilot jobs from each trial.

Test B

Test B was a deployment and deletion of workspaces. This simulated interruption from the LRMS by deleting Workspace Pilot jobs from the LRMS queue.

Test C

Test C was a deployment and deletion of workspaces, as in Test B. Additionally, the GVW persistence database was reset between trials to determine reproducibility of resource slot discrepencies revealed by Test B.

4.3 Utilities

The testing utilities were comprised of two deployment scripts, a node memory monitor, and plotting tools.

4.3.1 Deployment

Two deployment scripts were created to deploy the workspaces at set intervals. Upon allowing a set time for the workspaces to deploy, the workspaces would be removed, in one case (Test A) by the Workspace Client and an EPR, and in the other (Test B, Test C) by using the TORQUE "qdel" command. These scripts looped for a specified number of times, then exited.

4.3.2 Monitoring

The node memory monitor recorded a worker node's dom0 memory at a fixed interval, along with the UNIX epoch time and a human-readable time, saved in CSV format for convenience. The CSV format could be easily transformed into another format or directly loaded into a spreadsheet.

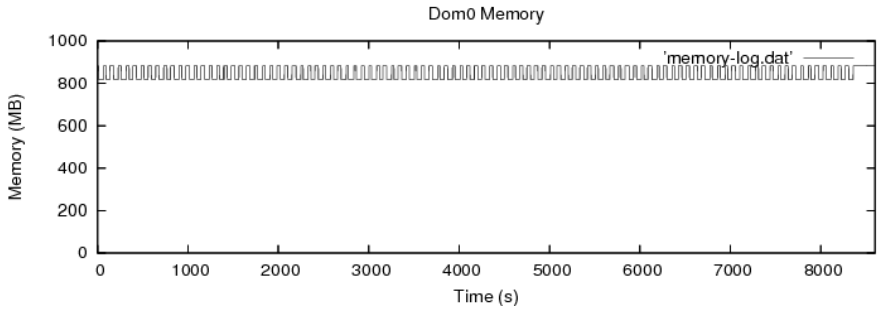

Figure 6: Test A: Success; 100 trials deployed and destroyed successfully.

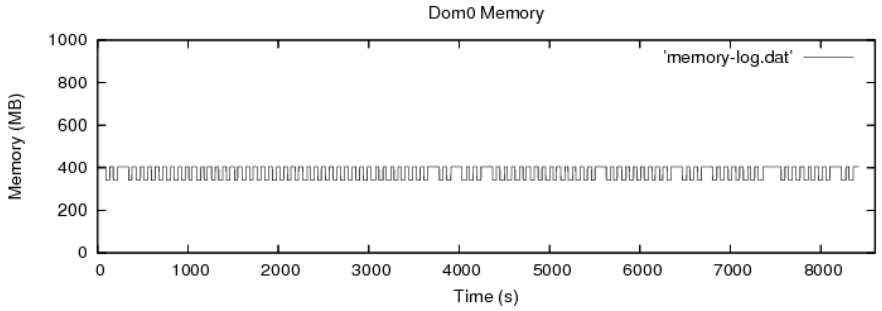

Figure 7: Test A: Several virtual workspaces do not deploy in time, due to LRMS latency.

4.3.3 Plotting

The plotting tools would convert CSV-formatted memory statistics into a format suitable for gnuplot, a plotting program. Plots were generated of dom0 memory on the y-axis versus time on the x-axis, to show the memory fluctuation as each trial proceeded.

5 Results

5.1 Local Cluster

Test A

Test A did not result in any memory discrepancies. However, the plots showed that some deployments did not complete in the allotted time, indicating excessive latency on the part of TORQUE.

Test B

Results from Test B revealed discrepancies in resource allocation. The dom0 memory does not correctly fluctuate between deployments in certain instances. These discrepancies could be reproduced, albeit randomly.

Test C

Results from Test C were akin to Test B. Resource slot discrepancies were reproduced.

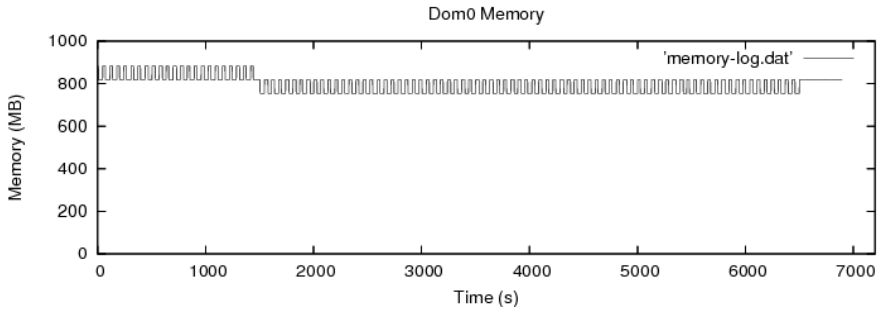

Figure 8: Tests B, C: A deployment amount discrepancy (64MiB) occurs.

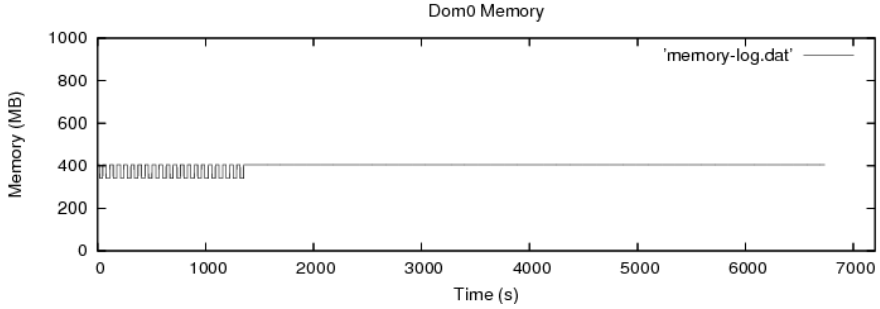

Figure 9: Tests B, C: A resource slot becomes permanently unavailable due to an inconsistent persistence database.

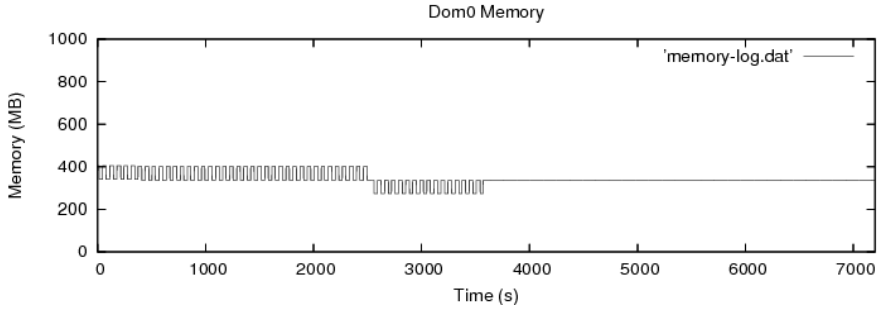

Figure 10: Tests B, C: A small discrepancy (4MiB) and a deployment amount discrepancy (64MiB) occur; a resource slot becomes permanently unavailable.

6 Discussion

Ideally, each plot would be a square wave with 100 high-to-low and 100 low-to-high transitions, corresponding to 100 successful deployments. Additionally, the duration of every low or high segment would be approximately equal, indicating that minimal latency occurred.

The plots of the various memory trials reveal characteristic patterns when problems occur. First, when a resource slot becomes unavailable, the square wave becomes a constant; the Dom0 memory is no longer ballooning up or down. Second, when Dom0 memory is not correctly restored there will be either a subtle drop of 2MiB to 5MiB or large drop of 64MiB. The plots will continue as before, shifted down by the amount of the disrepancy.

7 Conclusions

Testing of the Workspace Pilot revealed problems in the current implementation when interrupting jobs with the LRMS:

1. Small memory discrepancies of 1MiB and 4MiB.
2. Deployment discrepancies of 64MiB.
3. Persistence database inconsistency.

LRMS administrators must be able to manage queues independently of GVW. If Workspace Pilot jobs can not be reliably deleted by the LRMS administrator, this independence is breached. This was, however, the first release of the Workspace Pilot; solutions are being sought to address these outstanding issues for future versions.

8 Future Work

Similar testing should be repeated on subsequent versions of GVW. The test suites should be refined to allow use on arbitrary clusters. Additional testing should be performed (in no particular order) for:

- Propagation
- Large images (full Scientific Linux distributions)
- I/O

9 Acknowledgments

I would like to thank Dr. Randy Sobie for this work term opportunity. Thanks to Ian Gable and Dr. Ashok Agarwal for guidance on this work term. Additional thanks to Dan Vanderster, Ron Desmarais, Gregory King and David Grundy for technical help and advice. I would also like to acknowledge the support of the Globus Virtual Workspaces developers at Argonne National Laboratory: Tim Freeman and Kate Keahey.

10 Glossary

AMD-V™
AMD Virtualization; hardware virtualization support for AMD processors such as AMD Opteron™.

EPR
End Point Reference; used for tracking deployed virtual workspaces.

FIFO
First In, First Out; a simple scheduling algorithm, where jobs are processed in "First come, first served" fashion.

GT4
Globus Toolkit 4. The *de facto* grid middleware.

GVW
Globus Virtual Workspaces.

Intel® VT
Intel Virtualization Technology; hardware virtualization support for Intel processors such as Intel Xeon™.

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

NTP
Network Time Protocol.

PBS
Portable Batch System.

SCP
Secure Copy. Part of the SSH suite, used for secure remote copies; deprecates RCP.

TORQUE
Terascale Open-source Resource and QUEue manager, an LRMS.

VMM
Virtual Machine Monitor, used for managing virtual machines.

Xen
Open-source VMM used by GVW.

References

[1] Globus Virtual Workspaces http://workspace.globus.org/index.html

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

[3] TORQUE Resource Manager http://www.clusterresources.com/pages/products/torque-resource-manager.php

[4] Freeman T, Keahey K 2008 Flying Low: Simple Leases with the Workspace Pilot. http://workspace.globus.org/papers/workspace-pilot-paper-submitted.pdf

[5] The Scientific Linux Distribution https://www.scientificlinux.org/

[6] Portable Batch System (PBS) http://www.pbsgridworks.com/

[7] Maui Cluster Scheduler http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php

[8] Xen Virtual Machine Monitor http://www.xen.org

[9] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, and Warfield A, 2003 Xen and the art of virtualization. Proc. of the Ninteenth ACM Symposium on Operating Systems Principles (Bolton Landing, NY, USA) 164-177

[10] The Ubuntu Linux Distribution http://www.ubuntu.com/

[11] Xen FAQs http://xen.xensource.com/faqs.html

[12] Intel Virtualization Technology http://www.intel.com/technology/virtualization/index.htm

[13] AMD Virtualization www.amd.com/virtualization

[14] Bartle D, Gable I 2008 TP1.3.1 Testing. https://wiki.gridx1.ca/twiki/bin/viewfile/Main/WorkspacePilotTesting?rev=1;filename=uvic-gvwtp131-testing.pdf