Elsevier Future Generation Computer Systems 2006

Seamless Live Migration of Virtual Machines over the MAN/WAN

Franco Travostino, Paul Daspit, Leon Gommans, Chetan Jog, Cees de Laat,
Joe Mambretti, Inder Monga,Bas van Oudenaarde, Satish Raghunath, Phil Wang

Abstract

The "VM Turntable" demonstrator at iGRID 2005 pioneered the integration of Virtual Machines (VMs) with deterministic "lightpath" network services across MAN/WAN. The results provide for a new stage of virtualization—one for which computation is no longer localized within a data center but rather can be migrated across geographical distances, with negligible downtime, transparently to running applications and external clients. A noteworthy data point indicates that a live VM was migrated between Amsterdam, NL and San Diego, USA with just 1 to 2 seconds of application downtime. When compared to intra-LAN local migrations, downtime is only about 5-10 times greater despite 1,000 times higher round-trip-times.

Keywords: Virtualization, Virtual Machine, lightpaths, control planes, optical networks.

1 Introduction

Virtual Machines (VM) have long been a research topic in Computer Science because they hold a potential to accomplish tasks that are not dependent on particular hardware implementations or configurations. Recent advances [1] have shown that VM technology can support a fully featured operating system like Linux with a minimal runtime overhead when compared to native environments. However, to date VM instantiations still have to be localized at particular sites.

A capability for migrating live VMs among multiple distributed sites provides a significant new benefit for VM-based environments. In previous publications, Pratt et al. introduce the notion of live migration, as approximating continuous operation of a set of processes even while they are being sent to another location. [2] This process occurs in lieu of performing a VM-wide suspend, copy, and resume. The live migration minimizes the downtime, i.e., the time when no CPU cycle is devoted to any of the VM-resident applications, neither at the source nor at the destination system.

This paper describes the "VM Turntable" demonstrator, which shows that VMs can be migrated live, this time across geographical distances, transparently to applications and any external entity interacting with such applications. The demonstrator has been named "VM Turntable" to signify the recurring motion of VMs across sites. The migration process previously described [2] is shown to be ideally matched to the deterministic guarantees of the "lightpath" [3]—an end-to-end logical link in an optical physical network. This new approach contributes a

Franco Travostino, Paul Daspit, Chetan Jog, Inder Monga and Phil Wang ({travos, pdaspit, cjog, imonga, pywang}@nortel.com) are with Nortel Labs. Satish Raghunath was with Nortel Labs, he is now with Juniper Networks (rsatish@yahoo.com). Joe Mambretti (j-mambretti@northwestern.edu) is with International Center for Advanced Internet Research (iCAIR) at Northwestern University. Leon Gommans and Cees de Laat ({lgommans,delaat}@science.uva.nl)) are with the Advanced Internet Research group of the Universiteit van Amsterdam. Bas van Oudernaarde was with the University of Amsterdam, he is now with Finalist IT Group (bas@finalist.com).

.

novel perspective to the view of modern optical networks precisely supporting Grid operations [4].

The paper continues with the analysis of the processes that allow the live migration of VMs over long-haul networks Section 3 delineates the new set of requirements posed by such live migration. Section 4 makes the case for active orchestration of cpu, data, and network resources. The different functional entities in the VM Turntable demonstrator are delineated in Section 5. Section 6 describes the actual behavior of the iGRID 2005 demonstration. Furthermore, the Section reports on the characterization effort performed after iGRID 2005 in an equivalent lab setup. Section 7 singles out a number of lessons that were learned throughout the VM Turntable effort and the iGRID demonstration specifically.

2 Motivations for long-haul live migration

The capability to migrate live VMs across an enterprise or even across a Metropolitan Area Network (MAN) or a Wide Area Network (WAN)—from a handful of miles to planet-scale, transparently to running applications and external clients—meets the needs of many key case scenarios, especially when they are supported by high performance networks. These scenarios exhibit one or more of the following traits:

- It is impossible or impractical to bring the data (or devices) close to the computation engines. This may result from policy limitations (e.g., a data set that is embargoed from export) or capacity limitations (e.g., a data set is exceedingly large, thus adding an unwieldy preamble to the computation phase). An application's style of operation may exacerbate the latter characteristic (e.g., an application that infers from intermediate computation results which data sets are required next, lending itself to traveling salesman optimizations);
- It is desirable to load balance computation workloads with a scope that transcends the confines of individual data centers. One policy may dictate that during off hours the workload be consolidated into fewer data centers over a regional area to limit operating expenses or power consumption. Another policy might dictate that the workload "tracks the moon" to harvest spare computational power in different geographical areas that happen to be off business hours;
- It is required for operational business continuance, disaster recovery capabilities while meeting regulations for geographic diversity. The migration of VMs enables orderly evacuation of computation out of data centers that experience or anticipate failures, security compromises, or storms in their access network.

These benefits are in turn amplified by properties that are inherent to the realtime migration of a complete live VM and as such, the benefits are realized regardless of the distance of the migration, e.g., LAN, MAN, or WAN. Specifically, the migration of a VM is a powerful alternative to remote execution via Remote Procedure Call (RPC, [11]) or Globus Resource Allocation and Management (GRAM, [12]) when:

- There are issues concerning extending trust to a remote execution environment, which might have fallen victim of subtle exploitations by malware (e.g., Trojans);
- Interoperability at the level of Web Services, Java, etc. reveals gaps in versioning and standards maturity. On the other hand, the homogeneity of chipset and operating system warrants interoperating at the level of a virtual machine interface;

- The ensemble of applications (e.g., legacy applications) is such that it is impractical to sever inter-application synapses. The outright “forklifting” of one or more VMs is seen as an expedient way to transport such applications as a whole.

3 Long-haul live migration poses new requirements

In previous publications, the authors provide a quantitative characterization of live migration within a local sub-network. [2] The sub-network yields very short round trip times (RTT) and can operate with network endpoints that change physical coordinates while preserving the same address.

For a long-haul live migration to benefit the scenarios described, the downtime must be short and upper bound, in spite of RTTs that can be 1,000 times higher than the local sub-network case. The chance of a live migration being caught in a “world wide wait” syndrome would single handedly outweigh the benefits of a live migration. This is especially true for data centers with hundreds or thousands virtual machines lined up for an orderly migration over long-haul networks.

Upon migration, the long-haul scenario requires that the network endpoints be seamlessly adapted to the recipient environment, which, without any prior agreement, implements a different addressing schema than the sourcing environment. The adaptation of the endpoints must occur transparently to external clients interacting with the virtual machine being migrated. TCP sessions and higher-level sessions must not be restated. The conventional approach of leaving “breadcrumbs” at the sourcing environment for subsequent forwarding is inadequate because of the high RTTs and potential store-and-forward handling of virtual machines resulting in long forwarding chains.

In many circumstances, the recipient environment is thought to yield equivalent or superior access to the working data set that is required by the virtual machine. What about the intermediate state? Given the elevated RTTs, it would be undesirable for a virtual machine to access its intermediate state at the sourcing environment. The migration process must, therefore, be amenable to packing and transfering more state—specifically, state stored in hard disk— than just memory pages.

A long-haul migration occurs across multiple domains—e.g., sourcing domain, recipient domain, and one or more intervening network domains—with limited trust among them. Each of the domains and their connections are vulnerable to security exploitations. For example, a virtual machine can be compromised by an insider subverting the recipient domain. It can be hijacked completely into a faked recipient domain. Since the migration entails pre-allocation of resources, there is a risk of resource theft disrupting the migration as well as the remote execution.

4 The case for an integrated orchestration of cpu, data, and networks

This complex set of requirements requires an agent that is trusted to allocate different types of resources and coordinate them with the live migration. It would not be possible to plan for peak-provisioning of resources while also planning for all migration permutations.

To provide for computational resources, the agent must broker an environment that provides both CPU cycles and memory ahead of the migration. In a typical site policy, for instance, preference will be given to those systems with under-subscribed hyper-threads.

.

To provide for network resources, the agent must support the migration with a highly deterministic end-to-end network path. For example, the agent can broker an end-to-end service that closely approximates a circuit of highly predictable performance for the short duration of the live migration(s). Once the migration is completed, such service could be torn down or re-cast to a lesser service to allow for a trickle of remote data references.

To provide for data, the agent has the dual role of information directory service and replica manager. The former locates data whereas the latter migrates data as an alternative or complement to the live migration of virtual machines.

The 2005 VM Turntable demonstrator emphasizes the coordinated brokering of computational resources and network resources.

5 The 2005 VM Turntable demonstrator

The theme chosen for VM Turntable demonstrator is a case scenario in which an application suite performs a "needle in the haystack" iterative search workflow for a pattern hiding in digital images. The images are stored in three unique sets at Amsterdam, Chicago, and San Diego. The demonstrator features the live migration of Xen virtual machines loaded with the environment and executables for such search applications. The live migration allows the applications to move close to the image sets and thus access the "right" data at the "right" time with the best access latency possible.

The live migration occurs over dynamically established, dedicated lightpaths—circuits of 1 Gbps capacity between distant sites like Amsterdam and San Diego. The lightpaths were comprised exclusively of L1 and L2 circuits, without any routed paths. By avoiding layer 3 routing, the demonstrator steered clear of any potential problem introduced by delays in packet forwarding.

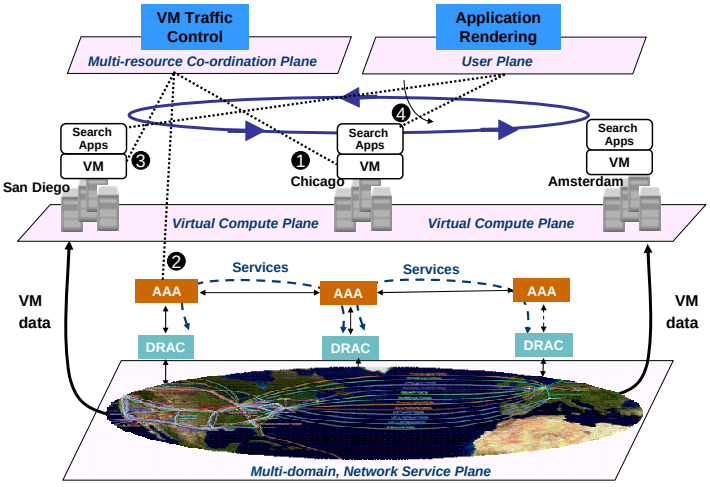

Figure 1. The 2005 VM Turntable Demonstrator

The VM Turntable demonstrator builds upon the following entities:

5.1 VM "Traffic Controller"

The VM Traffic Controller (VMTC) embodies the agent described in the previous section. It interfaces and co-ordinates the migration of VMs with provisioning of network resources and the re-provisioning of the IP tunnel to ensure seamless layer 3-7 connectivity to the applications inside of the VM container.

First, the VMTC maintains connectivity and state information with the site that is the target recipient of the VM migration (step 1 in Figure 1). Second, The VMTC communicates with the Authentication, Authorization, and Accounting (AAA) [5] entity and requests pre-authorization to use a lightpath for the VM migration (step 2 in Figure 1). The authorization is returned in form of a "token". The VMTC then uses the token and migration parameters to setup an on-demand, end-to-end, authenticated setup request for a lightpath. Third, upon receiving confirmation on the existence of the lightpath, the VMTC issues the migrate command (step 3 in Figure 1). Lastly, VMTC reconfigures the IP tunnel giving the external clients a seamless connectivity to the applications running in the migrating VM (step 4 in Figure 1).

5.2 AAA

The AAA agent governs the pre-allocation of extra-VM resources required for the migration. It is appropriate to consider the AAA function as one that extends the VM's sandbox capabilities to the environment outside of the VM. AAA extends a sandbox in space—i.e., to all intervening domains between migration source and destination—and in time—i.e., for the duration of the migration. To accomplish this objective, the AAA agent peers with AAA implementations in other domains, as described in [6]. AAA supports the formulation of complex authorization policies.

5.3 Token-based security

A novel authorization message sequence [7] involving a token has been devised to thwart resource theft. Specifically, the token prevents an unauthorized user from accessing pre-allocated resources such as the network lightpath. The token represents a crypto-strong unforgeable capability. It can be implemented either out-of-band (i.e., it is propagated at the control plane level) or in-band (i.e., it is interspersed within the data payload). The 2005 VM Turntable demonstrator featured the out-of-band version.

5.4 DRAC

Nortel's DRAC (Dynamic Resource Allocation Controller) [8] agent sets up the end-to-end lightpaths required by the VM migration. DRAC exposes a service-oriented API for the coupling with applications. Using Web Services exchanges, VMTC requests DRAC a "cut-through" network service, which is a high bandwidth, low latency service that bypasses layer 3 and directly transfers data over layer 1 or layer 2 connections. VMTC and other clients can further specify whether they want the service on demand or via a time-of-day reservation.

One or more instances of DRAC software runs in each participating network domain. In case multiple instances of DRAC are running in a single domain, a master instance is selected. The DRAC master instance manages the domain and the inter-domain connectivity through peer messaging and close interaction with the AAA component. The DRAC core framework features a policy engine, a topology and discovery engine, workflow utilities, inter-domain routing facilities, and dynamic bandwidth management fixtures. To steer network resources, DRAC binds to network control planes or individual network elements (e.g., via SNMP or TL1 signaling).

.

5.5 Preservation of TCP and higher-level sessions

The long-haul migration requirements require a layer-3 mechanism that handles connectivity to the virtual machine, while not relying on the usual routing infrastructure. If the routing infrastructure were to be used, the virtual machine would have to acquire an IP address dictated by the network in which it is currently executing—a choice that would break active TCP sessions with clients of the applications on the virtual machine. Rather, the virtual machine maintains the same IP address(es) across migrations.

To accomplish this, dynamically configured IP tunnels allow client connectivity to the virtual machine as it migrates across network domains. The IP tunnels are reconfigured with each migration event to retain layer-3 connectivity. These reconfigurations are invisible to TCP or any other higher-layer session since the IP address itself remains unchanged.

The reconfiguration of the IP tunnel is shown in Figure 2 using a simple scenario consisting of two sites between which the virtual machine is migrated. The two sites feature different and global network addresses (12.x.x.x and 13.x.x.x). The virtual machine itself is assigned a fictitious address (10.1.1.2). It communicates with the external world through a "virtual gateway interface" (10.1.1.1) which is created by the Xen control plane when the VM is started. The element labeled "Visualization host" indicates the client which retrieves the results of an application that is running on the virtual machine and could be located in a third location unrelated to either of the sites that host the virtual machine (e.g., an end-user's premises).

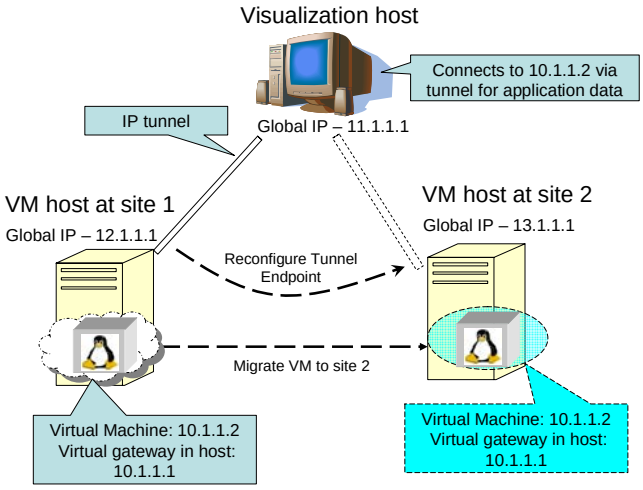

Figure 2. Tunnel migration for seamless external connectivity

5.6 The Xen virtual environment

The VM Traffic Controller issues migration commands to the Xen environment—an unmodified 2.0.7 configuration, with the sole exception of patches to support jumbo frames. The live option for the migration is a key asset in integrating VM migration with the lighpath's high bandwidth delay product.

5.7 State stored on hard disk

The high latency experienced in long-haul migration motivates a policy preventing inspection of previous instantiations, a policy that extends to any and all VM states. Ideally, the state on hard disk ought to be handled the same way and at the same time the memory pages are being transferred. In one particular design, this goal can be accomplished by means of the snapshot capability of a Logical Volume Manager. [9] Its built-in copy-on-write logic for disk blocks

enables a promising delta protocol for those disk contents that mutate while the live migration is in progress. This feature was not implemented on time for iGRID 2005.

6 Experimental results

The iGRID 2005 demonstration showcased a reasonable subset of the VM Turntable demonstrator. It ran over a network layout featuring two sites, Amsterdam and San Diego, connected via a 1 Gbps non-switched link with RTT of 198 milliseconds (ms). The software layers implemented the entities described in Section 5 and depicted in Figure 1, with the sole exclusion of the logic to migrate accrued state off of the hard disk. Because the layout did not have a switching point, the DRAC capability to re-provision the network in an automated fashion was not exercised.

The application downtime caused by a live migration was measured as an interval of 0.8-1.6 seconds. When compared to the downtime seen in the intra-LAN setup, it appears that downtime had only grown 5-10 times despite a RTT that is 1,000 times higher.

The low precision and high variance of the measurements made at iGRID 2005 required follow up testing. A chief goal was to remove any dependency on long-range clock synchronization while applying lab-quality instrumentation of 1 ms or less known precision. To accomplish this, the iGRID 2005 demonstrator was re-hosted such that the University of Amsterdam lab sourced and sunk the VM after the VM had traveled over a long loop (a 1 Gbps non-switched Amsterdam to New York to Amsterdam service), with a RTT (175 ms) comparable to the one observed over the Amsterdam – San Diego distance. Source and sink nodes were equipped with Intel Xeon CPU 2.80GHz and 1GB memory. They ran Linux 2.6.11 and Xen 2.0.7, with a patch to correctly handle large MTU (8174 bytes). They ran two Xen domains each, a parent domain and a guest domain. The parent domain "dom0" ran with a preset 256MB memory. The guest domain "domU" ran the demo application for face detection, which continuously looped through images fetched from storage. It had a preset 200MB memory. All the migrations were tested with the "live" option unless stated otherwise.

6.1 Application-level—internal downtime measurement

This experiment measured the downtime from within the VM being migrated. Results are shown in Table 1. An ad-hoc measurement program running in "domU" acquired the current time with a fixed period and computes the time difference since last time recorded. The period was set to 0.2 ms for better accuracy. It produced a downtime report when it detected an interruption time over 50 ms.

Table 1: Application-level internal measurement of downtime

Downtime (ms)	Without background image processing	With image processing
Min	279	1877
Max	349	2045
Mean	315	1939

6.2 Application-level—external downtime measurement

In this experiment, the downtime was measured via a high precision oscilloscope that monitored the signals on the serial port of the two systems involved in the VM migration. Results are shown in Table 2. A simple signaling program between domU and dom0 was put in place to overcome a current limitation preventing applications in domU from accessing the TTY driver directly. The

.

TTY rates and application-level signaling rates were chosen such that the systems are known not to buffer any character.

Table 2: Application-level external measurement of downtime

Downtime (ms)	Without background image processing	With image processing
Min	198	1680
Max	280	2120
Mean	230	1940

6.3 Network-level downtime measurement

This experiment showed how an external client application that interacts with the guest domain in a transaction style would be affected by the downtime of the guest domain. The command "ping" was used to test the network accessibility of the guest domain, with a 50 ms interval between ping requests. Results are shown in Table 3. During a migration, 76 ping packets did not produce any reply. The first ping after migration had an RTT of 10 ms. Thus, the total downtime perceived by the client application was = 76 * 50 + 10 – 0.621 = 3809 ms.

Table 3: Ping Response Times

Time (ms)	Normal	With Migration
Min	0.030	0.03
Max	0.141	3810
Mean	0.053	0.621

6.4 Live vs. non-live ratio

Throughout the aforementioned experiments, the elapsed time for a non-live migration—i.e., through a VM-wide suspend, move, resume— turned out to be 50 to 100 times larger than the time required by a live migration. This data point confirmed that the pre-copy algorithm, noted previously [2], was ideally suited to the WAN network put in place for the VM Turntable demonstrator.

7 Lessons learned at iGRID 2005

The demonstration showed that the live migration of a Xen virtual machine across geographical distances can be completed with a 1-2 second downtime, with much anticipated variance on the application workload. The sophisticated pipelining of the migration steps previously described [2] proved to be an excellent match with the deterministic performance guarantees of the lightpath network service.

The same pipelining—specifically, the iterative pre-copy of memory pages while applications are still running—avoids the negative impact of TCP's slow start over a channel with a remarkably high bandwidth delay product. Over such a link, it is well known that TCP requires several seconds to open up its window to a useful value. In the demonstrator, this sub-optimal training period affects the pre-copy phase only. The applications and their actual downtime are not impacted. The lightpath otherwise proved itself highly reliable for this demonstration. It is thus quite unlikely that TCP would enter an AIMD state upon packet loss at the very critical time when the downtime would be affected. Another implication of this result is that substituting "enhanced TCP stacks"—non-standard TCP implementations or TCP alternatives (e.g., as noted [10])—would add no value.

A key observation is that the lightpath service yielded the required predictable performances. Predictability was quite good even when the lightpath was carried on a lightly-loaded layer 2 network. For lifecycle control of the lightpath, the service-oriented software layers (i.e. DRAC and AAA) provided powerful and flexible intermediation. This level of high-quality predictable performance was essential to ensuring the success of the demonstration.

Although the demonstration of the migrated applications clearly supported the use case chosen, it failed to give attendees a sense of the actual downtime, in part, because the downtime segment was so brief. Furthermore, the remote rendering of the images was an aperiodic task with update intervals greater than the downtime itself. It would have been best to plot a periodic curve (e.g., a sinus function, a radar sweep) and show a sector missing as a consequence of downtime.

The measurement of actual downtime proved to be an elusive task, especially in terms of clock synchronization and probes to start/stop the downtime stopwatch. The post iGRID 2005 experiments detailed in Section 6 provided the missing lab-quality characterization while preserving the network behavior of a large-scale WAN experiment.

8 Conclusions and Future Work

The integration of VMs and deterministic lightpath network services creates the VM Turntable demonstrator, an environment wherein a VM is no longer limited to execute within the confines of a data center. Rather, the live VM can be migrated across distances, securely, with near second downtime and with no impact on applications or external observers. The demonstrator features computation engines that can be dynamically deployed closest to working data sets, as a powerful alternative to migrating data near fixed computation sites. No longer restricted to LANs, the demonstrator is an alternative to other global techniques for remote execution such as RPC [11] and GRAM [12].

After the first demonstration at iGRID 2005, the VM Turntable demonstrator underwent lab characterization efforts, also described in this paper. In subsequent phases, the demonstrator will be augmented with a Web Services representation of the migratable VM resource, for the VM to be fully integrated with other Grid managed resources. To further minimize the impact of migration, a fabric technology capable of Remote Direct Memory Access (RDMA) like Infiniband [13] can terminate the lightpath and optimize the copy of memory pages during migration. As such, it realizes an ideal extension of the lightpath directly into memory.

Acknowledgements

Franco Travostino invented the VM Turntable demonstrator after becoming aware of the capabilities in [1] and [2]. Furthermore, Franco is grateful to Ian Pratt for his advices on how to quantify the impact of a VM live migration. The authors are indebted to the iGRID 05 organizers, Rodney Wilson, Paola Grosso, Pieter De Boer, Fei Yeh for their dedicated support to making the VM Turntable a successful demonstration.

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the art of virtualization, Proceedings of the nineteenth ACM symposium on Operating Systems (SOSP19), October 2003.

.

[2] C. Clark, K. Fraser, S. Hand, J.G. Hansem, E. Jul, C. Limpach, I. Pratt, A. Warfield, Live Migration of Virtual Machines, 2nd Symposium on Networked Systems Design and Implementation (NSDI) 2005, May 2005.

[3] T. DeFanti, C. de Laat, J. Mambretti, K. Neggers, B. St. Arnaud, TransLight, a Global-scale LambdaGrid for e-Science, Communications of the ACM, Volume 46, Number 11, November 2003.

[4] Cees de Laat, Erik Radius, Steven Wallace, "The Rationale of the Current Optical Networking Initiatives", iGrid2002 special issue, Future Generation Computer Systems, volume 19 issue 6 (2003).

[5] Vollbrecht J., Calhoun P., Farrell S., Gommans L., Gross G., de Bruijn B., de Laat C., Holdrege M., Spence D., "AAA Authorization Framework", Internet Engineering Task Force RFC2904, August 2000.

[6] L. Gommans, B. van Oudenaarde, F. Dijkstra, C. de Laat, T. Lavian, I. Monga, A. Taal, F. Travostino, A. Wan, Applications Drive Secure Lightpath Creation across Heterogeneous Domains, to appear in IEEE Communications Magazine, Feature topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision.

[7] L. Gommans, F. Travostino, J. Vollbrecht, Cees de Laat, R. Meijer, Token-Based Authorization of Connection Oriented Network Resources, 1st International Workshop on Networks for Grid Applications (GridNets 2004), October 29th 2004.

[8] The Dynamic Resource Allocation Controller, DRAC, www.nortel.com/drac.

[9] The Linux Logical Volume Manager, http://tldp.org/HOWTO/LVM-HOWTO/.

[10] Y. Gu and R. Grossman, UDT: An Application Level Transport Protocol for Grid Computing, Second International Workshop on Protocols for Fast Long-Distance Networks, February 2004.

[11] Birrel, A. D. and B. J. Nelson, Implementing remote procedure calls, ACM Transactions on Computer Systems 2 (1984), pp. 39—59.

[12] I. Foster, Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005.

[13] The Infiniband Trade Alliance, http://www.infinibandta.org/home.