Secure Integration of Desktop Grids and Compute Clusters Based on Virtualization and Meta-Scheduling

Christian Schridde, Hans-Joachim Picht, Michael Heidt, Matthew Smith and B. Freisleben

[1] Department of Mathematics and Computer Science, University of Marburg,
Hans-Meerwein Straße, D-35032 Marburg, Germany
[2] *email:* {schriddc, picht, heidt, matthew,
freisleb}@informatik.uni-marburg.de
phone: (+49 6421) 28 21 561, *fax:* (+49 6421) 28 573

Abstract

Reducing the cost for business or scientific computations, is a commonly expressed goal in today's companies. Using the available computers of local employees or the outsourcing of such computations are two obvious solutions to save money for additional hardware. Both possibilities exhibit security related disadvantages, since the deployed software and data can be copied or tampered if appropriate countermeasures are not taken. In this paper, an approach is presented to let a local desktop machines and remote cluster resources be securely combined into a singel Grid environment. Solutions to several problems in the areas of secure virtual networks, meta-scheduling and accessing cluster schedulers from desktop Grids are proposed.

1 Introduction

Many business applications have high computational demands while at the same time being on a tight schedule. This leads to the tentative adoption of in-house high-performance computing centres (HIPC) and in-house Grid computing environments. Both the operation of an in-house HIPC and the installation of an in-house Grid are costly solutions, since the underlying hardware must be powerful enough to deal with computational peak loads at any time, leading to an over-investment in computing power. From a security viewpoint, it makes sense to restrict access to software, data and hardware resources to the departments involved in a particular application. However, this further increases the amount of hardware needed, since each department must cope with peak loads separately. Two alternative approaches offer solutions to the problem of computational peak loads: (a) the use of non-dedicated in-house hardware resources in a desktop Grid, and (b) the outsourcing of peakloads to high performance computing centres in an inter-departmental or even inter-organizational on-demand Grid. Both these approaches offer great potential for reducing the cost of business applications, by utilizing an on-demand Grid to reduce the amount of in-house computational hardware and infrastructure needed to fulfil company goals. However, both the integration of in-house desktop workstations and the on-demand integration of out-of-house HIPCs create a number of new security threats that must be dealt with. In this paper, we discuss how these new threats can be treated and present a security solution which works

for desktop Grids as well as for high performance computing centres. A proposal to connect both types of computation infrastructures, tied together with a shared security concept, is also presented. The proposed approach is based on using dynamic virtualization, VPN networks and meta-scheduling.

This paper is organized as follows: Section 2 presents an overview of our approach to securely coupling a virtualized desktop Grid with a virtualized Cluster environment. Section 3 discusses the issues concerning the Virtual Private Network needed for the coupling of the two environments. Section 4 introduces the changes made the GridWay Meta-Scheduler to facilitate transparent secure scheduling across the desktop Grid and discuses the integration with the virtual cluster scheduling solution XGE. Section 5 introduces the related work and Section 6 finishes paper with some conclusions and future work.

2 Proposed Approach

By definition, a Grid is a multi-user infrastructure which allows multiple and concurrent usage of resources. In a desktop Grid, data is deployed onto idling computers of desktop workstations, whose computing power is utilized to process subtasks of a Grid application. To ensure a secure execution of the copmute job both the software and the data needed for the computation must be copied and stored in an encrypted from. Furthermore the computation can not be executed in the current users session. To approach the associated security threats of a desktop Grid, users on the same node have to be separated from each other. No user is allowed to gain knowledge about another party on the same resource, no matter if it is a local or a remote user. All running processes, the data and all arising logs must be secured from unauthorized access. Only if these requirements are fulfilled, companies will integrate desktop Grids into their business models. Keahey et al. [1] have presented an approach to set up and manage *virtual workspaces* within a Globus GT4 Grid environment. The intention is to allow Grid users to have individual demands on the computing environment. The authors utilize the virtualization tools XEN [3] and VMWare [4] to implement their idea. Since virtualization accomplishes the required separation of users against each other, the virtual workspaces approach is used in our approach.

In figure 1, a Grid infrastructure based on virtual workspaces is presented. A Grid user (bottom left) of a large company wants her self coded Linux tool to compute some complex problem. She specifies the required operating system environment, the binary files of the programmed tool and some other config settings. The Virtual Workspace Service accepts the input and creates a XEN image based on the specifications, including all required ingredients. The Grid middleware allocates free resources within the Grid built, for example, from idling desktop computers of other employees which are able to assist in the computation. Next, the XEN images are deployed to each of the allocated Grid resources. The installed virtualization software receives the image and starts it using job specific network interfaces and sub-networks. Finally, the job is executed within the secured virtual environment. For some applications, it is desirable to extend the virtualized desktop Grid environment by the computational power of an appropriate high performance computing cluster. To bring these two worlds together,

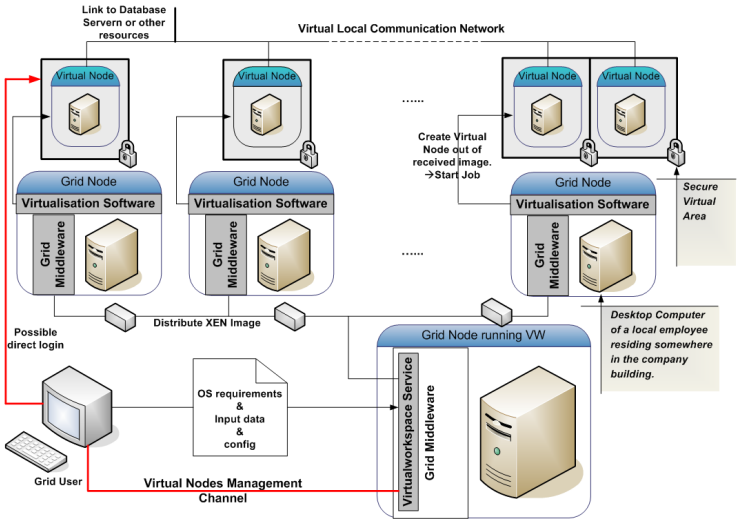

Figure 1: A Grid Infrastructure based on Virtual Workspaces

there must be a functional interface which connects the virtualized desktop Grid infrastructure to the cluster scheduling software. In [2], we have shown how the Sun Grid Engine (SGE) [5] can be extended to submit jobs to virtualized cluster nodes. Figure 2 shows how this solution can be used to achieve the "'marriage"' between the virtualized desktop Grid and the virtualized cluster nodes. The cluster's headnode is now part of the Grid, acting as an especially powerful Grid node with additional parameters. The enhanced meta-scheduler GridWay [7] manages the course of the computation, and the Grid-enabled VPN gateway provides secure virtual communication.

To realize the approach shown in Figure 2, several problems had to be solved,

1. *How to setup a secure virtual communication network, which is interconnected with the cluster headnode*
2. *How to combine existing Grid schedulers with the running cluster scheduler*
3. *How to use the modified SGE (XGE) from within the virtualized desktop Grid*

The rest of the paper deals with the solution of these three problems.

3 Grid-VPN

To solve the first problem, a VPN[1]-like approach to build-up a virtual, local network between all virtualized Grid nodes was employed. While the idea is straightforward to implement for a group of virtualized desktop nodes, the connection to the external cluster-head requires additional effort. A dynamic, grid-enabled VPN gateway was necessary to realize this approach. There exist several types of VPN connections, e.g, Site-to-Site, Site-to-End, and End-to-End. In this work we need to join two, or

[1] Virtual Private Network

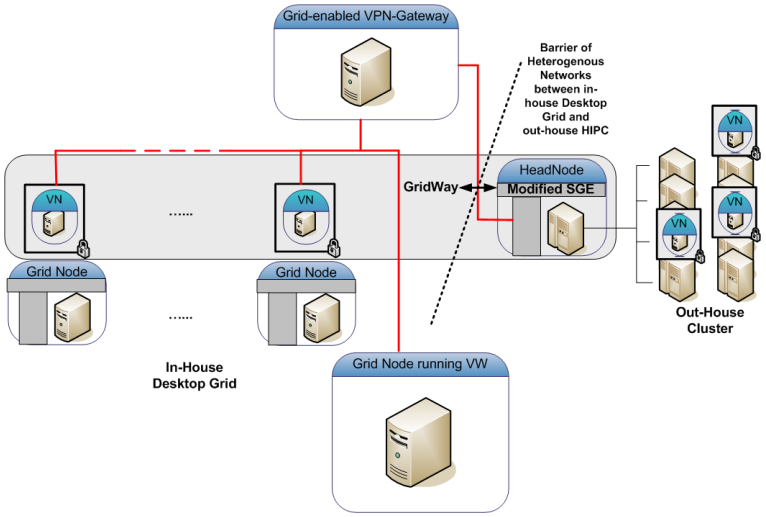

Figure 2: A combined virtualized Desktop Grid and Cluster Environment

probably more sites using the VPN technology using the Site-to-Site technology. The advantage of VPN is that the participating nodes feel like they are in a Local Area Network and which we encrypted. In order to establish a connection link between the high performance cluster and the grid nodes a VPN-based overlay network on top of an existing internet connection is established. This requires the cluster headnode to act as a communication gateway for the cluster nodes as well as the desktop grid nodes. As the baseline communication protocol for the virtual network, OpenVPN[17] is used. [2] We configured OpenVPN in such a way, that point to point tunnels between the virtualized Grid nodes are established. To authenticate the virtualized desktop nodes certificates can be used as well as a pre-shared private key. We have chosen OpenVPN, since IPSec VPNs are either too expensive or too complex. Through the virtual private network the virtualized grid nodes as well as a high performance cluster (HPC) can share a certain ip address space. This technique works for both for the entire HPC setup as well as a certain partition of a dedicated cluster. Furthermore the fact of using virtualized nodes simplifies the process of choosing a common private ip-space.

4 The Meta-Scheduler GridWay

The second and third problem dealt with in this work, was addressed by modifying the meta-scheduler GridWay [7]. The cluster acts as a "'normal"' Grid node, but has a larger computing power and the additional parameter of available (virtualized) compute nodes. The meta-scheduler must take this into account to work properly and to benefit from these two environments. We used GridWay's plugin-mechanism to specify new scheduling properties to solve this problem. In its current version, GridWay supports

[2] OpenVPN is an open source virtual private network software, written by James Yonan.

the Sun Grid Engine as a backend cluster scheduler. Since we extended the SGE to support virtualization, we had to adapt the interface between both schedulers.

4.1 GridWay and the Desktop-Grid

For several purposes we decided to connect the virtualized desktop nodes among each other via a P2P Software. The two main reasons are to gracefully deal with node failure and to utilize the distributed file storage offered by the p2p middleware. The nodes are organized in a logical overlay network in form of a dynamic hash table (DHT) using the implementation of FreePastry[22]. Since GridWay's submission mechanisms revolve around the concept of dedicated cluster headnodes, our Desktop Grid solution employs headnode virtualization, following a reverse server model. For each virtualized desktop pool, a headnode is elected dynamically[3] at run time. This chosen machine contacts the GridWay node, updating it with status information and providing a communication channel for receiving GridWay commands. In order to account for different performance characteristics exhibited by desktop and cluster resources, appropriate performance metrics had to be defined and implemented. To achieve this, we made use of GridWay's modular architecture. GridWay delegates most of its functionality to Mid-

[3]Using, e.g, the hash of the string "'headnode"' to determine the unique DHT node.

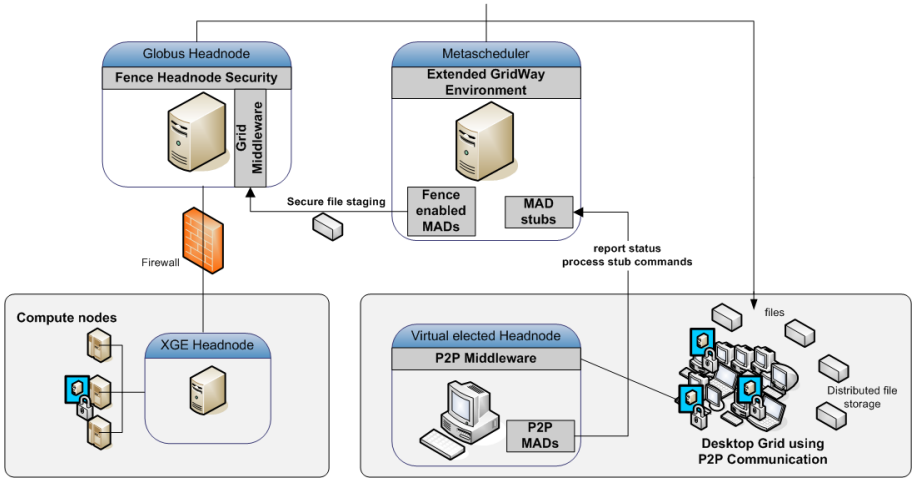

Figure 3: On the left side the cluster environment together with the Grid Headnode is illustrated. The Meta-Scheduler GridWay is connected with its designated interfaces (MAD) to both sides. On the right side the desktop Grid is shown. The nodes are connected together with a P2P System to circumvent the problem of desktop node failure. A random node is elected dynamically as a contact point for the Meta-Scheduler to receive and relay commands.

dleware Access Drivers (MADs)[4]. By providing MADs of our own, we were able to add the aforementioned functionality. On the right side in figure 3 the desktop grid with its elected virtual headnode is illustrated.

4.2 GridWay and the XGE

Our extended SGE is protected by a custom security solution called *Fence*[5], separating Globus and XGE headnodes. In order to deal with the implications this decision has upon file-staging mechanisms, GridWay's staging mechanism had to be modified and additional preprocessing of job descriptions became necessary. Normally GridWay uses GridFTP to copy and retrieve required data to and from the Globus-Headnode and furthermore uses GRAM to submit the actual jobs. This approach assumes that there exists a shared-filesystem, like NFS, between the nodes. For security reasons a shared filesystem was not an option, so in our approach ZIP-Archives are used to store all the required data and are transfered via custom data handling daemons across the Fence. To accomplish this, the GridWay's Middleware Access Drivers Mechanism needed to be extended to specify the all the necessary actions. Once received at the Fence-Headnode, RSL is used to hands the data over to the XGE.

Below, the extension to the Execution MAD of GridWay to support Fence during job-preprocessing is shown.

```
package xml;
import javax.xml.transform.TransformerConfigurationException;
import org.apache.log4j.Logger;
public class FenceRSLTransform extends StagingRSLTransform {
static {
logger = Logger.getLogger(FenceRSLTransform.class.getName());
}
public FenceRSLTransform()
throws TransformerConfigurationException {
super();
}
@Override
protected String xsltAddFilename() {
return "etc/rsl-add-fence.xml";
}
@Override
protected String xsltGetFilename() {
return "etc/rsl-get-fence.xml";
}

}
```

[4]There exits four different types: Information MADs, Execution MADs, Transfer MADs and Scheduler MADs

[5]Fence is a tool developed at the University of Marburg within the diploma thesis of Matthias Schmidt

5 Related Work

A lot of work was done regarding the research of secure grid environments. Foster et al. [12] have identified the need to integrate the advantages of virtual machines in the cluster and Grid area. It is argued that virtual machines offer the ability to instantiate an independently configured guest environment for different users on a dynamic basis. The concept of virtual clusters is also treated in [9]. The proposed system Maestro is a method for creating secure on-demand virtual clusters. The concept addresses on-the-fly-virtual machine creation used in on-demand environment as well as the security advantages which the program execution within sandboxes brings. Three other approaches are [13],[13] and [14], whereof the first two are utilizing VMWare and are not meant to be run on a cluster. The third one, based on a special Linux Kernel, allows process migration for cluster balancing. None of the above approaches deal with the concerns of embedding a desktop Grid in a virtual Cluster environment.

Yang et al. [18] contributed to the current literature of grid computing research by introducing the concept of a virtual private grid (VPG) and corresponding supporting environments. They do also deal with the grid resource management and take into account the network security issues for grid application. Their approach leaves the grid resource management to the grid tool globus alone and is in an early and experimental state. Mache et al. [19] analyzed the performance implications of using the OpenVPN software. The main performance implications are additional latency and potentially peak CPU load on the VPN gateways. Effective bandwidth and the execution time of applications that run across VPN-connected clusters was affected only if both (1) the wide-area link is fast (above 83 Mbits/s, given on a 1.4 GHz Athlon CPU) and (2) the application is communication-intensive. Thysebert et al. [20] developed a Grid Simulator capable of evaluating Grid Scheduling Strategies on different Grid topologies used to provide accurate simulation on the network packet level. During their research the modelled Computational, Storage and Information Resources and VPN connections. These resources are managed by the Grid Scheduler, Information Service and VPN Management components. Furthermore the concept of Grid-VPN is mentioned in projects like [10], [11] but in the way to let clients connect securely into Grid using VPN technology. The proposed aim to interconnect different grid execution environments to a virtual private network to benefit from its properties has not been addressed yet.

6 Conclusion

In this paper we presented an infrastructure, which merges together the classical desktop grid with a virtualized cluster environment. By using virtualization on both sides and a continuous secure interconnection a high level of security is achieved. The connection via the Meta-Scheduler GridWay was made possible through an extension which describes a working interface between both sides. On the one side it is plugged to the extended SGE. On the other side it is connected to a dynamically elected Desktop-Grid Headnode. Using the GridWay's plugin interface mechanism MAD, we were able to manage the connection as well as to consider the particular properties of the desktop

Grid. Through the usage of a grid-enabled VPN Gateway to connect interorganizational SubGrids a large "'local"' Grid-Infrastructure was formed, allowing easy data transfer and resource access in a secure manner.

Future work consists of an extensive analysis of the performance characteristics of the presented solutions and its impact on peak-load management. To further increase the level of security, an integration of Trusted Computing technology will be analyzed in respect to remote atestation of the desktop Grid resources.

7 Acknowledgements

This work is financially supported by the German Federal Ministry of Education and Research (BMBF) (D-Grid Initiative, InGrid Project).

References

1. K. Keahey, I. Foster, T. Freeman, X. Zhang, D. Galron, *Virtual Workspaces in the Grid*, Proceedings of Parallel Processing, 11th International Euro-Par Conference, Lisbon, Portugal, August 30 - September 2, 2005, pp 421–431
2. Niels Fallenbeck, Hans-Joachim Picht, Matthew Smith, Bernd Freisleben, *Xen and the Art of Cluster Scheduling*, In: First IEEE/ACM International Workshop on Virtualization Technology in Distributed Computing (held in conjunction with SC06,) Tampa, Florida, IEEE Press, 2006
3. Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebar, I. Pratt, and A. Warfield. *Xen and the Art of Virtualization*. SOSP '03: Proceedings of the nineteenth ACM Symposium on Operating Systems Principles, pp 164–177, Bolton Landing, NY, USA
4. VMware: http://www.vmware.com.
5. W. Gentzsch (Sun Microsystems), *Sun Grid Engine: Towards Creating a Compute Power Grid*, CCGRID '01: Proceedings of the 1st International Symposium on Cluster Computing and the Grid, 2001, pp 35, IEEE Computer Society, Washington, DC, USA
6. S. Santhanam, P. Elango, A. Arpaci-Dusseau, and M. Livny, *Deploying Virtual Machines as Sandboxes for the Grid*, in Second Workshop on Real, Large Distributed Systems, 2005, p. 712.
7. GridWay: http://www.gridway.org/
8. Hans Lï¿½r, Hari Govind V. Ramasamy, Stefan Schulz, Matthias Schunter, Christian Stble: *Enhancing Grid Security Using Trusted Virtualization*, accepted to be presented at The Second Workshop on Advances in Trusted Computing (WATC '06 Fall).
9. N. Kiyanclar, G. A. Koenig, and W. Yurcik, *Maestro-VC: A Paravirtualized Execution Environment for Secure On-Demand Cluster Computing*, in CCGRID 2006: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID 2006). Washington, DC, USA: IEEE Computer Society, 2006, p. 28.
10. Herbert Rosmanith and Dieter Kranzlmuller, *glogin - A Multifunctional, Interactive Tunnel into the Grid*, Grid,Vol. 00,2004,pp 266-272, IEEE Computer Society, Los Alamitos, CA, USA
11. Fabio Baroncelli and Barbara Martini and Luca Valcarenghi and Piero Castoldi,*A Service Composition Model for Automatically Switched Transport Networks*,icns,Vol. 0,2006,ISBN 0-7695-2622-5,p. 61, IEEE Computer Society, Los Alamitos, CA, USA
12. I. Foster, T. Freeman, K. Keahy, D. Scheftner, B. Sotomayer, and X. Zhang, *Virtual Clusters for Grid Communities*, in CCGRID 2006: Proceedings of the Sixth IEEE International

Symposium on Cluster Computing and the Grid (CCGRID2006). Washington, DC, USA: IEEE Computer Society, 2006, pp. 513-520.

13. I. Krsul, A. Ganguly, J. Zhang, J. A. B. Fortes, and R. J. Figueiredo, *VMPlants: Providing and Managing Virtual Machine Execution Environments for Grid Computing*, in SC 2004: Proceedings of the 2004 ACM/IEEE conference on Supercomputing. Washington, DC, USA: IEEE Computer Society, 2004, p. 7.
14. G. Valle, R. Lottiaux, D. Margery, C. Morin, and J.-Y. Berthou, *Ghost Process: a Sound Basis to Implement Process Duplication, Migration and Checkpoint/Restart in Linux Clusters*, in The 4th International Symposium on Parallel and Distributed Computing, Lille, France, July 2005, pp. 97–104.
15. B. Lin and P. A. Dinda, *VSched: Mixing Batch And Interactive Virtual Machines Using Periodic Real-time Scheduling*, in SC 2005: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing. Washington, DC, USA: IEEE Computer Society, 2005, p. 8.
16. Eduardo Huedo, Ruben S. Montero, and Ignacio M. Llorente, *The GridWay Framework for Adaptive Scheduling and Execution on Grid*, in Scalable Computing: Practice and Experience, Vol. 6, No. 3, pp. 1–8. 2005
17. OpenVPN http://www.openvpn.net
18. K. Yang, X. Guo, D. Liu, B. Yang, *Towards virtual private grid through policy-based management*, in Parallel and Distributed Computing, Applications and Technologies, 2003. p. 603 - 607
19. J. Mache, D. Tyman, A. Pinter, C. Allick, *Performance Implications of Using VPN Technology for Cluster Integration and Grid Computing*, International conference on Networking and Services (ICNS'06) , p. 75, 2006.
20. P. Thysebaert, B. Volckaert, F. de Turck, B. Dhoedt and P. Demeester, *Evaluation of grid scheduling strategies through NSGrid: a network-aware grid simulator*, Journal of Neural, Parallel Sci. Computing, Volume 12, 3, 2004, p. 253-378
21. S. Samll, A. Terzis, F. Monrose, B. Doshi, Band A. De Simone, *Scalable VPNs for the global information grid*, Military Communications Conference (MILCOM 2005), 2005. p 305-311
22. Antony Rowstron and Peter Druschel, *Pastry: A Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems*, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), November, 329–350, 2001