An Approach to High Availability for Cloud Servers with Snapshot Mechanism

Hoi Chan, Trieu Chieu
IBM T. J. Watson Research Center
1101 Kitchawan Road
Route 134
Yorktown Heights, N.Y, 10598

E-mail: {hchan,tchieu}@us.ibm.com

Abstract— *Virtualization technologies enable the execution of multiple virtual machine instances (VMs) with different operating systems (OSs) on the same physical host. Each VM instance functions independently as an isolated system with its own physical resources, OS and applications. Due to significant cost saving and efficiency, the virtualization model has been increasingly adapted by enterprises and service providers as their main computing and service delivery infrastructure, running critical internal business and external customer facing applications. To minimize down time due to unexpected VM crashes, a high availability or backup system is usually built into the infrastructure. There are many high availability technology options available such as replication, mirroring and fail over clustering. Most of these solutions are usually designed based on the traditional computing model, they are costly to implement, complicated and tedious to maintain, especially in a virtualized environment, and they often require additional expensive hardware and software components. In this paper, we introduce a simple, flexible, scalable, extensible, efficient and cost effective system which utilizes the current VM infrastructure and common utilities to provide a high availability solution in the virtualization environment. Our smart adaptive snapshot replication technique provides a smooth and reliable mechanism for cost-performance, wherein the amount of resources allocated for high availability solution can be adjusted based on available resources, utilization and customer requirements.*

Categoris and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]:Reliability, availability and serviceability

General Terms

Performance, reliability, management

Keywords

Cloud, VM, cloud, high availability, virtualization

I. INTRODUCTION

With the widespread utilization of virtualization technologies and Cloud Computing [1,2,3] due to its cost benefit and efficiency, the virtualization model has been increasingly adapted by enterprises and service providers as their main computing and service delivery infrastructure, running critical internal business and external customer facing applications. To minimize down time due to unexpected VM crashes, a high availability system is usually built into the infrastructure. In a virtualized computing environment, unlike the traditional computing model in which servers and hardware components are relatively static, the number of deployed VM instances can increase and decrease in a relatively short period of time, the rapid elastic nature of the Cloud computing model [4, 5, 6] presents a number of challenges to the high availability solution and infrastructure designers such as scalability, reliability and increased complexity in maintenance. Thus, it is obvious that the ability of the Cloud infrastructure to provide a simple, flexible, scalable and yet cost effective high availability solution will certainly add to the benefit of providing services in a virtualized environment.

High availability solutions are critical to business applications such as stock trading and banking transaction, and they have been studied extensively. Numerous solutions [7, 8, 9], such as replication, fail over cluster and mirroring are available and relatively matured, and many vendors provide the high availability solutions as complete commercial packages. Many researches focus on efficient recovery mechanisms such as removing I/O bottlenecks, fast image retrieval structure and efficient and optimized replication and mirroring [10, 11], as well as redundant hardware solutions [12, 13]. Most Cloud technologies providers include high availability solutions [15, 16, 17, 18] in their infrastructure offerings, but they tend to be expensive and require additional resources to manage.

There is little emphasis on utilizing the unique characteristics of Cloud computing and the utilities provided by the Cloud infrastructure to provide a simple, cost effective and efficient solution. A question that almost every CIO and software architect would ask: "How can innovative software techniques be used to enable better user experiences and increased reliability in the Cloud setting in a cost effective way?" These problems spur innovation in software design for Cloud infrastructure. In this paper, we introduce a simple, efficient and yet cost effective solution which utilizes the current VM infrastructure's snapshot mechanism and available API [22], coupled with a smart on-demand snapshot collection mechanism to provide a high availability solution in the virtualization environment.

The rest of this paper is organized as follows: Section 2 gives a brief summary of the current high availability solutions. Section 3 gives an overview of the snapshot mechanism in VM, and introduces the architecture of a high availability solution for VMs with adaptive snapshot merging. Section 4 shows the details and algorithm for the design and implementation using VMware's ESX snapshot mechanism and command line interface. Section 5 discusses failure detection and the tradeoff between high availability and performance. Section 6 describes the experiments running on ESX server and shows some initial performance result. Section 7 discusses issues related to the adaptive snapshot merging approach and possible future works while Section 8 concludes.

II. HIGH AVAILABILITY SOLUTIONS

High availability is an important requirement of mission critical applications such as stock trading, banking and on line business transactions. It maintains virtually seamless applications and systems availability in the events of hardware or software failure which results in the perceived downtime by the users to be negligible or minimal. Some of the common high availability solutions include 1) Mirroring – a typical mirroring system configuration involves a principal server and a mirror server that continuously brings itself current with the principal server. 2) Replication – based on the publish-subscribe model, the primary server publishes data and transaction details and distribute them to one or more subscriber servers. 3) Failover clustering – a group of independent servers collaborate to maximize the availability of applications and services. Each of the servers in the cluster is a node which is connected to the others by physical wires and the cluster as a whole is managed by specific software. If any of the nodes in the cluster fails, another node takes over and continues to provide services; disruptions of services to users are minimal. 4) Snapshot – taking snapshots of the states of the servers at regular interval such as Windows 7's backup system. Most of these high availability solutions require significant additional resources such as hardware and software to operate and manpower to maintain, which add to the operation cost. More importantly, they themselves often suffer availability issues. In addition, these solutions may not work well in a virtualized environment or be responsive or scalable enough to handle the sudden increase and decrease of deployed VM instances due to their fixed design and inflexible resource allocation in response to changes in VM deployment. Our proposed high availability solution focuses on the Cloud environment and takes advantage of the elastic nature of virtual computing and the underlying utilities and tools provided by the infrastructure.

III. OVERVIEW OF HIGH AVAILABILITY SOLUTION FOR VMS WITH SMART ON-DEMAND SNAPSHOT MERGING

Snapshot utilities are commonly included as part of the VM infrastructure, and they are virtually standard tools which help to better manage a Cloud infrastructure. In our first design and experimentation, we utilized a virtualization system which includes VMware's ESX 4.0 [19, 20] server with native Linux OS platform; we used VMware's ESX snapshot management system [21] because it provides readily available tools and API [22] for snapshot operations for our high availability solution. A brief introduction of VMware's snapshot operation is given here and more details are available from VMware documentation [19, 20, 21, 22].

A snapshot freezes and preserves the state and data of a virtual machine at a specific point in time. The data includes all of the files that constitute the virtual machine which includes disks, memory, and other devices, such as virtual network interface cards. When a snapshot is created, a child disk is initiated with the base disk as its parent. The child disk is a sparse disk which uses the copy-on-write (COW) mechanism. It contains no data, until a write operation copies data and stores in the disk resulting storage space optimization. This snapshot will become the parent of the next created snapshot and so on. In VMware's snapshot management system, when a specific snapshot is removed, it commits (merges) the snapshot content to its parent snapshot. When all the snapshots are deleted, the snapshot furthest from the base disk is merged (committed) to its parent first. When the first commit operation is complete, the snapshot is removed and the process starts over on the newly merged snapshot to its parent. This continues until all the snapshots in the snapshot ladder have been committed.

Figure 1 shows the architectural overview of our high availability solution with the following basic components:

Snapshot Agent (Figure 2): it resides and runs on the OS of the physical host. Each physical host will have one instance of the snapshot agent. It registers itself to the snapshot processor when activated. It gets VM operating data from the hypervisor and collects snapshots for each of the VMs running on the host at a configurable interval and sends the snapshot to a remote host for processing (snapshot processor). In addition, as a secondary function, it receives image to restart a specific VM if a VM recovery is needed. The snapshot agent includes a pluggable policy engine to dynamically control the snapshot collection and merging operations. It collects system and application performance data and decides when a snapshot should be taken; it also determines the merging frequency so as to optimize the performance of the entire system as a whole. The operation details and specifics of the policy engine for snapshot management is not the focus of this paper and will be discussed in a separate paper. In addition to an embedded decision making engine, the snapshot manager continuously monitors the health of its managed VM instances running on the physical host and is responsible to notify the snapshot manager if any of its monitored VM instances experience abnormal behavior or crash to enable the snapshot manager to take corrective actions.

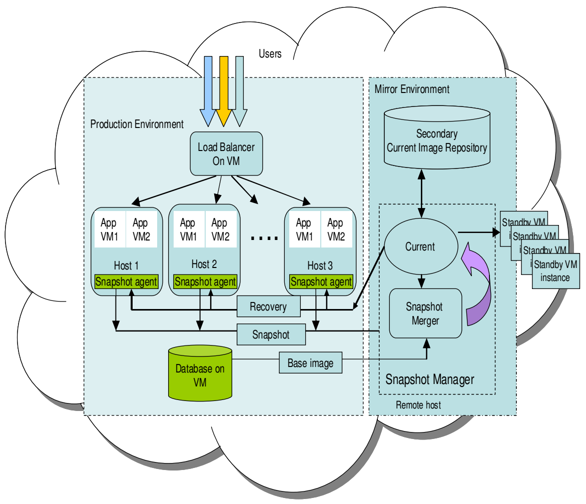

Figure 1 Overview of high availability system in Cloud using VMware's snapshot mechanism.

Snapshot Manager (Figure 3): it resides and runs on a separate and independent physical host. It maintains a registry for client snapshot agents and is responsible for updating the snapshot collection policy and configuration of its client snapshot agents. It connects and maintains communication with each of its registered snapshot agents and retrieves the most recent fully committed copy of any the VM running on its managed hosts once a VM instance is created. It receives periodical snapshots collected by the snapshot agent and performs merging with their respective parents as soon as the snapshots are received; the resulting committed image is then made available in a repository for recovery purpose. For critical deployed VMs, the snapshot manager also, according to SLAs or user settings, maintains an array of standby VM instances in separate hosts with its most recent committed image as a direct replacement (after configuration such as IP settings) if their associated VM image experiences abnormal behavior or crashes, as observed by the snapshot agent.

Secondary Image Repository (Figure 3): an optional secondary storage component which will serve as a further backup storage for the most updated image for each of the deployed VMs.

Standby instance option (Figure 3): to provide near-zero loss of work or data for high value VM instances in case of unexpected failure, standby VM instances are created (optionally) to enable seamless switch over. The collection of standby VM instances mirrors exactly their associated VM instances except the current delta. These standby instances are inactive (no IP configured) until they are called upon to replace their associated instances.

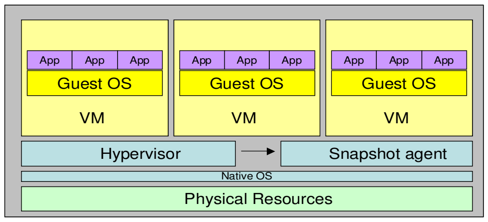

Figure 2 Snapshot agents controls snapshot collection and recovery.

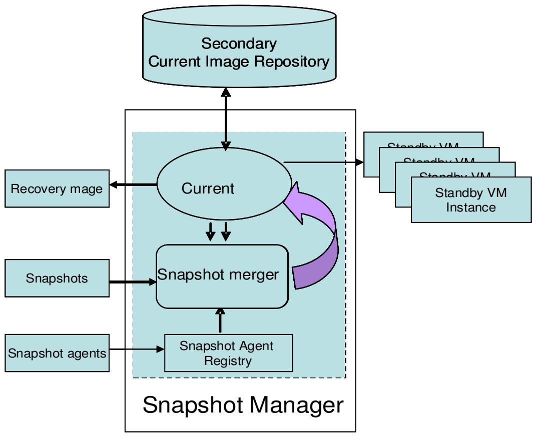

Figure 3 Snapshot processor and secondary image storage

IV. ON DEMAND SNAPSHOT MERGING ALGORITHM

For simplicity and illustration purpose, our first set up and experiments are based on the popular VMware's ESX 4.0 snapshot management systems and its utilities. The steps listed below highlights the basic algorithm used in our snapshot mirroring system.

1. When a new VM instance is created, a snapshot agent automatically starts and runs in the background and registers itself with the remote Snapshot Processor.
2. Upon registration of the new snapshot agent, the snapshot manager will instantiate the snapshot agents with a specified policy engine together with an operation policy.

3. After registration and the successful start of the snapshot manager, the snapshot processor acquires a copy of the image used by the instance including the state of its configuration (vmx and vmxf files), memory contents (vmsn files), and disk state (vmdk files).
4. Snapshot manager will create standby images according to SLAs (not all VMs need standby instances, some VMs will tolerate s specified delay in powering on the VMs from the mirror images).
5. Snapshot agent takes snapshots and sends snapshots to the snapshot manager according to its current policy and register the child to the ESX Server. (Snapshots can be sent to the process individually or as a batch)
6. For the snapshots (n) of each of the VM instances, delete n-1 snapshots. This will merge all the n-1 snapshots with their respective parents.
7. The newly created merged image is put in the recovery image directory as the most updated image needed for recovery.
8. The newly created merged image becomes the parent of any new incoming snapshot.
9. If VM instance replacement (recovery) is needed, start the VM using the most updated merged image in standard mode or configure and activate the standby instance. IP remains the same for the new instance as the recovery image is an exact duplicate of the original instance up to last snapshot.
10. For each registered snapshot agent, repeat 3 to 8.

Figure 4 shows the evolution of the active and mirror VM image in a timeline. The sizes of both active and mirror VM images change with time with the mirror image always lags behind the active image with the running delta. Since the backup snapshots are copied to a remote node, merging can be done immediately upon new snapshot arrival while the merging of snapshots on the active VM can be performed at longer interval.

As mentioned briefly in the previous paragraph, a pluggable policy engine can be embedded into the snapshot agent to dynamically control the snapshot collection and merging operations. Service providers can utilize an intelligent algorithm or policy to determine the most optimized time for snapshot collection and merging operation for different categories of Cloud service consumers with different requirements for high availability solutions.

V. Failure Detection And Optimized HA

The snapshot manager exchanges heartbeats with the snapshot agents in each of the hosts, the absence of heartbeats for a certain amount of time indicates host failure and a complete recovery using the merged images for each of its VMs will be initiated. The snapshot agent in each of the hosts notifies the snapshot manager if any of its monitored VM instances crashes which in turn triggers recovery for the crashed VM using the latest merged image.

Availability is usually expressed as a percentage of system uptime in a given year [23]. In general, the higher the availability, the more it will cost to implement and maintain the infrastructure. The goal of our continuous snapshot collection and merging algorithm is to provide a cost effective high availability solution without significant performance impact and minimal data loss. Assuming we are operating in a Cloud environment in which we can allocate spare resources without delay, there are a few major factors which will impact system and availability performance: snapshot frequency (f), snapshot creation time (ct), snapshot merging time (mt), switching secondary image to primary image (st) and recovery time (rt).

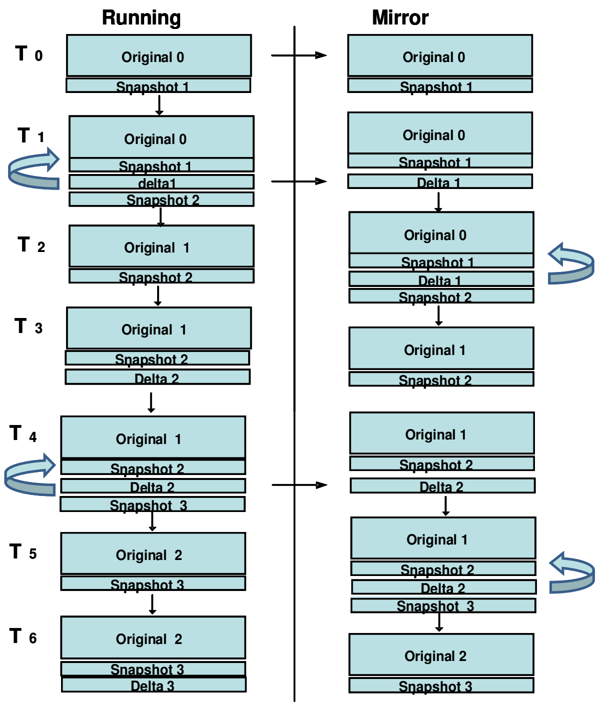

Figure 4 evolution of the active and mirror VM image in a timeline

are operating in a Cloud environment in which we can allocate spare resources without delay, there are a few major factors which will impact system and availability performance: snapshot frequency (f), snapshot creation time (ct), snapshot merging time (mt), switching secondary image to primary image (st) and recovery time (rt).

$$HA = Fn\ (f, ct, st, mt, rt)$$

Snapshot frequency and snapshot creation time are the two important factors determining the effectiveness of the snapshot merging algorithm. Creating a snapshot involves copying the memory content into delta files and typically for a several GB data block; it takes a few seconds using high performance network hardware. If we limit performance impact to 1% of system time and assume that it takes about 5 seconds to create a snapshot, then the snapshot collection frequency should not be more than 8 minutes/snapshot. Higher snapshot collection frequency will result in less data loss during recovery but incurs a significant performance impact. In addition, it will take longer to merge the snapshots (figure 5) if the delta size becomes large due to delayed snapshot taking. Switching time from primary to secondary image in the event of VM failure also affects the availability performance, and it is mostly the time needed to boot the secondary image into operation. Cloud Service providers will use these parameters with optimization algorithm to create high availability policy for each category of consumers to comply with their respective SLAs. We expect such policy to utilize the "fixed time interval" or/and "delta size threshold" snapshot collection approach with the proper optimization strategy to determine the optimum snapshot collection interval.

VI. EXPERIMENTS

We conducted a series of tests and experiments on a popular virtualization platform; a VMware ESX* 4 server [18] hosted on an IBM Blade server (IBM Blade Center** HS22 - 7870 - 4GB RAM - 2.53 GHz) and IBM System x3550 M2 8 CPUs x 2.66 GHz 64 GB RAM. VM instance was RHEL 5.4 (32-bit) with 1 CPU & 4 GB RAM.

1. **Snapshot collection**: We installed a DB2** server and used a read/write workload (load conditions: CPU 50-70%, VM network ~120 kBps, Disk read rate ~50 KBps) and measured delta and vmsn file sizes with different intervals. While vmsn file size matched the memory size, delta size varied roughly linearly with collection interval (figure 5). As expected, delta size depends strongly on network and workload characteristics.

2. **Snapshot completion**: We measured snapshot completion time using ESX Snapshot Manager with 4GB of VM memory and without memory. Snapshot completion time was roughly constant at ~350-400 secs with 4GB of VM memory and 6-8 secs without memory. While it took roughly ~350-400 secs for the snapshot to be completed, the primary VM was on hold and became available for transactions in roughly 90 secs. We further measured the snapshot completion time for VM with 1 and 2 GB of memory (figure 6) and observed that the primary VM was on hold and became available after approximately 60 secs. Memory copy plays an important role in snapshot processing and roughly equals to the time the primary VM is unavailable. Delta files could grow fast for multiple snapshots, it is very important that real time merging of snapshots is performed to avoid impact to storage.

3. **Snapshot merging**: We measured snapshot merging time with snapshots of different delta sizes using ESX Snapshot Manager. For a delta image with no new data, it took about 5-7 seconds to merge; we consider the 5-7 seconds elapsed time to be the overhead for the merging process. Figure 7 shows delta size varies linearly with merging time from the initial 5-7 overhead time.

From the data, with a typical VM with 2GB of memory, creating a snapshot takes roughly 160 secs while the VM is in hold state for about 60 secs. If we limit host performance impact to less than 8% to 10%, then snapshot collection frequency should not exceed 10-12 minutes per snapshot. Snapshot merging can be done remotely on separate hosts or other under-utilized resources; we do not consider it to be a major factor in determining the snapshot collection frequency. On the other hand, snapshot merging time is important in determining recovery time as the recovery process must wait for the completion of latest merged after a VM crash.

Frequent snapshot collection decreases data loss in the event of VM failure but will negatively impact overall system performance. Infrequent snapshot collection results in potentially large delta image and causes intolerable data loss in the event of VM failure in addition to the negative impact on recovery time due long snapshot merging time. The idea is to balance the snapshot collection interval and size as well as the snapshot merging process with the SLA objective.

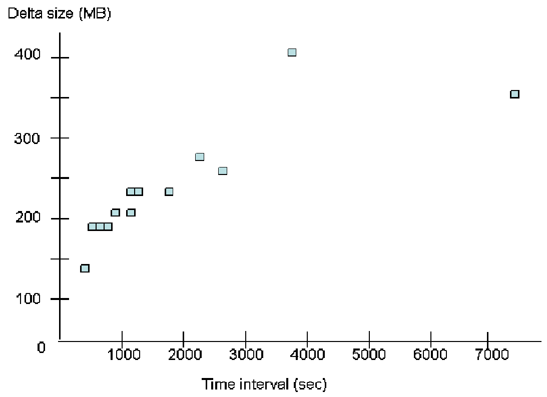

Figure 5 shows delta file size vs snapshot interval

VM Memory	Snapshot completion time
None	6-8 secs
1GB	~70 secs
2GB	~160 secs
4GB	350-400 secs

Figure 6 shows VM memory and snapshot completion time

Our system includes a pluggable policy engine to enable service providers to control the frequency, time of collecting and merging the snapshots with respect to SLA objective. Likewise, our system also support using an autonomic, self optimizing and learning engine to automatically adjust the frequency and time of snapshot collection and merging based on current and projected workloads, environment conditions and SLA objectives.

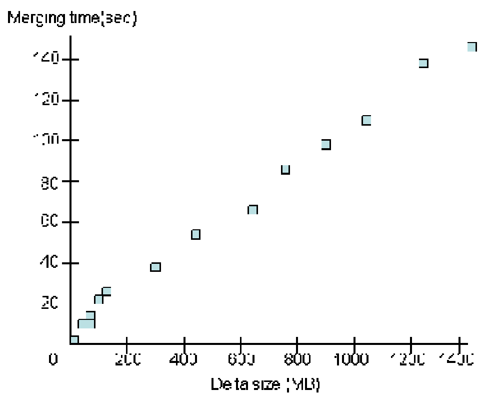

Figure 7 shows delta file size vs merging time

VII. Issues And Future works

There are issues to be explored before its full benefit and potential can be realized: (1) Not all current virtualization platforms and infrastructures can be adapted for this approach due to availability of tools. (2) We rely on tools such as snapshot management tools provided by the platform, which are not likely to be standard and platform independent. (3) More experiments needed to be conducted to understand better the effect of snapshot collection under different VM operating conditions such as disk access and such effect on the primary system. These are interesting issues, especially in a virtualized environment that may stimulate further research interests. For next steps, we will focus on addition performance measurements such as building a complete system with possible on-demand mirroring to further enhance the backup and recovery process in the Cloud environment.

VIII. Conclusion

The primary objective of a high availability solution is to enable seamless and continuous VM operation (or its applications) with no or negligible interruption in a simple and yet efficient way. VM technologies have evolved over the years that there are useful and powerful tools available even in the basic VM platform. To optimize cost and performance, we take advantage of these built-in tools and the special elastic nature of the virtualized environment to build a cost-effect solution for high availability.

In summary, our initial works have showed the feasibility of using the continuously snapshot collection and merging to provide a cost effective high availability solution in Cloud environment. We have demonstrated indirectly the key concept and steps in a popular commercial virtualization platform and shows performance data. We also explained that this approach utilizes the current available tools provided by platform without purchasing application software licenses. The simplest form of this high availability solution can be implemented using ordinary OS scripts such as PERL. Its simplicity, low cost and highly adaptable and scalability provides an alternative to expensive and dedicated high availability solution in certain applications.

References

[1] G. Gruman, "What cloud computing really means", InfoWorld, Jan. 2009.

[2] R. Buyya, Y. S. Chee, and V. Srikumar, “Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities”, Dept of CS & SE. University of Melbourne, Australia, July 2008, pp. 9.

[3] D. Chappell, “A Short Introduction to Cloud Platforms”, David Chappell & Associates, August 2008.

[4] Amazon (EC2). http://aws.amazon.com/ec2/.

[5] http://websphere.sys-con.com/node/1017378

[6] Y.Wei, M.Blake. Service-Oriented Computing and Cloud Computing: Challenges and Opportunities: Internet Computing, IEE Issue: Nov.-Dec. 2010 Volume: 14 Issue:6

[7] R. Tewari, D. Dias, R. Mukherjee, H. Vin, High availability in clustered multimedia servers, Data Engineering, 1996. Proceedings of the Twelfth International Conference on Issue Date: 26 Feb-1 Mar 1996

[8] R. Moreno-Vozmediano, R. Montero, I Llorente.Elastic management of cluster-based services in the cloud :· ACDC '09 Proceedings of the 1st workshop on Automated control for datacenters and clouds ACM New York, NY, USA ©2009

[9] L. Valcarenghi, M. Kantor, P. Cholda, K. Wajda, HA to client-server communications: Transparent Optical Networks, 2008. ICTON 2008. 10th Anniversary Issue Date: 22-26 June 2008 On page(s): 34 – 37, 645 – 654

[10] H. Sun; J. Han, H. Levendel, Availability requirement for a fault-management server in high-availability communication systems, High Reliability & Availability Technol. Center, Motorola, Deer Park, IL, USA: Reliability, IEEE Transactions on Issue Date: June 2003 Volume: 52 Issue:2 On page(s): 238

[11] D. Dias, W Kish, R. Mukherjee, R. Tewari.A scalable and highly available web server, Compcon '96. 'Technologies for the Information Superhighway' Digest of Papers Issue Date: 25-28 Feb 1996 On page(s): 85 – 92

[12] R. Katz, W. Hong, The performance of disk arrays in shared-memory database machines Distributed and Parallel Databases Volume 1, Number 2, 167-198

[13] Golubchik, L.; Lui, J.C.S.; Muntz, R.R.; Chained declustering: load balancing and robustness to skew and failures: Research Issues on Data Engineering, 1992: Transaction & Query Processing, 2nd International Workshop

[14] K Bowers, A Juels, A. Oprea.HAIL: a high-availability and integrity layer for cloud storage. CCS '09 Proceedings of the 16th ACM conference on Computer and communications security ACM New York, NY, USA

[15] http://www.vm6software.com/microsoft-hyper-v-virtualization-solutions

[16] http://www.VMware.com/solutions/smb/

[17] http://www.clone-systems.com/private-cloud-high-availability.html

[18] http://www.stratus.com/Solutions/ByITNeed/Cloud.aspx

[19] http://www.VMware.com/products/vsphere/esxi-and-esx/index.html

[20] http://www.VMware.com/support/ws55/doc/ws_preserve_snapshot_manager.html

[21] http://kb.VMware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1009402

[22] http://www.VMware.com/support/developer

[23] http://en.wikipedia.org/wiki/High_availability

**IBM, BladeCenter are registered trade marks of IBM Corp.

*VMware, ESX are registered trade marks of VMware Corp.