

Live Data Center Migration across WANs: A Robust Cooperative Context Aware Approach

K.K. Ramakrishnan, Prashant Shenoy[†], Jacobus Van der Merwe
AT&T Labs-Research / [†]University of Massachusetts

ABSTRACT

A significant concern for Internet-based service providers is the continued operation and availability of services in the face of outages, whether planned or unplanned. In this paper we advocate a cooperative, context-aware approach to data center migration across WANs to deal with outages in a non-disruptive manner. We specifically seek to achieve high availability of data center services in the face of both planned and unanticipated outages of data center facilities. We make use of *server* virtualization technologies to enable the replication and migration of server functions. We propose new *network* functions to enable server migration and replication across wide area networks (e.g., the Internet), and finally show the utility of intelligent and dynamic *storage* replication technology to ensure applications have access to data in the face of outages with very tight recovery point objectives.

Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed Systems

General Terms

Design, Reliability

Keywords

Data Center Migration, Virtual Server, Storage

1. INTRODUCTION

A significant concern for Internet-based service providers is the continued operation and availability of services in the face of outages, whether planned or unplanned. These concerns are exacerbated by the increased use of the Internet for mission critical business and real-time entertainment applications. A relatively minor outage can disrupt and inconvenience a large number of users. Today these services are almost exclusively hosted in data centers.

Recent advances in server virtualization technologies [8, 14, 22] allow for the **live** migration of services within a local area network

(LAN) environment. In the LAN environment, these technologies have proven to be a very effective tool to enable data center management in a non-disruptive fashion. Not only can it support planned maintenance events [8], but it can also be used in a more dynamic fashion to automatically balance load between the physical servers in a data center [22]. When using these technologies in a LAN environment, services execute in a virtual server, and the migration services provided by the underlying virtualization framework allows for a virtual server to be migrated from one physical server to another, without any significant downtime for the service or application. In particular, since the virtual server retains the same network address as before, any ongoing network level interactions are not disrupted. Similarly, in a LAN environment, storage requirements are normally met via either network attached storage (NAS) or via a storage area network (SAN) which is still reachable from the new physical server location to allow for continued storage access.

Unfortunately in a wide area environment (WAN), live server migration is not as easily achievable for two reasons: First, live migration requires the virtual server to maintain the same network address so that from a network connectivity viewpoint the migrated server is indistinguishable from the original. While this is fairly easily achieved in a shared LAN environment, no current mechanisms are available to efficiently achieve the same feat in a WAN environment. Second, while fairly sophisticated remote replication mechanisms have been developed in the context of disaster recovery [20, 7, 11], these mechanisms are ill suited to live data center migration, because in general the available technologies are unaware of application/service level semantics.

In this paper we outline a design for live service migration across WANs. Our design makes use of existing server virtualization technologies and propose network and storage mechanisms to facilitate migration across a WAN. The essence of our approach is cooperative, context aware migration, where a migration management system orchestrates the data center migration across all three subsystems involved, namely the server platforms, the wide area network and the disk storage system. While conceptually similar in nature to the LAN based work described above, using migration technologies across a wide area network presents unique challenges and has to our knowledge not been achieved. Our main contribution is the design of a framework that will allow the migration across a WAN of *all* subsystems involved with enabling data center services. We describe new mechanisms as well as extensions to existing technologies to enable this and outline the cooperative, context aware functionality needed across the different subsystems to enable this.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

INM'07, August 27–31, 2007, Kyoto, Japan.

Copyright 2007 ACM 978-1-59593-788-9/07/0008 ...\$5.00.

2. LIVE DATA CENTER MIGRATION ACROSS WANS

Three essential subsystems are involved with hosting services in a data center: First, the *servers* host the application or service logic. Second, services are normally hosted in a data center to provide shared access through a *network*, either the Internet or virtual private networks (VPNs). Finally, most applications require disk *storage* for storing data and the amount of disk space and the frequency of access varies greatly between different services/applications.

Disruptions, failures, or in general, outages of any kind of any of these components will cause service disruption. For this reason, prior work and current practices have addressed the robustness of individual components. For example, data centers typically have multiple network connections and redundant LAN devices to ensure redundancy at the networking level. Similarly, physical servers are being designed with redundant hot-swappable components (disks, processor blades, power supplies etc). Finally, redundancy at the storage level can be provided through sophisticated data mirroring technologies.

The focus of our work, however, is on the case where such *local* redundancy mechanisms are not sufficient. Specifically, we are interested in providing service availability when the data center as a whole becomes unavailable, for example because of data center wide maintenance operations, or because of catastrophic events. As such, our basic approach is to migrate services between data centers across the wide area network (WAN).

By necessity, moving or migrating services from one data center to another needs to consider all three of these components. Historically, such migration has been disruptive in nature, requiring downtime of the actual services involved, or requiring heavy weight replication techniques. In the latter case concurrently running replicas of a service can be made available thus allowing a subset of the service to be migrated or maintained without impacting the service as a whole. We argue that these existing mechanisms are inadequate to meet the needs of network-based services, including real-time services, in terms of continuous availability and operation. Instead, we advocate an approach where server, network and storage subsystems cooperate and coordinate actions, in a manner that is cognizant of the service context in order to realize seamless migration across wide area networks.

In this section we briefly describe the technical building blocks that would enable our approach. As outlined below, some of these building blocks exist, or exist in part, while in other cases we use the desire for high availability of services as the driver for the changes we are proposing.

2.1 Live Virtual Server Migration

The main enabler for our approach is the live server migration capabilities that have been developed in the context of server virtualization in recent years [5, 8]. In this approach an entire running operating system (including any active applications) executing as a virtual server is being transferred from one physical machine to another. Since the virtual server is migrated in its entirety, both application and kernel level state gets migrated, including any state associated with ongoing network connections. Assuming that network level reachability to the virtual server's network addresses are maintained after the migration, the implication is that applications executing in the virtual server experience very little downtime (in the order of tens to hundreds of milliseconds) and ongoing network connections remain intact.

In order to maintain network level reachability, the IP address(es)

associated with the virtual server has to be reachable at the physical server where the virtual server is migrated to. In a LAN environment this is achieved either by issuing an unsolicited ARP reply to establish the binding between the new MAC address and the IP address, or by relying on layer-two technologies to allow the virtual server to reuse its (old) MAC address [8].

Because of the difficulty of moving network level (i.e., IP addresses) in a routed non-LAN environment, use of live server migration as a management tool has been limited to the LAN environments [22]. However, virtual server migration across the wide area will also be an attractive tool, specifically to deal with outages, and therefore propose networking mechanisms to enable this.

If disk storage needs are being met with network attached storage (NAS), the storage becomes just another network based application and can therefore be addressed in the same way with LAN based migration [8]. Modern virtualization environments also include support for other forms of (local) storage including storage area networks (SANs) [23]. However, since we propose to use WAN server migration as a means to deal with complete data center outages, these mechanisms are inadequate for our purposes and below we propose extension to remote replication technologies which can work in concert with server migration to minimize service downtime.

2.2 Networking Requirements

From the discussion above, a key requirement for live server migration across a WAN is the ability to have the IP address(es) of the virtual server be reachable at the new data center location immediately after the migration has completed. This presents a significant challenge for a number of reasons. First, despite decades of work in this area, IP address mobility remains an unresolved problem that is typically only addressed at manual configuration time scales. The second challenge comes from the fact that current routing protocols are well known to have convergence issues which is ill suited to the time constraints imposed by live migration. Third, in today's WAN networking environment connectivity changes are typically initiated, and controlled, by network operators or network management systems. Again this is poorly suited to WAN server migration where it is essential that the migration software, which is closely monitoring the status of the server migration process, initiate this change at the appropriate time.

Our approach to addressing the networking requirements for live WAN migration builds on the observations that not all networking changes in this approach are time critical and further that instantaneous changes are best achieved in a localized manner. Specifically, in our solution, described in detail in Section 3, we allow the migration software to initiate the necessary networking changes as soon as the need for migration has been identified. We make use of tunneling technologies during this initial phase to preemptively establish connectivity between the data centers involved. Once server migration is complete, the migration software initiates a local change to direct traffic towards the new data center via the tunnel. Slower time scale network changes then phase out this local network connectivity change for a more optimal network wide path to the new data center.

2.3 Storage Replication Requirements

Data availability is typically addressed by replicating business data on a local/primary storage system, to some remote location from where it can be accessed. From a business/usability point of view, such remote replication is driven by two metrics [9]. First

is the *recovery-point-objective* which is the consistent data point to which data can be restored after a disaster. Second is the *recovery-time-objective* which is the time it takes to recover to that consistent data point after a disaster [13].

Remote replication can be broadly classified into the following two categories:

- Synchronous replication: every data block written to a local storage system is replicated to the remote location before the local write operation returns.
- Asynchronous replication: in this case the local and remote storage systems are allowed to diverge. The amount of divergence between the local and remote copies is typically bounded by either a certain amount of data, or by a certain amount of time.

Synchronous replication is normally recommended for applications, such as financial databases, where consistency between local and remote storage systems is a high priority. However, these desirable properties come at a price. First, because every data block needs to be replicated remotely, synchronous replication systems can not benefit from any local write coalescing of data if the same data blocks are written repeatedly [16]. Second, because data have to be copied to the remote location before the write operation returns, synchronous replication has a direct performance impact on the application, since both lower throughput and increased latency of the path between the primary and the remote systems are reflected in the time it takes for the local disk write to complete.

An alternative is to use asynchronous replication. However, because the local and remote systems are allowed to diverge, asynchronous replication always involves some data loss in the event of a failure of the primary system. But, because write operations can be batched and pipelined, asynchronous replication systems can move data across the network in a much more efficient manner than synchronous replication systems.

For WAN live server migration we seek a more flexible replication system where the mode can be dictated by the migration semantics. Specifically, to support live server migration we propose a remote replication system where the initial transfer of data between the data centers is performed via asynchronous replication to benefit from the efficiency of that mode of operation. When the bulk of the data have been transferred in this manner, replication switches to synchronous replication in anticipation of the completion of the server migration step. The final server migration step triggers a simultaneous switch-over to the storage system at the new data center. In this manner, when the virtual server starts executing in the new data center, storage requirements can be locally met.

3. WAN MIGRATION SCENARIOS

In this section we illustrate how our cooperative, context aware approach can combine the technical building blocks described in the previous section to realize live server migration across a wide area network. We demonstrate how the coordination of server virtualization and migration technologies, the storage replication subsystem and the network can achieve live migration of the entire data center across the WAN. We utilize different scenarios to demonstrate our approach. In Section 3.1 we outline how our approach can be used to achieve the safe live migration of a data center when planned maintenance events are handled. In Section 3.2 we show the use of live server migration to mitigate the effects of unplanned outages or failures.

3.1 Maintenance Outages

We deal with maintenance outages in two parts. First, we consider the case where the service has no (or very limited) storage requirements. This might for example be the case with a network element such as a voice-over-IP (VoIP) gateway. Second, we deal with the more general case where the service also requires the migration of data storage to the new data center.

Without Requiring Storage to be Migrated: Without storage to be replicated, the primary components that we need to coordinate are the server migration and network “mobility”. Figure 1 shows the environment where the application running in a virtual server “VS” has to be moved from a physical server in data center A to a physical server in data center B.

Prior to the maintenance event, the coordinating “migration management system” (MMS) would signal to both the server management system as well as the network that a migration is imminent. The server management system would initiate the migration of the virtual server from physical server “a” (PS_a) to physical server “b” (PS_b). After an initial bulk state transfer as “preparation for migration”, the server management system will mirror any state changes between the two virtual servers.

Similarly, for the network part, based on the signal received from the MMS, the service provider edge (PE) router will initiate a number of steps to prepare for the migration. Specifically, as shown in Figure 1(b), the migration system will cause the network to create a tunnel between PE_a and PE_b which will be used subsequently to transfer data destined to VS to data center B.

When the MMS determines a convenient point to quiesce the VS, another signal is sent to both the server management system and the network. For the server management system, this signal will indicate the final migration of the VS from data center A to data center B, i.e., after this the VS will become active in data center B. For the network, this second signal enables the network data path to switchover locally at PE_a to the remote data center. Specifically, from this point in time, any traffic destined for the virtual server address that arrives at PE_a will be switched onto the tunnel to PE_b for delivery to data center B.

Note that at this point, from a server perspective the migration is complete as the VS is now active in data center B. However, traffic is sub-optimally flowing first to PE_a and then across the tunnel to PE_b . To rectify this situation another networking step is involved. Specifically, PE_b starts to advertise a more preferred route to reach VS, than the route currently being advertised by PE_a . In this manner, as ingress PEs to the network (PE_c to PE_e in Figure 1) receive the more preferred route, traffic will start to flow to PE_b directly and the tunnel between PE_a and PE_b can be torn down leading to the final state shown in Figure 1(c).

Requiring Storage Migration: When storage has to also be replicated, it is critical that we achieve the right balance between performance (impact on the application) and the recovery point or data loss when the switchover occurs to the remote data center. To achieve this, we allow the storage to be replicated asynchronously, prior to any initiation of the maintenance event, or, assuming the amount of data to be transferred is relatively small, asynchronous replication can be started in anticipation of a migration that is expected to happen shortly. Asynchronous replication during this initial phase allows for the application to see no performance impact. However, when the maintenance event is imminent, the MMS would signal to the replication system to switch from asynchronous replication to synchronous replication to ensure that there is no loss of data during migration. When data is being replicated synchronously, there will be a performance impact on the application.

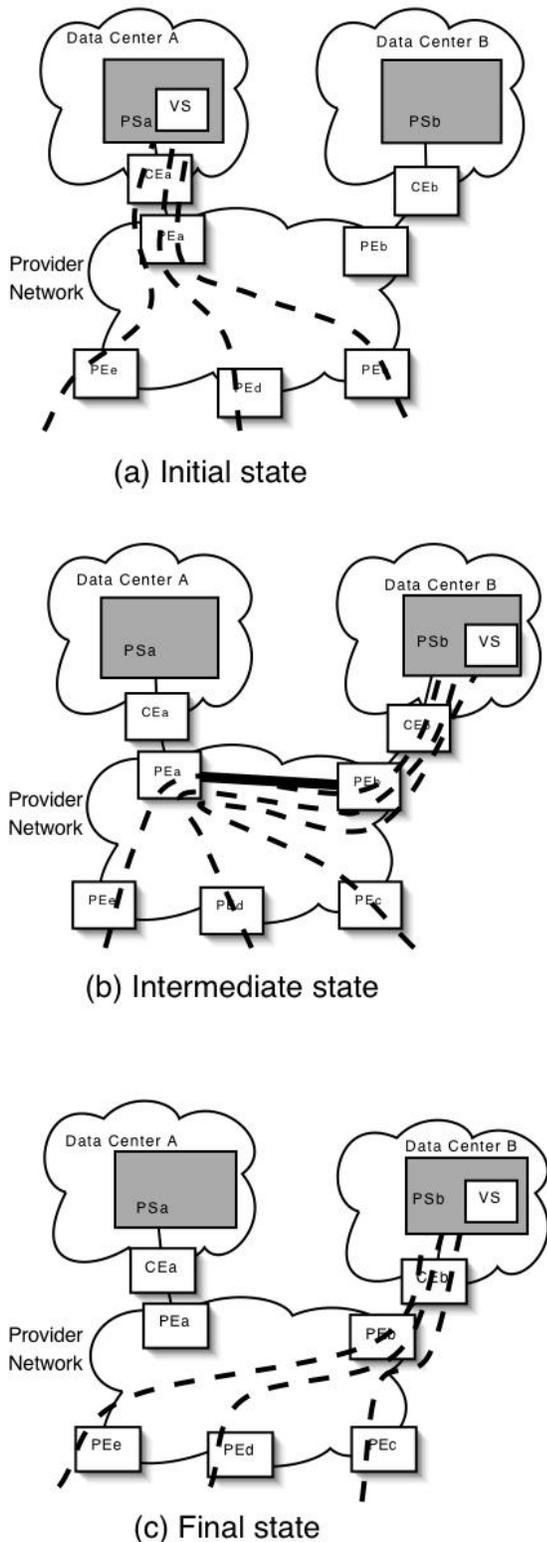


Figure 1: Live server migration across a WAN

This requires us to keep the exposure to the amount of time we replicate on a synchronous basis to a minimum.

When the MMS signals to the storage system the requirement to switch to synchronous replication, the storage system completes all the pending asynchronous operations and then proceeds to perform all the subsequent writes by synchronously replicating it to the remote data center. Thus, between the server migration and synchronous replication, both the application state and all the storage operations are mirrored at the two environments in the two data centers. When all the pending write operations are copied over, then as in the previous case, we quiesce the application and the network is signaled to switch traffic over to the remote data center. From this point on, both storage and server migration operations are complete and activated in data center B. As above, the network state still needs to be updated to ensure optimal data flow directly to data center B.

Note that while we have described the live server migration process as involving the service provider for the networking part, it is possible for a data center provider to perform a similar set of functions without involving the service provider. Specifically, by creating a tunnel between the customer edge (CE) routers in the data center, and performing local switching on the appropriate CE, rather than on the PE, the data center provider can realize the same functionality.

3.2 Unplanned Outages

We propose to also use cooperative, context aware migration to deal with unplanned data center outages. There are multiple considerations that go into managing data center operations to plan and overcome failures through migration. Some of these are: (1) amount of overhead under normal operation to overcome anticipated failures; (2) amount of data loss affordable (recovery point objective - RPO); (3) amount of state that has to be migrated; and (4) time available from anticipated failure to occurrence of event.

At the one extreme, one might incur the overhead of completely mirroring the application at the remote site. This has the consequence of both incurring processing and network overhead under normal operation as well as impacting application performance (latency and throughput) throughout. The other extreme is to only ensure data recovery and to start a new copy of the application at the remote site after an outage. In this case, application memory state such as ongoing sessions are lost, but data stored on disk is replicated and available in a consistent state. Neither this hot standby nor the cold standby approach described are desirable due to the overhead or the loss of application memory state.

An intermediate approach is to recover control and essential state of the application, in addition to data stored on disk, to further minimize disruptions to users. A spectrum of approaches are possible. In a VoIP server, for instance, session-based information can be mirrored without mirroring the data flowing through each session. More generally, this points to the need to checkpoint some application state in addition to mirroring data on disk. Checkpointing application state involves storing application state either periodically or in an application-aware manner like databases do and then copying it to the remote site. Of course, this has the consequence that the application can be restarted remotely at the checkpoint boundary only. Similarly, for storage one may use asynchronous replication with a periodic snapshot ensuring all writes are up-to-date at the remote site at the time of checkpointing. Some data loss may occur upon an unanticipated, catastrophic failure, but the recovery point may be fairly small, depending on the frequency of checkpointing application and storage state. Coordination between

the checkpointing of the application state and the snapshot of storage is key to successful migration while meeting the desired RPOs. Incremental checkpointing of application and storage is key to efficiency, and we see existing techniques to achieve this [4, 3, 11]. For instance, rather than full application mirroring, a virtualized replica can be maintained as a “warm standby”—in dormant or hibernating state—enabling a quick switch-over to the previously checkpointed state. To make the switch-over seamless, in addition to replicating data and recovering state, network support is needed. Specifically, on detecting the unavailability of the primary site, the secondary site is made active, and the same mechanism described in Section 3.1 is used to switch traffic over to reach the secondary site via the pre-established tunnel. Note that for simplicity of exposition we assume here that the PE that performs the local switch over is not affected by the failure. The approach can however, easily be extended to make use of a switchover at a router “deeper” in the network.

The amount of state and storage that has to be migrated may vary widely from application to application. There may be many situations where, in principle, the server can be stateless. For example, a SIP proxy server may not have any persistent state and the communication between the clients and the proxy server may be using UDP. In such a case, the primary activity to be performed is in the network to move the communication over to the new data center site. Little or no overhead is incurred under normal operation to enable the migration to a new data center. Failure recovery involves no data loss and we can deal with near instantaneous, catastrophic failures.

As more and more state is involved with the server, more overhead is incurred to checkpoint application state and potentially to take storage snapshots, either periodically or upon application prompting. It also means that the RPO is a function of the interval between checkpoints, when we have to deal with instantaneous failures. The more advanced information we have of an impending failure, the more effective we can be in having the state migrated over to the new data center, so that we can still have a tighter RPO when operations are resumed at the new site.

4. RELATED WORK

Prior work on this topic falls into several categories: virtual machine migration, storage replication and network support.

At the core of our technique is the ability of encapsulate applications within virtual machines that can be migrated without application downtimes [15]. Most virtual machine software, such as Xen [8] and VMWare [14] support “live” migration of VMs that involve extremely short downtimes ranging from tens of milliseconds to a second; details of Xen’s live migration techniques are discussed in [8]. As indicated earlier, these techniques assume that migration is being done on a LAN. VM migration has also been studied in the Shirako system [10] and for grid environments [17, 19].

Current virtual machine software support a suspend and resume feature that can be used to support WAN migration, but with downtimes [18, 12]. Recently live WAN migration using IP tunnels was demonstrated in [21], where an IP tunnel is set up from the source to destination server to transparently forward packets to and from the application; we advocate an alternate approach that assumes edge router support.

In the context of storage, there exist numerous commercial products that perform replication, such as IBM Extended Remote Copy, HP Continuous Access XP, and EMC RepliStor. An excellent description of these and others, as well as a detailed taxonomy of the different approaches for replication can be found in [11]. The Ursa

Minor system argues that no single fault model is optimal for all applications and proposed supporting data-type specific selections of fault models and encoding schemes for replication [1]. Recently, we proposed the notion of semantic-aware replication [13] where the system supports both synchronous and asynchronous replication concurrently and use “signals” from the file system to determine whether to replicate a particular write synchronously and asynchronously.

In the context of network support, our work is related to the RouterFarm approach [2], which makes use of orchestrated network changes to realize near hitless maintenance on provider edge routers. In addition to being in a different application area, our approach differs from the RouterFarm work in two regards. First, we propose to have the required network changes be triggered by functionality outside of the network (as opposed to network management functions inside the network). Second, due to the stringent timing requirements of live migration, we expect that our approach would require new router functionality (as opposed to being realizable via the existing configuration interfaces).

Finally, the recovery oriented computing (ROC) work emphasizes recovery from failures rather than failure avoidance [6]. In a similar spirit to ROC, we advocate using mechanisms from live VM migration to storage replication to support planned and unplanned outages in data centers (rather than full replication to mask such failures).

5. CONCLUSION

A significant concern for Internet-based service providers is the continued operation and availability of services in the face of outages, whether planned or unplanned. In this paper we advocated a cooperative, context-aware approach to data center migration across WANs to deal with outages in a non-disruptive manner. We sought to achieve high availability of data center services in the face of both planned and incidental outages of data center facilities. We advocated using *server* virtualization technologies to enable the replication and migration of server functions. We proposed new *network* functions to enable server migration and replication across wide area networks (such as the Internet or a geographically distributed virtual private network), and finally showed the utility of intelligent and dynamic *storage* replication technology to ensure applications have access to data in the face of outages with very tight recovery point objectives.

6. REFERENCES

- [1] M. Abd-El-Malek, W. V. Courtright II, C. Cranor, G. R. Ganger, J. Hendricks, A. J. Klosterman, M. Mesnier, M. Prasad, B. Salmon, R. R. Sambasivan, S. Sinnamohideen, J. D. Strunk, E. Thereska, M. Wachs, and J. J. Wylie. Ursa minor: versatile cluster-based storage. USENIX Conference on File and Storage Technologies, December 2005.
- [2] Mukesh Agrawal, Susan Bailey, Albert Greenberg, Jorge Pastor, Panagiotis Sebos, Srinivasan Seshan, Kobus van der Merwe, and Jennifer Yates. Routerfarm: Towards a dynamic, manageable network edge. SIGCOMM Workshop on Internet Network Management (INM), September 2006.
- [3] L. Alvisi. *Understanding the Message Logging Paradigm for Masking Process Crashes*. PhD thesis, Cornell, January 1996.
- [4] L. Alvisi and K. Marzullo. Message logging: Pessimistic, optimistic, and causal. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, pages 229–236. IEEE Computer Society, June 1995.

- [5] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebar, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In the Proceedings of the ACM Symposium on Operating Systems Principles (SOSP), October 2003.
- [6] A. Brown and D. A. Patterson. Embracing failure: A case for recovery-oriented computing (roc). 2001 High Performance Transaction Processing Symposium, October 2001.
- [7] K. Brown, J. Katcher, R. Walters, and A. Watson. Snapmirror and snapstore: Advances in snapshot technology. Network Appliance Technical Report TR3043. www.netapp.com/tech_library/3043.html.
- [8] C. Clark, K. Fraser, S. Hand, J. Hanse, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proceedings of NSDI*, May 2005.
- [9] Disaster Recovery Journal. Business continuity glossary. <http://www.drj.com/glossary/drjglossary.html>.
- [10] Laura Grit, David Irwin, Aydan Yumerefendi, and Jeff Chase. Virtual machine hosting for networked clusters: Building the foundations for autonomic orchestration. In *In the First International Workshop on Virtualization Technology in Distributed Computing (VTDC)*, November 2006.
- [11] M. Ji, A. Veitch, and J. Wilkes. Seneca: Remote mirroring done write. USENIX 2003 Annual Technical Conference, June 2003.
- [12] M. Kozuch and M. Satyanarayanan. Internet suspend and resume. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, Calicoon, NY*, June 2002.
- [13] Xiaotao Liu, Gal Niv, K. K. Ramakrishnan, Prashant Shenoy, and Jacobus Van der Merwe. The case for semantic aware remote replication. In *Proc. 2nd International Workshop on Storage Security and Survivability (StorageSS 2006)*, Alexandria, VA, October 2006.
- [14] Michael Nelson, Beng-Hong Lim, and Greg Hutchins. Fast Transparent Migration for Virtual Machines. In *USENIX Annual Technical Conference*, 2005.
- [15] Mendel Rosenblum and Tal Garfinkel. Virtual machine monitors: Current technology and future trends. *Computer*, 38(5):39–47, 2005.
- [16] C. Ruemmler and J. Wilkes. Unix disk access patterns. Proceedings of Winter 1993 USENIX, Jan 1993.
- [17] Paul Ruth, Junghwan Rhee, Dongyan Xu, Rick Kennell, and Sebastien Goasguen. Autonomic Live Adaptation of Virtual Computational Environments in a Multi-Domain Infrastructure. In *IEEE International Conference on Autonomic Computing (ICAC)*, June 2006.
- [18] Constantine P. Sapuntzakis, Ramesh Chandra, Ben Pfaff, Jim Chow, Monica S. Lam, and Mendel Rosenblum. Optimizing the migration of virtual computers. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, December 2002.
- [19] A. Sundararaj, A. Gupta, and P. Dinda. Increasing Application Performance in Virtual Environments through Run-time Inference and Adaptation. In *Fourteenth International Symposium on High Performance Distributed Computing (HPDC)*, July 2005.
- [20] Symantec Corporation. *Veritas Volume Replicator Administrator's Guide*. http://ftp.support.veritas.com/pub/support/products/Volume_Replicator/2%83842.pdf, 5.0 edition, 2006.
- [21] F. Travostino, P. Daspit, L. Gommans, C. Jog, C. de Laat, J. Mambretti, I. Monga, B. van Oudenaarde, S. Raghunath, and P. Wang. Seamless live migration of virtual machines over the man/wan. Elsevier Future Generations Computer Systems, 2006.
- [22] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and gray-box strategies for virtual machine migration. In *Proceedings of the Usenix Symposium on Networked System Design and Implementation (NSDI)*, Cambridge, MA, April 2007.
- [23] A xen way to iscsi virtualization? <http://www.internetnews.com/dev-news/article.php/3669246>, April 2007.