

A virtual TestGrid or how to replicate a national Grid

Stephen Childs, Brian Coghlan, John Walsh, David O’Callaghan, Geoff Quigley, Eamonn Kenny

Department of Computer Science

Trinity College Dublin

Ireland

Email: {stephen.childs,coghlan,john.walsh,david.ocallaghan,geoff.quigley,eamonn.kenny}@cs.tcd.ie

Abstract—A real Grid is composed of nodes at three levels. At the lowest level are the compute nodes on which jobs are run. Next there are the infrastructure servers which enable grid access to each site. Finally, there are high-level servers (such as resource brokers) that link multiple sites together to form a Grid. Complex interoperations between nodes make it difficult to adequately test middleware and application software before deployment. A realistic, large-scale testbed is needed that accurately reflects the configuration of the real infrastructure. We present TestGrid, a testbed that replicates all three levels of a national infrastructure. This replica is hosted in a self-contained network which uses the same system and network configurations as on the real infrastructure. We describe the virtualisation techniques used to reduce hardware requirements and to replicate the real network infrastructure. We also describe the benefits this testbed provides for system administrators and software developers.

I. INTRODUCTION

Grids are complicated. This is partly because Grid software is still maturing, and partly because of the inherent complexities of a large geographically distributed system. This complexity causes particular problems for the testing and deployment of new middleware. Large-scale test environments are required to adequately represent the real infrastructure and this can be an unwelcome supplement to the already large budgets required for Grid and cluster computing. However, we believe that such realistic testbeds play a critical role in the development of stable middleware, and in enabling smooth deployment of updates with minimum disruption to users.

We propose that it is possible to create a faithful replica of a relatively large Grid using modest hardware resources. The key to this is virtualisation: using virtual machine (VM) technology to run multiple OSes on one physical host, and using virtual networking to create isolated networks that mimic the real networking. In this paper we describe the design and implementation of a replica national Grid hosted within a self-contained network and demonstrate its usefulness for testing and validation of middleware and application software.

A. Context

1) *Grid-Ireland*: Grid-Ireland is the organisation that runs the national computational Grid infrastructure for Ireland. It currently comprises eighteen sites based at research institutions around the country. The core infrastructure is managed and monitored from an Operations Centre based at Trinity College Dublin. The Operations Centre also hosts national

servers that provide resource brokerage, replica location, virtual organisation management and information services to all sites. Thus, Grid-Ireland is a fully self-contained Grid providing for the needs of Irish virtual organisations, as well as providing a channel for extra-national collaborations.

The small size of Grid-Ireland makes it possible to run a full range of Grid services from a single centre: this makes us ideally placed to develop a comprehensive Grid testbed. This work has been motivated by operational reasons as well as research priorities: as the Grid matures, users expect a stable infrastructure and ad hoc procedures for system administration are unacceptable. Any upgrade procedures must be thoroughly tested to avoid unforeseen hitches during rollout, and any experimental changes made by administrators must not be made on the real infrastructure. Small-scale testbeds are of some use, but their limited functionality implies that significant issues may only become apparent when the new software is deployed on the real infrastructure.

Grid-Ireland sites form part of the international Grid infrastructure being developed as part of the EU project Enabling Grids for E-Science (EGEE). This project mandates the use of a particular distribution of Grid middleware (currently called the *LCG* distribution but soon to be rebranded as *gLite*), which is released at regular intervals by a central deployment group. A test suite known as the *Site Functional Test* is submitted as a job to sites many times a day; this test suite performs a range of standard Grid operations in order to verify that all the functionality required by jobs is operating correctly at a site. For an introductory guide to the basic services and tools used on the LCG/EGEE grid, please refer to [10].

2) *State of the art*: The use cases mentioned already exist. What are the current approaches used for developing and certifying Grid software and testing its deployment? The predominant approach is the construction of a Internet-accessible development testbed that is operated in parallel with the production service. Examples of this approach include the CrossGrid Development Testbed [6] and the EGEE Pre-Production Testbed. Another approach is to create a private network internal to a site and set up artificial Grid sites. Either approach tends to limit the applicability of results (because the test environment differs significantly from the real infrastructure) or the control available over the environment.

The PlanetLab [2] and Grid’5000 [3] projects are imple-

menting large-scale distributed testbeds with a somewhat different focus. These testbeds are primarily intended to provide a platform for deploying experimental network services rather than for testing entire middleware distributions designed for production science work.

B. Our approach

In this paper, we propose a third approach: a testbed hosted in an isolated network that contains a combination of sites with real network addresses replicating servers on the production Grid, and sites with private network addresses for development of new middleware. All infrastructure controlled by Grid-Ireland is replicated within the testbed so that upgrades and configuration changes can be tested; network address translation (NAT) is used to provide access to Grid services external to Grid-Ireland.

This approach provides a controlled environment for experiments and allows administrators to test deployment of services using *exactly* the same configurations that will be used on the production infrastructure. The combination of experimental and replica sites allows developers of new middleware to take advantage of existing services and to verify that their software will work correctly in a production environment. We have built such a testbed, called TestGrid, at the Grid-Ireland Operations Centre and in the remainder of this paper we describe the techniques used to build it, the applications it has enabled and the lessons we have learned in over a year of use.

C. Aims

The aims of our replica national Grid testbed are as follows:

- **Comprehensiveness:** The infrastructure should be replicated in **all** essential details. All core servers in the real infrastructure should have identically-configured counterparts within the testbed.
- **Consistency:** Replica machines should normally maintain the same configuration as their external counterparts. It should be possible to resynchronise with the real servers quickly and straightforwardly.
- **Isolation:** Traffic from within the test environment destined for other machines in the Grid-Ireland infrastructure should not be allowed to escape to the “real” machines. Access to the test environment should only be allowed on a limited number of carefully controlled paths.
- **Security:** Access to the test environment should be strictly controlled and network virtualisation techniques used should not pose any additional risk either to computers within the test environment or in the real Grid infrastructure.
- **Cost-effectiveness:** A replica of a large infrastructure should be possible with many fewer machines than in the real infrastructure. This implies the widescale use of virtual machines.
- **Performance:** The test environment should have performance comparable to the real servers: this implies that the VM technology used should perform well.

- **Ease of use:** Once users have logged into the test environment, they should be able to use the same software and configuration as on the production infrastructure.
- **Flexibility:** The environment should provide users with the freedom to create their own infrastructure, and to use as much or as little of the replica infrastructure as they need to.

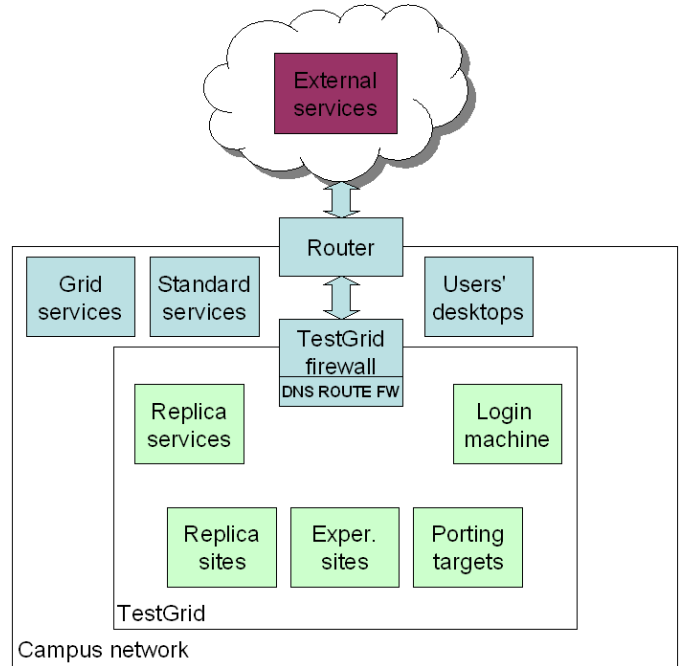


Fig. 1. TestGrid architecture

II. TESTGRID ARCHITECTURE

Figure 1 shows the basic architecture of TestGrid and its location within our network. The TestGrid firewall machine is responsible for routing traffic within TestGrid. The routing tables are carefully configured to ensure that traffic between TestGrid machines is not allowed out onto the external network, and that traffic on the external network does not accidentally enter the TestGrid.

There are two kinds of sites within TestGrid: replica sites and experimental sites. Replica sites mirror the configuration of real sites and are primarily intended for validation and testing of middleware updates and configuration changes. Experimental sites do not represent real sites and as such may deviate from the reference Grid-Ireland configuration. They are intended for use by middleware developers who may want to install experimental software or port middleware to new platforms. Figure 2 illustrates the variety of nodes and networks within TestGrid: the numbers in brackets indicate the number of physical machines dedicated to each application.

III. NETWORKING

Before describing the configuration, we will first clarify our terminology. In the following text “public” means a real

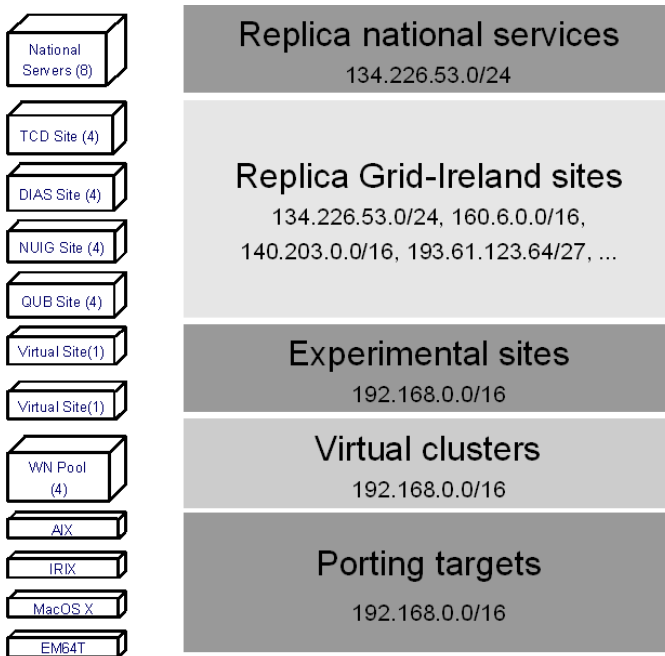


Fig. 2. High-level view of node types within TestGrid

routable address such as 134.226.53.197; “private” means a non-public address such as 192.168.0.1; “external” means outside TestGrid, on the Internet; and “internal” means inside TestGrid. So it’s possible to have “public internal” addresses. The TestGrid firewall intervenes between the internal and external networks. Because many other server-specific iptables firewalls are also used, hereafter the *TestGrid firewall* is specifically named as such in order to avoid confusion.

Nodes within TestGrid that replicate real Grid-Ireland nodes have public addresses. Their network settings (default gateway, name server, time server, etc.) are identical to those of their counterpart machines on the public Internet. The replica machines do not have general access to the public network.

TestGrid also includes private network addresses in the range 192.168.0.0/16. These are intended for setting up nodes that are intended for experimentation and so do not replicate real external nodes. The domain `.testgrid` is associated with this network and domain name resolution (DNS) services are provided to allow the assignment of hostnames within this network.

The normal entry route for user logins is via an SSH gateway located on a User Interface node (e.g. `ui.testgrid` in Figure 3). Tunnels may be set up to ease access to TestGrid nodes. Masquerading is used to provide SSH access out of TestGrid; this allows users to access their workstations from within TestGrid and makes it possible to copy data hosted on machines outside TestGrid in a single operation.

All networking routing and firewalling is performed on the TestGrid firewall, which is the single direct point of access to TestGrid. The firewall is dual-homed: one interface is connected to the external network, and the other is connected

to the internal TestGrid network.

A. Routing

For each replica site hosted within TestGrid, the internal network interface on the TestGrid firewall is assigned an additional IP address corresponding to the default gateway for that site. When thinking about the router configuration, it proved helpful to classify machines as being either external infrastructure machines, internal replicas of machines outside TestGrid, or experimental (private) machines with no counterpart in the real world.

The TestGrid firewall is set up with static routes for each of the networks in TestGrid. These routes direct all traffic for these networks over the internal interface. Thus, sites within TestGrid can contact each other: their requests go to the TestGrid firewall, which then routes them back into TestGrid to the other sites. In future we plan to add network emulation to the routing functions to allow more realistic evaluation of inter-site network behaviour.

B. iptables firewall

The rules on the TestGrid firewall prevent any traffic from within TestGrid from escaping. This is an important additional barrier within the network configuration. For example, if the routing is set up incorrectly so that traffic between TestGrid sites is being routed to the external site, the firewall will simply drop this traffic, preventing disruption of the production network.

C. Address Resolution Protocol (ARP) responses

When configured as a router, the Linux kernel normally responds to ARP requests for machines to which it has a route. We have disabled this feature on the TestGrid firewall to ensure that it does not advertise machines within the isolated TestGrid network. There is one exception: the TestGrid login machine (`ui.testgrid`) needs to be accessible from outside TestGrid, and so we configure the firewall to respond to ARP requests for an external public address that is network address translated (NAT) to the private address of the machine.

D. Standard services

Each replica site has the same network configuration as its counterpart on the real infrastructure. Therefore these machines will be looking for the name servers, web proxies and time servers at their local sites. We have configured extra aliased IP addresses on the TestGrid firewall machine allowing it to impersonate these servers, answering requests from the replica sites.

The DNS server running on the TestGrid firewall performs two roles. Firstly, it caches mappings for the replica sites from the real DNS servers on the Internet. When a new replica site is added to TestGrid, the TestGrid DNS server initially retrieves the mappings for the site’s IP addresses from an external name server (as shown in Figure 3) and then caches them so it can subsequent requests itself. Secondly, it provides name resolution for the experimental hosts that have private

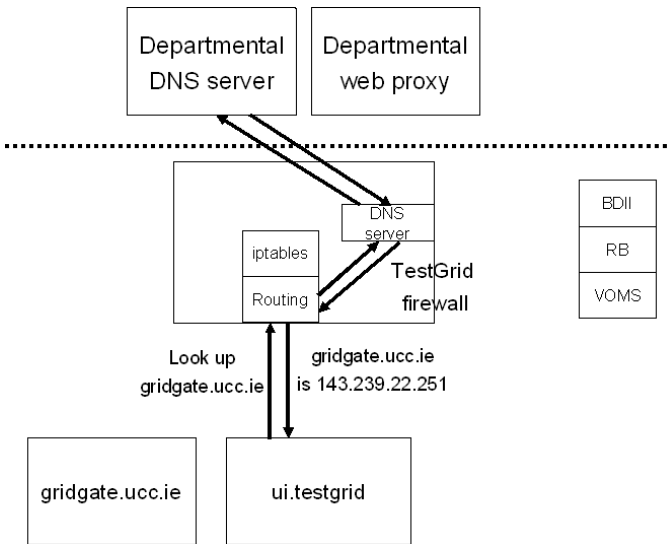


Fig. 3. Looking up an external host

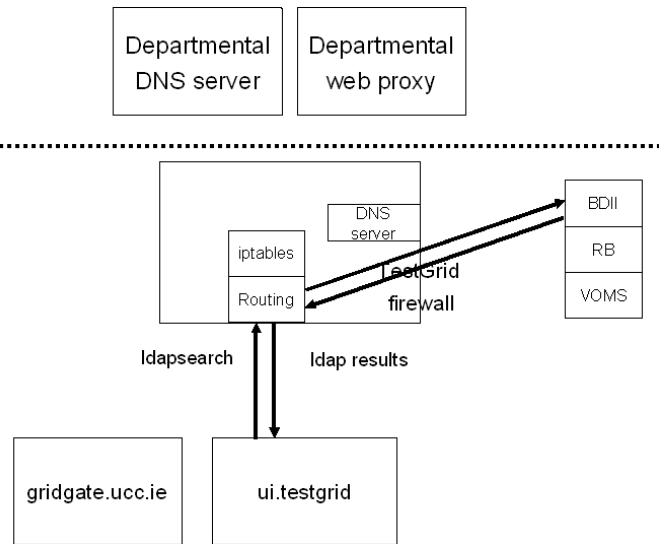


Fig. 4. Accessing Grid services within TestGrid

addresses in the `.testgrid` domain. As this is a private domain, many mappings can be set up as needed by users of TestGrid.

E. Grid services

TestGrid includes replicas of the top-level Grid-Ireland servers that provide resource brokerage, information services and replica location. Requests from TestGrid nodes for these services are handled entirely by the TestGrid replica servers. Figure 4 shows a fundamental Grid operation performed by all Grid nodes: a query of the information index server (BDII). When requests directed to the Grid-Ireland BDII reach the TestGrid firewall, they are routed back into TestGrid to be handled by the replica server. Requests to external BDII servers (for example those hosted at CERN) would be routed outside using masquerading as described in the next section.

F. Masquerading

Access to services external to TestGrid should be kept to a minimum as most essential network and Grid services are replicated as described above. However, there are a number of cases where it is not appropriate or practical to replicate services within TestGrid. For example, the Grid machines need access to external LDAP servers to determine the membership of Virtual Organisations external to Grid-Ireland. We use the masquerade feature of iptables to provide access out of TestGrid for such services. These services are enabled on a case-by-case basis – the default should be to prevent requests from travelling out of TestGrid except when this is explicitly required.

G. Summary

Table I summarises the basic services within TestGrid and how they are accessed by users.

IV. USE OF VIRTUAL MACHINES

We are aiming to substantially replicate a Grid that contains more than 100 grid infrastructure nodes, quite apart from the compute and storage resources at sites, so it is important to minimise the number of machines needed. We use virtual machine (VM) technology to run one or more sites per physical host, thus making it feasible to replicate many sites.

We have been using virtual machine technology within Grid-Ireland for some time; our User Interface servers have run as User-Mode Linux virtual machines since early 2004. Following performance evaluations [4], in late 2004 we migrated these systems to Xen [1] VM technology developed at the University of Cambridge. During 2005 we deployed mini-gateways running all nodes as Xen VMs at twelve new sites.

A. Configuration

The eighteen sites that make up Grid-Ireland have two basic configurations: mini-gateway and full gateway, and these already use VMs. Sites with mini-gateways run all servers (install server, Computing Element (CE), Storage Element (SE), Worker Nodes (WN), User Interface (UI), and network monitor (NM)) in separate virtual machines on a single physical host. Sites with full gateways have four machines which run the following servers:

- **Machine 1:** Site Firewall
- **Machine 2:** Installation server
- **Machine 3:** Computing Element (VM 0) and Test Worker Node (VM 1)
- **Machine 4:** Storage Element (VM 0), User Interface (VM 1) and optional network monitor (VM2).

Both kinds of site are replicated within TestGrid. In addition, we have installed servers that host multiple sites, virtual clusters composed of many worker nodes, build servers hosting multiple Linux distributions and corresponding test worker nodes.

Service type	Service scope	Handled by	Via
Grid	Grid-Ireland VOs	TestGrid replicas	Routing
Grid	External VOs	External Grid servers	Masquerading
DNS	TestGrid experimental machines	TestGrid DNS	Direct access
DNS	Machines with public addresses	External DNS server	Caching in TestGrid DNS
Default gateway	Replica machines	TestGrid firewall	Aliased network interfaces
Secure shell	Out of TestGrid	External servers	Masquerading

TABLE I
SUMMARY OF TESTGRID SERVICES

tg11.testgrid			tg15.testgrid		
Name	Id	Mem(MB)	Name	Id	Mem(MB)
Domain-0	0	251	gridgate05	1	199
centos4-build	1	255	gridinstall05	21	199
fc4-build	6	255	gridmon05	23	199
rh73-build	3	255	gridstore05	28	199
rh9-build	2	255	gridwn050	24	199
sl3-build	5	255			
suse9-build	7	255			
tg4.testgrid			tg18.testgrid		
Name	Id	Mem(MB)	Name	Id	Mem(MB)
Domain-0	0	251	Domain-0	0	190
c01	25	199	infogrid1	7	255
c02	27	199	infogrid3	21	255
c03	29	199	infogrid4	13	255
c04	31	199			
c05	33	199			
c06	35	199			
infogrid2	49	255			

TABLE II
CONTENTS OF VARIOUS VM HOSTS

B. Organisation

The VMs within TestGrid are organised as follows: a set of VM host machines is allocated with hostnames of the form `tg<n>.testgrid`. Each of these machines is typically associated with a particular developer or experiment, and they are responsible for managing and installing the VMs they need on that host. Table II shows a few examples of VMs running on different hosts. `tg11.testgrid` and `tg15.testgrid` are used for porting, and contain various build machines for different Linux distributions, and a test site used in validation. `tg4.testgrid` and `tg18.testgrid` are used as part of a project called “InfoGrid”, and host various experimental resource brokers, computing elements and worker nodes.

V. MANAGEMENT

A. Ensuring consistency

If the replicated machines within TestGrid are to be representative of the real infrastructure, then their configurations must be kept synchronised with their counterparts in the production infrastructure. We use the Quattor fabric management system [9] to control the configuration of OS and Grid middleware components: this ensures that machines with the same profile should in general have the same configuration.

Quattor manages almost all service configuration state, but there are a few exceptions where further synchronisation is

required. For example, the virtual organisation management (VOMS) server stores user lists in a database. When changes were made on the real network (for example, adding a new user to a particular VO), they were not propagated to TestGrid. This caused problems for users as they found that they had different access rights TestGrid than they had on the real infrastructure. We solved this by setting up a cron job that runs nightly to dump the database on the real server and copy it over to the TestGrid replica. Currently, the combination of Quattor and simple per-service synchronisation is sufficient for our needs. If more exact replicas are required, tools such as `rsync` can be used to resynchronise the replicas within TestGrid.

B. Adding new sites

There are a number of different ways to install nodes for a new site. The first option is to follow the normal LCG installation procedure. In this case, an installation server is created by installing the OS and the LCG software repository. Then profiles are created for the server and cluster nodes, and DHCP and PXE configuration is set up to allow the individual nodes to be installed automatically.

The second option is to install from a pre-existing image. As part of the deployment process for our mini-gateway sites, we have created a complete filesystem image for a Grid-Ireland sites. This includes an installation server and files hosting filesystems for the standard server nodes. This image can be transferred directly to the hard disk of a fresh machine, and then reconfigured to reflect the desired identity of the site.

C. Transactional deployment

The Grid-Ireland transactional deployment (TransDeploy) system provides a method for reliably moving a set of sites between two configuration states. Deployment occurs in two stages: a potentially lengthy prepare stage in which software packages and configuration profiles are compiled and deployed to install servers at sites, and a fast commit phase in which the deployed configuration is published to the site nodes.

TransDeploy was developed within TestGrid and then first deployed on the production infrastructure. As TestGrid grew in scope it became apparent that we needed to use TransDeploy within TestGrid to manage the increasing number of sites. We configured a server within TestGrid to act as a replica TransDeploy server, directly mounting directories from the real Grid-Ireland software and configuration repository. This was necessary because the TransDeploy software was written

to run on a repository server. Once the TestGrid TransDeploy server was operational, the task of managing the TestGrid sites became much easier: with a few clicks, operators could roll out test configurations to the replica sites, or restore them to the production configuration.

TransDeploy is a good example of the symbiotic relationship between TestGrid and the production environment. Due to the greater frequency of configuration changes in TestGrid, the performance of the TransDeploy became an issue. Enhancements were made to the TransDeploy software, tested within TestGrid and then the improved tool was deployed on the real infrastructure.

D. Router configuration

The configuration of the TestGrid firewall has evolved over time as new features have been added. We have been maintaining the configuration manually but this is not sustainable as for each new site added to TestGrid, a number of inter-dependent changes must be made: iptables rules must be configured to accept traffic for the site, routes must be set up to keep all internal traffic destined for that site inside TestGrid and network aliases must be added to impersonate the new site's default gateway and DNS servers.

All of these components can be managed by Quattor, and so we are currently testing a system for configuring all relevant components based on a single specification of the network to be added.

VI. USE CASES

TestGrid has already become an indispensable part of our development and testing procedures. In this section, we describe a few of the many ways we use it (also illustrated in Figure 5).

A. Testing transactional deployment

The Grid-Ireland transactional deployment system [5] has been designed to increase the reliability of a Grid infrastructure by ensuring the consistency of middleware between sites and by reducing the scope for operator errors during upgrades. In order to test the system, we needed a number of sites. It proved straightforward to create these within TestGrid and integrate them with the infrastructure.

B. Pre-deployment testing

We have recently deployed eleven new site gateways to institutions distributed across Ireland. Our normal procedure was to install and configure the servers at the Operations Centre, thus minimising the duration of site visits. Once the basic configuration of the server was complete, we installed it in TestGrid. We were then able to verify the configuration of the site and confirm interoperability with the Grid-Ireland infrastructure before installation at the real site. This proved to be an effective way of catching configuration errors early, thus making the actual deployment to remote sites much easier than it would otherwise have been.

C. Experimental middleware development

As part of the WebCom-G project, researchers within our group are working to develop new Grid middleware. Active projects include development of an economic resource broker, a new Grid file system, visualisation research and a novel transport mechanism using Grid information systems. TestGrid is a good development environment because developers can access a standard set of Grid servers yet can deploy potentially disruptive new middleware in complete isolation from the production service.

D. Porting

Grid-Ireland is involved in porting the LCG/EGEE worker node middleware to a number of different platforms (both Linux distributions and other Unix-like operating systems) [7]. TestGrid has proved to be an ideal environment for validating the software produced. Again, what is crucial is the mix of experimental and production-equivalent components: this gives the freedom to create new scheduler queues to host the different worker nodes, but also the assurance that comes from testing against services configured in a standard way. We have been using the EGEE Site Functional Test suite to verify that the ported software functions correctly.

VII. EVALUATION

A. Performance

1) *Networking*: The performance of TestGrid is at least equivalent to that of the real infrastructure. In fact, measurements taken on the real and replica versions of the TCD site indicate (Table III) that it tends to perform around 30% better. This is due to two factors: the hardware on which TestGrid is running is slightly more powerful, and the TestGrid infrastructure tends to be less heavily loaded than the production infrastructure. This discrepancy is acceptable as TestGrid is intended to be a functionally equivalent replica of the real infrastructure: it is not necessary for it to perform exactly the same way.

Operation	Production	TestGrid
LDAP search	103.5	70.9
Globus authentication	248	176.6
Ping remote site	9	2.3

TABLE III

BENCHMARKS OF GRID OPERATIONS (TIMES IN MS)

However, the performance of TestGrid will tend to differ greatly from the real infrastructure when it comes to the timing of network traffic between sites. This is inevitable as TestGrid sites are connected via local networking while the real sites are geographically dispersed and connected over the Irish public research network. The third row in Table III shows the results of timing TCP connections from the TCD user interface to port 2119 (Globus gatekeeper) port on the machine `gridgate.dit.ie`: the average connection time is almost four times shorter within TestGrid. While such differences are not significant for our current use of the testbed, they

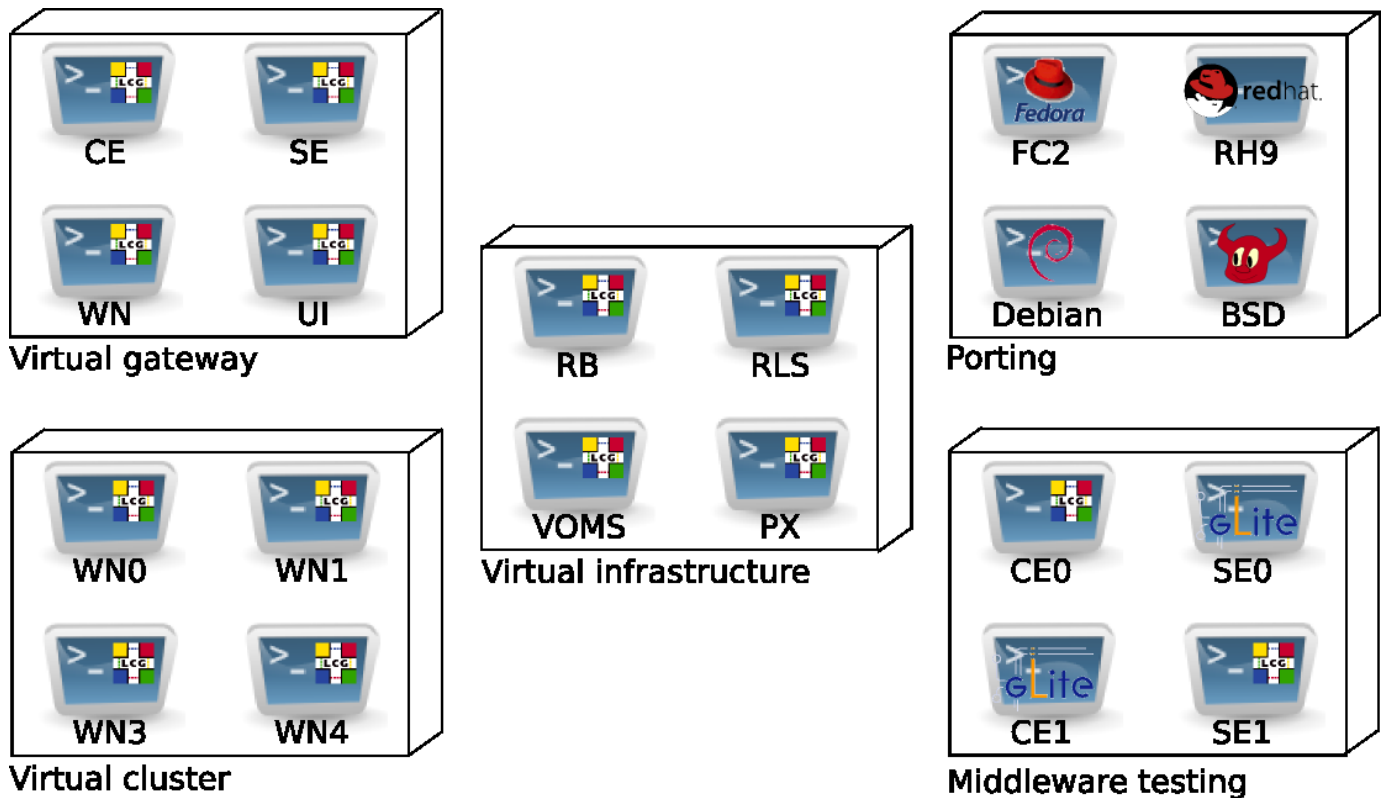


Fig. 5. TestGrid applications

will prove important future as we seek to use the testbed for characterising the performance of user applications. As such, we plan to add network modelling features into TestGrid, using round-trip time measurements obtained using a locally-developed Grid monitoring framework [8].

2) *Virtual machines:* The Xen virtual machine software has been shown to perform to within a few percent of the underlying hardware [1] and our own evaluations have shown its suitability for hosting Grid servers [4]. The principal limitation for this environment is that memory must essentially be statically partitioned between VMs on a particular host. We have ensured that the amount of memory installed in VM hosts is sufficient for the number of VMs that will run on it.

B. Lessons learned

1) *Even testbed users expect production-quality service:* There is a tension between the desire to use the replica servers for testing middleware upgrades, and the need for developers to have a relatively stable infrastructure against which to test their code. In the period before a major upgrade, the TestGrid infrastructure is likely to be unstable, and it has proved important to communicate this to testbed users in good time.

2) *Fabric management of all nodes is essential:* As TestGrid has grown and users have installed new nodes week by week, it has become increasingly difficult for administrators to keep track of the state of installed machines. We have now

decided that all VM host machines at least should be managed by Quattor: this allows for easy installation of software updates as required, and also ensures that suitable access is available to allow administrators to monitor and configure machines as necessary. We will continue to provide users with the freedom to install guest VMs however they wish: in many cases, they may want to test an particular distribution (e.g. for porting) or installation method that does not fit with the standard fabric management configuration.

3) *Network emulation should be kept as simple as possible:* We originally planned to provide public aliases for all machines within TestGrid; this would have allowed direct logins to the TestGrid machines. However, this proved to be an overly complicated approach as we had to set up MAC-level filtering and translation rules for all machines using the Linux `ebtables` kernel module. This resulted in a fragile configuration and did not actually make the system easier to use: users had to remember a list of mappings between the public aliases and the machine names within TestGrid, which was actually more awkward than “logging into TestGrid” via a single machine. The lesson we learned is to work out what users actually require and implement the simplest network configuration that can meet those requirements.

VIII. CONCLUSION

The TestGrid approach outlined in this paper enables the creation of a large-scale testbed that serves as an excellent

platform for development and deployment testing. The use of virtual machine technology allows the creation of a comprehensive replica with modest hardware resources, and the use of virtual networking within an isolated network allows real configuration parameters to be used.

The combination of accurate replicas of production servers with experimental sites provides flexibility without requiring developers to build their own complete test environments from scratch. New services can be deployed on VMs at will, yet have access to a full range of standard Internet and Grid services.

Requirement	Provided by
Comprehensiveness	Virtual machines
Consistency	Fab. management, transactional deployment
Isolation	Network router configuration
Security	Limited access paths
Cost-effectiveness	Virtual machines
Performance	Xen VM technology
Ease of use	Replica sites and services
Flexibility	Experimental sites, <code>.testgrid</code> addresses

TABLE IV
EVALUATION RELATIVE TO GOALS

Table IV summarises the techniques we used to meet the various requirements set for our test environment. We believe that this combination of standard tools and technologies provide useful building blocks for anyone seeking to construct a similar environment.

We have been using TestGrid on a daily basis for over a year now and it has become invaluable. It has grown to include a full replica of our national servers, site replicas, experimental sites, porting targets – in all, over 140 virtual machines hosted on more than 40 physical machines. Based on this experience, we would recommend our approach to anyone interested in testing large-scale Grid applications and middleware.

ACKNOWLEDGEMENTS

This work was supported by the EU EGEE project under contract INFOS-RI-508833.

REFERENCES

[1] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the Art of Virtualization. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*. ACM, 2003.

[2] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating system support for planetary-scale network services. In *Symposium on Networked Systems Design and Implementation (NSDI)*, pages 253–266, March 2004.

[3] Franck Cappello, Frederic Desprez, Michel Dayde, Emmanuel Jeannot, Yvon Jegou, Stephane Lanteri, Nouredine Melab, Raymond Namyst, Pascale Primet, Olivier Richard, Eddy Caron, Julien Leduc, and Guillaume Mornet. Grid’5000: A Large Scale, Reconfigurable, Controllable and Monitorable Grid Platform. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing, Grid 2005*, November 2005.

[4] S. Childs, B. Coghlan, D. O’Callaghan, G. Quigley, and J. Walsh. A single-computer grid gateway using virtual machines. In *Proc. AINA 2005*, pages 761–770, Taiwan, March 2005. IEEE Computer Society.

[5] B.A. Coghlan, J. Walsh, G. Quigley, D. O’Callaghan, S. Childs, and E. Kenny. Principles of transactional grid deployment. In Peter M.A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, and Marian Bubak, editors, *Advances in Grid Computing - EGC 2005*, LNCS3470, pages 88–97, Amsterdam, The Netherlands, February 2005. Springer.

[6] J. Gomes et al. Experience with the international testbed in the crossgrid project. In P.M.A. Sloot et al, editor, *Advances in Grid Computing — EGC 2005*, volume LNCS3470, Amsterdam, The Netherlands, Feb 2005. Springer.

[7] E. Kenny, B. Coghlan, J. Walsh, S. Childs, D. O’Callaghan, and G. Quigley. Autobuilding multiple ports of computing nodes for grid computing. In *Cracow Grid Workshop (CGW’05)*, Cracow, Poland, November 2005.

[8] Stuart Kenny. *An instrument monitoring framework for the Grid*. PhD thesis, Department of Computer Science, University of Dublin, Trinity College, 2006.

[9] R. Garcia Leiva, M. Barroso Lopez, G. Cancio Melia, B. Chardi Marco, L. Cons, P. Poznanski, A. Washbrook, E. Ferro, and A. Holt. Quattor: Tools and Techniques for the Configuration, Installation and Management of Large-Scale Grid Computing Fabrics. *Journal of Grid Computing*, 2(4), 2004.

[10] Antonio Delgado Peris, Patricia Mendez Lorenzo, Flavia Donno, Andrea Sciaba, Simone Campana, and Roberto Santinelli. LCG-2 user guide. LHC Computing Grid Manuals Series. <https://edms.cern.ch/file/454439//LCG-2-UserGuide.pdf>, September 2004.