Pp. 47-64 of the *Proceedings of LISA '02: Sixteenth Systems Administration Conference* (Berkeley, CA: USENIX Association, 2002).

Geographically Distributed System for Catastrophic Recovery

Kevin Adams – NSWCDD

ABSTRACT

This paper presents the results of a proof-of-concept implementation of an on-going project to create a cost effective method to provide geographic distribution of critical portions of a data center along with methods to make the transition to these backup services quick and accurate. The project emphasizes data integrity over timeliness and prioritizes services to be offered at the remote site. The paper explores the tradeoff of using some common clustering techniques to distribute a backup system over a significant geographical area by relaxing the timing requirements of the cluster technologies at a cost of fidelity.

The trade-off is that the fail-over node is not suitable for high availability use as some loss of data is expected and fail-over time is measured in minutes not in seconds. Asynchronous mirroring, exploitation of file commonality in file updates, IP Quality of Service and network efficiency mechanisms are enabling technologies used to provide a low bandwidth solution for the communications requirements. Exploitation of file commonality in file updates decreases the overall communications requirement. IP Quality of Service mechanisms are used to guarantee a minimum available bandwidth to ensure successful data updates. Traffic shaping in conjunction with asynchronous mirroring is used to provide an efficient use of network bandwidth.

Traffic shaping allows a maximum bandwidth to be set minimizing the impact on the existing infrastructure and provides a lower requirement for a service level agreement if shared media is used. The resulting disaster recovery site, allows off-line verification of disaster recovery procedures and quick recovery times of critical data center services that is more cost effective than a transactionally aware replication of the data center and more comprehensive than a commercial data replication solution used exclusively for data vaulting. The paper concludes with a discussion of the empirical results of a proof-of-concept implementation.

Introduction

Often data centers are built as a distributed system with a main computing core consisting of multiple enterprise class servers and some form of high performance storage subsystems all connected by a high speed interconnect such as Gigabit Ethernet [1]. The storage subsystems are generally combinations of Network Attached Storage (NAS), direct connected storage or a Storage Area Network (SAN). Alvarado and Pandit [2] provide a high-level overview of NAS and SAN technologies and argue that these technologies are complimentary and converging.

In order to increase the availability of such a system, the aspects of the systems' reliability, availability, and serviceability (RAS) must be addressed. Reliability, availability and serviceability are system level characteristics, which are invariably interdependent. Redundancy is the way resources are made more available. This redundancy permits work to continue whenever one, and in some cases, more components fail. Hardware fail-over and migration of software services are means of making the transition between redundant components more transparent. Computer system vendors generally address hardware and operating system software reliability. For example, Sun Microsystems has advertised a guaranteed 99.95% availability for its standalone Enterprise 10000 Servers [3]. Serviceability implies that a failure can be identified so that a service action can be taken. The serviceability of a system obviously directly affects that system's availability. A more subtle concern is the impact of increasing system reliability and redundancy through additional components. Each additional software or hardware component adds failure probabilities and thus any project to increase the availability of a system will involve a balance of reliability and serviceability as well. Network Appliance provides a good example of this tradeoff in the design of their highly available (HA) file servers [4].

Disaster protection and catastrophic recovery techniques are not generally considered as part of a vendor HA solution, but the economic reasons which drive HA solutions [5] demand contingency planning in case of a catastrophic event. In short, HA protects the system; disaster recovery (DR) protects the organization.

Recent technical capabilities, particularly in the area of networking have enabled common-off-the-shelf (COTS) clusters to emerge. This paper begins to examine the same technologies and general techniques used in COTS clusters for their feasibility as techniques to provide geographically distributed systems appropriate for use as remote disaster protection facilities at reasonable cost. In the paper we define geographically distributed in terms of limited communications bandwidth not a distance measurement.

The goal of this work is to outline a design which provides a DR facility that can be made operational quickly for critical functions, provide a means of verifying DR plans and procedures, minimize data loss during the disaster and provide the basis for the reconstruction of the company's computing base. The premise of the paper is to explore the use of HA cluster technologies to distribute a backup system over a significant geographical area by relaxing the timing requirements of the cluster technologies at a cost of fidelity.

The trade-off is that the fail-over node is not suitable for HA usage as some loss of data is expected and fail-over time is measured in minutes not in seconds. Asynchronous mirroring, exploitation of file commonality in file updates, IP Quality of Service (QoS) and network efficiency mechanisms are enabling technologies used to provide a low bandwidth solution for the communications requirements. Exploitation of file commonality in file updates decreases the overall communications requirement. IP QoS mechanisms have been designed to add support for real-time traffic.

The design presented takes the real-time requirements out of the HA cluster but uses the QoS mechanisms to provide a minimum bandwidth to ensure successful updates. Traffic shaping in conjunction with asynchronous mirroring is used to provide an efficient use of network bandwidth. Traffic shaping allows a maximum bandwidth to be set, minimizing the impact on the existing infrastructure and provides a lower requirement for the service level agreement (SLA).

The next section outlines the approach used for providing a geographically distributed system to support continued operation when a disaster has occurred. Then, DR, HA and IP QoS background is provided. The subsequent section provides details and results of a specific proof-of-concept study. Impacts of the DR elements on the production system are examined along with the issues found during the case study. The paper concludes with a summary and discussion of future work on this project.

The Approach

When developing a DR contingency plan, the restoration of *critical* operations of a data center takes priority. This section proposes a design for a "warm backup" site for such operations using existing communication channels or lower bandwidth commercial communication channels. This design is a compromise between the restoration from backup tapes and a fully synchronous DR facility. The third section offers further discussion of DR options. Tape restoration will be required for non-critical services. The backup site is also not kept synchronous with the primary but is synchronized at a point in the past through asynchronous mirroring. This solution is appropriate for organizations that can survive some loss of availability along with potentially the loss of some data updates. The case study was intended to evaluate the feasibility and gain of the proposed DR solution. Specifically, the case study is intended to evaluate the feasibility of deploying the DR system supporting one terabyte of

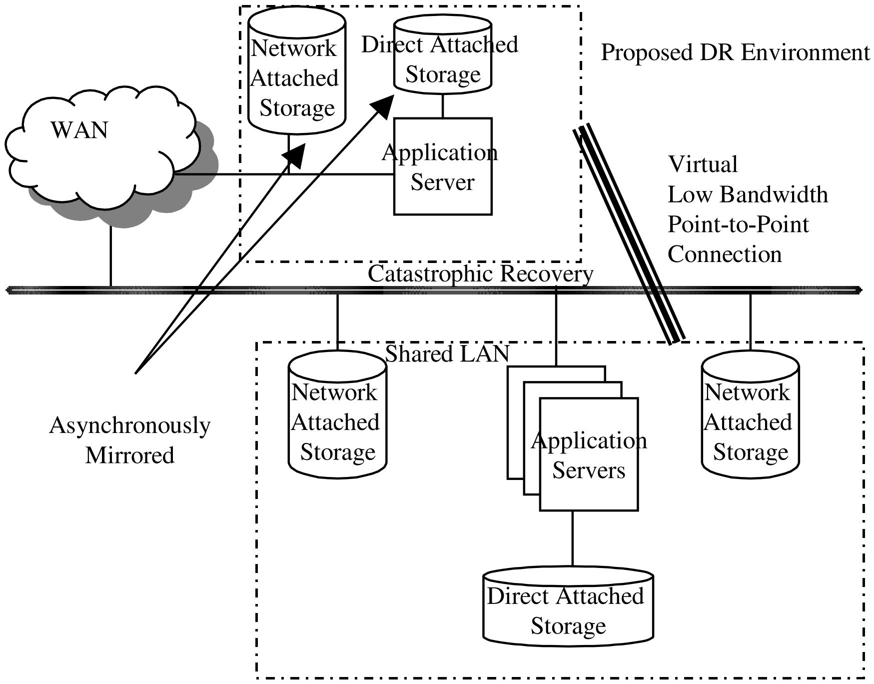

Diagram 1: Proposed DR architecture.

data with no more than 24 hours of lost data updates over an existing T1 (1.544 Mbps) line or another alternative is to use a small portion of a shared WAN.

Topology

The topology of the data center is assumed a centrally located facility of a few to tens of application, file and database servers. Users interface to these services through desktop clients.

The topology of the DR site is assumed to be one of each compatible server interconnected with its primary. Thus, there is a consolidation at the DR site of critical services on many production servers to a single DR server. The data associated with each service must be available to the service's server. All DR servers are interconnected over a LAN. The application consolidation greatly reduces the cost of the DR site. However, the application consolidation complicates the network identity fail-over and is discussed in the Network Identity Fail-over section.

The production environment for the case study is comprised of data service provided via multiple Network Appliance file servers (known as filers) and application services provided by multiple Sun Microsystems Enterprise 5x00 servers and an Enterprise 10000 server. At the DR site, the application services are being replicated on an Enterprise 5000 server and data services are being consolidated to a single Network Appliance F740 filer. Diagram 1 provides an overview of the target architecture. In the proof-of-concept a subset of applications and data are being tested on the local LAN to determine feasibility and SLA requirements for DR deployment. The proof-of-concept test environment is discussed further in the study results section.

Fail-over

A survey of commercial HA solutions (see the Background section) can be generalized as providing for the movement of three critical elements from the failed server(s) to the backup: the network identity, the data, and the set of processes associated with that data. Additionally, a service to monitor the health of each primary service, each backup server, and communications between primary and backup servers is required. This service is known as a heartbeat. In HA solutions this fail-over is normally automated. Since the problem at hand is providing availability in the face of a disaster, which may be predicted and a preventative fail-over initiated or it may be prudent to delay initiation of a fail-over until the culmination of a disaster has occurred, a manually initiated automated fail-over process is used.

Heartbeat

A mechanism to determine system and communications liveliness is required, but the determination is not required continuously as it is for HA. The main issue for this DR site is to keep the data and processes in synchronization to the fidelity of the DR requirements. Fail-over does not rely on a heartbeat for initiation and synchronization occurs through asynchronous mirroring or shadowing periodically not continuously. Therefore, the determination of system liveliness is required only before the initiation of a synchronization process. The heartbeat mechanism will need to be specific to the file service, mirroring software, and communication technology used. If any errors occur, operational personnel need to be alerted of the problem, but there should be no impact to the production data center.

In the case study, a Korn shell script was written to determine system liveliness. As described in the Data Migration subsection below, the remote mirroring occurs on a volume basis so to determine file system liveliness, prior to initiation of each volume synchronization, the primary and backup filer status is checked and logged via a series of remote status commands (e.g., from Solaris: rsh nacbac sysstat). The status of the primary and backup servers and communications network liveliness is verified and logged by checking the respective network interfaces using various operating system supplied status utilities (e.g., from Solaris: ping, netstat). In the prototype, if any errors occur, e-mail is sent to the personnel involved in the test. In the final production system, alerting operational personnel should be integrated into the system and network management platform.

Process Migration

If the DR site is a mirror of the production data center then commercial shadowing software can be used to synchronize data, applications and system configurations. Since it was assumed that the DR site is not a mirror of the data center, services must be prioritized into separate categories. Each category should have an increasing tolerance for unavailability. These services must be installed, configured and updated along with the primary servers in the data center.

In the case study, only select services are installed on the DR servers and all services must be restarted at fail-over. This fail-over involves bringing the data online with read and write access and a reboot of the servers.

Data Migration

In a DR situation a copy of the data associated with the migrated services must be provided to the DR facility. The integrity of the data must be ensured along with the synchronization of the system as a whole. Commercially, data replication solutions provide a method of supplying and updating a copy of the data at an alternate site. Commercial Data Replication solutions are database, file system, OS or disk subsystem specific; thus, enterprises may be required to use multiple solutions to protect their critical data. The Gartner Research Note T-13-6012 [6] provides a table that differentiates 24 products by the options they support.

In the production environment for the case study, data replication solution must be provided for the network attached storage, direct attached storage controlled under the Solaris OS and Veritas volume management, along with special considerations for Sybase and Oracle data. The initial proof-of-concept is looking at a subset of the production data environment, specifically the network attached storage with Oracle database data.

Bandwidth Utilization

For this project geographic distribution has been defined in terms of limited communications bandwidth, thus our design seeks to minimize communication requirements. Three bandwidth limiting techniques and compromises are used.

Our first compromise is with the heartbeat. The heartbeat, as previously discussed, is relaxed from a real-time or near real-time monitor to one that only requires activation upon a synchronization event. In the case study, this was once daily and the heartbeat's impact is negligible.

Our second compromise is with data replication. The data is shadowed not synchronously mirrored. This allowed the use of a network efficiency mechanism known as traffic shaping. See Diagram 2.

The objective of traffic shaping is to create a packet flow that conforms to the specified traffic descriptors. Shaping introduces a queuing of network traffic, which is transmitted at a fixed rate resulting in high network utilization. Shaping may change the characteristics of the data flow, by intentionally delaying some packets, thus the need for asynchronous mirroring. Traffic shaped asynchronous mirroring enables data synchronization between the local and remote copies to occur over long periods of time with a constant network impact.

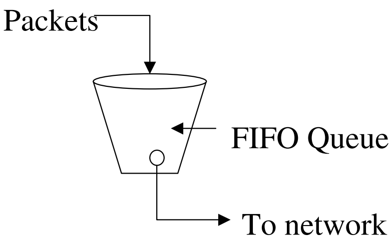

Diagram 2: Traffic shaping.

Even if traffic is shaped at the source, it may become jittered as traffic progresses through the network. To address this jitter, a point-to-point link is required or the traffic should be shaped just prior to entering the low bandwidth link.

Traffic shaping allows a maximum bandwidth to be set minimizing the impact on the existing infrastructure and provides a lower requirement for the service level agreement (SLA). In any communication system using traffic shaping, the finite queue must remain stable.

Queue stability relies on two parameters, the inter-arrival time and the service rate. The service rate of the queue must be greater than data inflow; in our case, this means setting the maximum data rate allowed on the network high enough. Secondly, the network must be able to successfully transmit the data when serviced out of the queue. IP QoS mechanisms are used to guarantee the necessary bandwidth availability.

Bandwidth availability is greater than required to perform traffic shaped data synchronization, but the high network utilization afforded from traffic shaping will prevent over design to accommodate peak loads over the low-bandwidth link. Traffic shaping and other IP QoS routing mechanisms specifically in a Cisco IOS environment are further discussed in the Background IP Quality of Service section

Our final effort to minimize communications between the local and remote site is an exploitation of file commonality in file updates. Data shadowing products were evaluated which allowed block level updates as opposed to file level updates. It is expected that block level updating will significantly reduce the required communications.

Network Identity Fail-over

The fail-over of the network identity is driven by client availability and for the purpose of DR is more properly stated restoration of client access. If the DR scenario allows for client survivability, the movement of network identity must be addressed. If the DR scenario requires clients to also be replaced, network identity becomes secondary to the client replacement process. An example of client replacement is provided in later in this section.

When a fail-over occurs, the IP address and logical host name used by the Data Center server need to migrate to the DR server. Normally, this is done by reconfiguring the public network interfaces on the takeover server to use the public IP address. This process is complicated by the mapping of the hardware MAC addresses to IP addresses.

The Address Resolution Protocol (ARP) is used to determine the mapping between IP addresses and MAC addresses. It is possible, and common on Sun platforms, to have all network interfaces on a host share the same MAC address. Many system administrators tune the ARP cache used by clients to store the IP-MAC addresses for anywhere from 30 seconds to several hours. When a fail-over occurs, and the IP address associated with the service is moved to a host with a different MAC address, the clients that have cached the IP-MAC address mapping have stale information. There are several ways to address the problem:

- Upon configuration of the DR server's network interfaces, a "gratuitous ARP" is sent out informing other listening network members that a new IP-MAC address mapping has been created. Not all machines or operating systems send gratuitous ARPs, nor do all clients handle them properly.

- The MAC address can be moved from the data center server to the DR server. The clients need to do nothing, as the IP-MAC address mapping is correct. Switches and hubs that track MAC addresses for selective forwarding need to handle the migration; not all equipment does this well. Binding an Ethernet address to an interface is shown using the Solaris naming and configuration syntax:

```
ifconfig qfe1 ether 8:0:20:1a:2b:33
```

- Wait for the clients ARP cache entries to expire, resulting in the clients realization that the host formerly listening on that MAC address is no longer available and send a new ARP requests for the public IP address.

Movement of the IP address and logical name from the data center server to the DR server is simpler. The use of a virtual hostname and IP address is common. Most network interface cards support multiple IP addresses on each physical network connection, handling IP packets sent to any configured address for the interface. The data center hostname and IP address are bound to virtual hostname and IP address by default. DR and data center server synchronization can occur using the "real" IP address/hostnames. At fail-over, the virtual hostname and IP address are migrated to the DR server. Clients continue to access data services through the virtual hostname or IP address.

Enabling a virtual IP address is as simple as configuring the appropriately named device with the logical IP address, here shown again using the Solaris naming and configuration syntax:

```
ifconfig hme0:1 jupiter up
```

and the "real" addresses are configured one of two ways: on a data center server named europa

```
# ifconfig hme0 plumb
# ifconfig hme0 europa up
```

or on a DR server named io

```
# ifconfig hme0 plumb
# ifconfig hme0 io up
```

The virtual IP is associated with the physical hme0 interface.

If a DR server is a consolidation of several data center servers, virtual IP addresses can be set up for each data center server on the DR server. MAC addresses are assigned per interface so installing an interface for each consolidated server allows movement of the Ethernet addresses. Otherwise, waiting for the ARP cache timeout or a gratuitous ARP can be used.

Client Service Migration

The final task is how to re-establish the clients. Assortments of clients are in use within the case study's production environment (PCs, UNIX Workstations and thin clients) but for DR, Sun Ray thin clients were chosen. The Sun Ray server software is installed on the DR server to drive the thin clients. A small number of thin clients are being set-up at the remote site to allow quick recovery with capabilities to add up to 50 units if needed. This is far below the production environment's normal user load (see Chart 1), but represents a first step towards a return to normalcy.

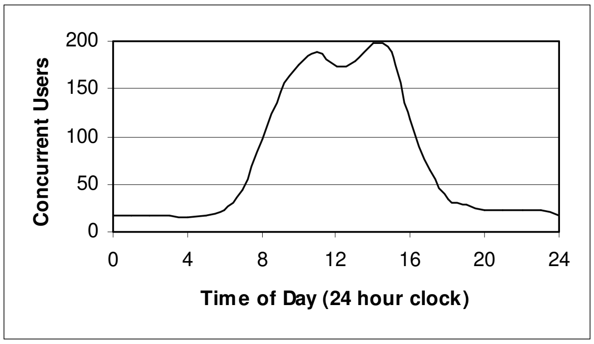

Chart 1: Usage load.

The Sun Ray enterprise system is a Sun Microsystems' solution based on an architecture that Sun calls the *Sun Ray Hot Desk Architecture* [7]. The Sun Ray enterprise system provides a low cost, desktop appliance that requires no desktop administration, is centrally managed, and provides a user experience equivalent to that of a Sun workstation if servers and networks are properly sized [8]. The Sun Ray appliance is stateless, with all data and processing located on the server. Access is provided to both Solaris and Microsoft Windows 2000 TSE through the Citrix ICA client from a single desktop [9]. The Windows Citrix Servers provide administrative services and are not part of the DR site design but will be required to be rebuilt from tape on new hardware in the event of a disaster.

Background

Failures caused by a catastrophic event are highly unlikely and difficult to quantify. As a result, catastrophic event failures are not normally accounted for in most HA calculations even though their rare occurrence obviously affects availability. The background section begins by defining availability and the levels of availability. DR, HA and the relationship between the two respectively is then introduced. The background section concludes with an introduction to IP QoS mechanisms provided by network routers, with a heavy bias toward QoS features supported in Cisco's IOS. Cisco routers and switches are used in the case study production environment.

Availability

Availability is the time that a system is capable of providing service to its users. Classically, availability is defined as uptime / (uptime + downtime) and provided as a percentage. High availability systems typically provide up to 99.999 percent availability, or about, five minutes of down time a year. The classic definition does not work well for distributed or client/server systems. Wood of Tandem Computers [10] presents an alternative definition where user downtime is used to make the availability calculation. Specifically,

$$\frac{\displaystyle\sum_{total\ users} \frac{user\ uptime}{user\ uptime + user\ downtime}}{total\ users}$$

expressed as a percentage.

Wood continues in his 1995 paper to predict client/server availability. The causes of failures used in Wood's predictions are summarized in Figure 1.

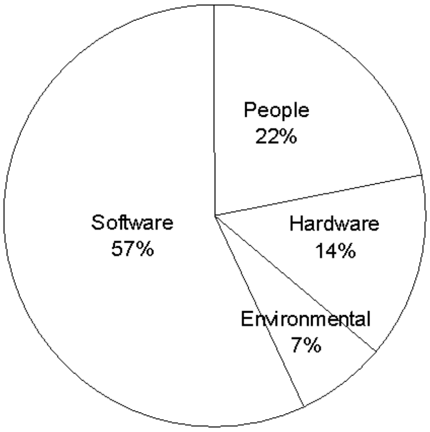

Figure 1: Causes of downtime.

Barr of Faulkner Information Services, relying heavily on research from the Sun Microsystems Corporation, provides a more modern breakdown of the causes of unplanned downtime reflecting improvements in hardware and software reliability [11]. Barr considers three factors as contributing to unplanned system downtime: Process, People and Product. Barr states that Process and People each account for 40 percent of unplanned downtime and Product accounts for the remaining 20 percent. Barr defines unplanned Product downtime to include: hardware, software and environmental failures. The comparison of Wood's and Barr's causes of unplanned downtime demonstrates the trend for vendor software, hardware and environmental reliability improvements to improve overall availability while driving the causes of unplanned downtime more toward their customers' implementations.

Availability Cost

As with any system, there are two ways to improve the cost structure of the system: increase productivity and/or decrease expenditures. As implied by Wood's availability definition, computers and computer systems were created to improve the performance of our work – thus our productivity. The quality of the service provided by the system is an end-to-end statement of how well the system assisted in increasing our productivity.

The way to increase productivity of the user community is to increase the availability of the system in a reliable way. HA solutions provide cost benefits by pricing out downtime verses the cost of hardware, software and support. The way to decrease expenditures is to increase the productivity of the system support staff by increasing the system's reliability and serviceability.

Availability Levels

Increasing levels of availability protect different areas of the system and ultimately the business. Redundancy and catastrophic recovery are insurance policies that offer some availability gains. A project to increase availability would expect an availability verses investment graph to look similar to the one presented in Figure 2 (adapted from [12]) and can be viewed as having four distinct levels of availability: no HA mechanisms; data redundancy to protect the data; system redundancy and fail-over to protect the system; and disaster recovery to protect the organization. As you move up the availability index, the costs are cumulative as the graph assumes the HA components are integrated in the system in the order presented.

At the basic system level, no availability enhancements have been implemented. The method used of data redundancy will be some form of backup normally to tape. System level failure recovery is accomplished by restoration from backup. The contingency planning for disaster recovery is most often what has been called the "Truck Access Method" (TAM) or a close variant. TAM is briefly discussed in the next section.

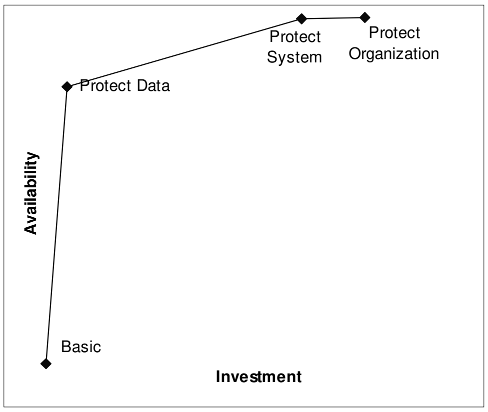

Figure 2: Availability index.

Data protection provides the largest cost benefit of all availability levels. Data protection is important for three reasons. First, data is unique. Second, data defines the business. Finally, the storage media housing data is the most likely component to fail. At the data protection level, a redundant array of inexpensive disks (RAID) [13] solution as appropriate for the environment is implemented. RAID solutions protect the data from loss and provide availability gains during the reconstruction of the data. Volume management software can be added to automate and enhance many of the system administration functions associated with the RAID disk abstraction.

At the system protection level, redundant HW components are added to enhance the ability to recover after a failure through system reconfiguration and fault tolerance. Automatic system reconfiguration (ASR – logically removing the failed components) and alternate pathing of networks and disk subsystems allow the recovery from hardware failures and coupled with a locally developed or commercial HA solution enables automatic detection and recovery (fail-over) from software service failures.

Disaster recovery is protection for the organization from a business or program-ending event. Disaster recovery differs from continuity of operations in that continuity of operations is proactive and common solutions often rely on transactionally aware duplicate of the production data center to a remote site. Disaster recovery is reactionary. DR assumes some loss of availability and data is acceptable and obtains a cost benefit for this assumption. The differences between the two therefore are cost, reliability and system availability.

Disaster Recovery

DR is a complex issue. DR forces an organization to prepare for events that no one wants to or really can prepare for and to discuss issues that few people are comfortable discussing, such as the loss of key personnel. This paper focuses only on the physical problem of moving critical functions from a data center to a DR site quickly and safely, which is only a portion of disaster contingency planning. There are several sources [14, 15] that discuss surviving and prioritizing [16] a computer disaster from a management perspective.

Kahane, et al. [17] define several solutions for large computer backup centers. The solutions differ by response time, cost and reliability. Among these solutions:

1. Hot Backup – Maintaining an additional site that operates in parallel to the main installation and immediately takes over in case of a failure of the main center.
2. Warm Backup – Maintaining another inactive site ready to become operational within a matter of hours.
3. Cold Backup – Maintaining empty computer premises with the relevant support ready to accept the immediate installation of appropriate hardware.
4. Pooling – A pooling arrangement where a few members join into a mutual agreement concerning a computer center, which is standing-by idle to offer a service to any member suffering from interruption in its computer center. The idle computer center may be employed in the form of ‘‘cold’’ or ‘‘warm’’ backup.

The focus of this project is the creation of a ‘‘warm backup’’ site that integrates an easily testable DR capability into Data Center operations.

Disaster Recovery Approaches

There are generally two common extremes to continuing operations planning, restoration from tape and a fully replicated synchronous backup facility.

The simplest method of DR preparation has been called the ‘‘Truck Access Method’’ (TAM) or a close variant. For TAM, periodic backups of all systems and data are made and stored at a safe and secure off-site facility. The advantage of this method is cost, but there are three main disadvantages affecting reliability and availability. First, the number of tapes can be quite large. A single bad or mislabeled tape can hamper DR extensively. Second, procurement and installation of infrastructure components is time-consuming and anything short of replication of the production data center greatly complicates restoration from tape. Full restoration from tape is also very time consuming. Lastly, testing the DR procedures is complicated and can result in downtime. Extensions to the TAM method include the use of a commercial DR facility or backup pool [17] or the construction of a redundant data center.

At the other extreme, a remote backup facility can be constructed for DR where order-preserving transactions are used to keep the primary and the backup data synchronous [18, 19]. This approach often involves the addition of front-end processors and a mainframe at the remote site. Additionally, the communication link between the data center and remote site will be expensive and potentially can degrade performance of the host applications. In designing a synchronous backup facility, replication is the driving consideration. Wiesmann, et al. [20] provides an overview of replication in distributed systems and databases along with providing a functional model that can be used in designing a specific replication solution.

A common compromise is to employ data vaulting [21] where commercially available data replication solutions mirror or shadow the data to an alternate location, reducing recovery time but mostly reducing risks.

HA Technologies

Replicated hardware solutions have traditionally been used to provide fault tolerance [22]. Fault tolerance refers to design techniques such as error correction, majority-voting, and triple modular redundancy (TMR), which are used to hide module failures from other parts of the system. Pfister [23] and Laprie, et al., [24] can provide the reader more background on fault tolerant systems. Fault Tolerant systems can be classified as primarily hardware or primarily software. Hardware fault tolerant systems tend to be more robust with quicker recovery time from faults but tend to be more expensive. Software systems create very fast recovery by providing a method of migrating a process and its state from the failed node to a fail-over node. Milojicic, et al. [25] provides a survey of the current state of process migration. Fault-tolerant systems are synchronous and the low latency requirements between systems make their use for DR impractical.

HA systems have the distinction from fault tolerant systems in that they recover from faults not correct them. A latent bug in application code which causes a fault is unlikely to re-occur after a fail-over in a HA solution as the application will be re-initialized. This distinction makes HA solutions the preferred solution for software failures and most operational failures. Fault tolerant systems are generally used in mission critical computing where faults must be masked and are often components in HA systems.

The basic model for building a HA system is known as the primary-backup [26] model. In this model, for each HA service one of the servers is designated as the primary and a set of the others are designated as backups. Clients make service requests by sending messages only to the primary service provider. If the primary fails, then a fail-over occurs and one of the backups take over offering the service. The virtues of this approach are its simplicity and its efficient use of computing resources. Servers providing backup services may be ''hot-standbys'' or themselves providers of one or more primary services. The primary-backup model of HA system provides a good model for the creation of a ''warm backup'' site.

In building an HA system, ensuring data integrity in persistent storage during the fail-over process is the most important criteria, even more important than availability itself. In general, there are two methods for providing a shared storage device in a HA cluster: direct attached storage or some form of network storage.

Direct attached storage is the most straightforward persistent storage method using dual-ported disk. The issue is one of scalability. As an HA cluster requires disk connection to all primary and fail-over nodes, clusters of greater than four to eight nodes cannot support direct attached persistent storage and require disk systems be deployed in some form of a storage network.

Storage Area Network (SAN) environments are dedicated networks that connect servers with storage devices such as RAID arrays, tape libraries and Fiber Channel host bus adapters, hubs and switches. Fiber Channel SANs are the most common SANs in use today [27]. Fiber Channel SANs offer gigabit performance, data mirroring and the flexibility to support up to 126 devices on a single Fiber Channel-Arbitrated Loop (FC-AL) [28]. SANs are a maturing technology as such there are numerous developing standards and alliances for Fiber Channel SAN design. Furthermore, distance limits for a SAN are the same as the underlying technologies upon which it is built. An emerging standard, Fiber-Channel over TCP/IP (FCIP) [29] has been proposed to the Internet Engineering Task Force (IETF). FCIP offers the potential to remove the distance limitations on SANs.

An alternative to SAN environments is NAS (Network Attached Storage) networks. NAS networks also connect servers with storage devices but they do so over TCP/IP via existing standard protocols such as CIFS (Common Internet File System) [30] or NFS (Network File System) [31].

The use of a storage network to provide data access during fail-over is sufficient for HA but does not provide for the separation of resources necessary in DR. DR needs a copy of the data at a remote location. The addition of a data mirroring capability is a potential solution. The mirroring process of the persistent storage can be synchronous, asynchronous or logged and resynchronized. Local mirroring, also known as RAID 1, consists of two disks that synchronously duplicate each other's data and are treated as one drive. One solution to providing a remote copy of the data would be to stretch the channel over which the data is mirrored. A single FC-AL disk subsystem can be up to 10 kilometers [28] from the host system. For some disaster contingency planning, ten kilometers may be sufficient. Channel extenders offer potential distances greater than ten kilometers [21].

An alternative to synchronous mirroring is asynchronous mirroring (also known as shadowing). In asynchronous mirroring, updates to the primary disk and mirror are not atomic, thus the primary and mirror disk are in different states at any given point in time. The advantage of asynchronous mirroring is a reduction in the required bandwidth as real-time updates are not required and a failure in the mirror does not affect the primary disk. The two basic approaches to asynchronous mirroring are: to take a ''snapshot'' of the primary data and use a copy-on-write [32] mechanism for updating both the mirror and the primary data; or to log updates [33] to the primary data over a period of time then transfer and apply the log to the mirror. The result of asynchronous mirroring is that the data and the mirror are synchronized at a point in the past.

Commercial HA Solutions

There have been small investigations [35] into the formal aspects of the primary-backup system but the traditional two to a few node HA systems have been widely used commercially. Many commercial cluster platforms support fail-over, migration, and automated restart of failed components, notably Compaq, HP, IBM, and Tandem [23], Sun's Full Moon [36] and Microsoft's Cluster Service [37]. All of the commercial cluster platforms mentioned offer only single-vendor proprietary solutions. Veritas Cluster [38] offers a multi-vendor HA solution.

Commercial products like BigIP [39] from F5Networks or TurboLinux's TurboCluster [40] have been introduced where clustering is used for load balancing across a cluster of nodes to provide system scalability with the added benefit of high availability. These systems are employing novel approaches to load balancing across various network components and IP protocols. The use of clustering across geographically distributed areas is gaining support for building highly available Web Servers [42, 43, 44]. The geographic distribution of the Web Servers enhances high availability of the Web interface and provide for virtually transparent disaster recovery of the Web Server. The DR issue with the design is with the back-end processor. In the Iyengar, et al. [43] article, the back-end processor was an IBM large-scale server in Nagano, Japan without which the Web Servers provide only static potentially out of date data.

High Availability and Disaster Recovery

On the surface, DR would seem to be an extension of HA. HA's goal is not only to minimize failures but also to minimize the time for recovery from them. In order to asymptotically approach 100% availability, fail-over services are created. One DR strategy would be to create a fail-over node at an appropriate off-site location by extending the communications between the clustered systems over geographic distances

However, DR and HA address very different problems. Marcus and Stern [12] made four distinctions. First, HA servers are colocated due to disk cable length restrictions and network latency; DR servers are far apart. Second, HA disk and subnets are shared; DR requires servers with separate resources. Third, HA clients see a fail-over as a reboot; DR clients may be affected also. Finally, HA provides for simple if not automatic recovery; DR will involve a complex return to normalcy.

Furthermore, commercial HA solutions assume adequate bandwidth, often requiring dedicated redundant 10 or 100 Megabit channels for a ''heartbeat.'' Data center performance requirements often require Gigabit channels for disk access. Even if network latency and disk cable length restrictions can be overcome with channel extension technologies [21] and bandwidth, the recurring communications cost associated with providing the required bandwidth to support HA clusters over geographic distances is currently prohibitive for most organizations.

The level to which HA technologies can be cost effectively leveraged in a DR solution offers some simplification and risk reduction of the DR process. In order to cost effectively use HA technologies in DR, the high bandwidth communication channels must be replaced with low bandwidth usage. Our focus is to minimize required communications between the primary and the backup, efficiently utilize the available bandwidth and rely on IP QoS mechanisms to insure a stable operational communications bandwidth. The next subsection provides an overview of IP QoS mechanisms.

IP Quality of Service

In order to provide end-to-end QoS, QoS features must be configured throughout the network. Specifically, QoS must be configured within a single network element, which include queuing, scheduling, and traffic shaping. QoS signaling techniques must be configured for coordinating QoS from end-to-end between network elements. Finally, QoS policies must be developed and configured to support policing and the management functions necessary for the control and administration of the end-to-end traffic across the network.

Not all QoS techniques are appropriate for all network routers as edge and core routers perform very different functions. Furthermore, the QoS tasks specific routers are performing may also differ. In general, edge routers perform packet classification and admission control while core routers perform congestion management and congestion avoidance. The following QoS overview of router support is biased toward what is available as part of Cisco's IOS as Cisco routers are used in the case study environment. Bhatti and Crowcroft provide a more general overview of IP QoS [45].

Three levels of end-to-end QoS are generally defined by router vendors, Best Effort, Differentiated and Guaranteed Service [46]. Best Effort Service, (a.k.a. lack of QoS) is the default service and is the current standard for the Internet. Differentiated Service (a.k.a. Soft QoS) provides definitions that are appropriate for aggregated flows at any level of aggregation. Examples of technologies that can provide differentiated service (DiffServ) in an IP environment are Weighted Fair Queuing (WFQ) with IP Precedence signaling or, under IOS, Priority Queuing (PQ) when only a single link is required [47]. Guaranteed Service (a.k.a. hard QoS) is the final QoS level as defined. Guaranteed Service provides a mechanism for an absolute reservation of network resources. Integrated Services (IntServ) guaranteed service could be configured using hard QoS mechanisms, for example, WFQ combined with Resource Reservation Protocol (RSVP) [48] signaling or Custom Queuing (CQ) on a single link [49] in a Cisco IOS environment.

In a router environment, end-to-end QoS levels are implemented using features provided as part of the router's operating system. These features typically fall into five basic categories: packet classification and marking, congestion management, congestion avoidance, traffic conditioning and signaling. In a Cisco router environment, the Internetworking Operating System (IOS) provides these QoS building blocks via what Cisco refers to as the ''QoS Toolkit'' [50].

QoS policies are implemented on an interface in a specific sequence [51]. First, the packet is classified. This is often referred to as coloring the packet. The packet is then queued and scheduled while being subject to congestion management techniques. Finally, the packet is transmitted. Packet classification is discussed next, followed by a discussion of queuing and scheduling. Congestion avoidance techniques are used to monitor the network traffic loads in an effort to identify the initial states of congestion and proactively avoid it. Congestion avoidance techniques are not used in this project and will not be discussed further. The IP Quality of Service section proceeds with a discussion of traffic shaping and policing; concluding with a discussion of RSVP signaling.

In order to implement any QoS strategy using the QoS toolkit, the router and version of IOS must support the features used. Cisco provides a matrix of IOS versions, routers and QoS features for cross-reference [52].

Packet Classification and Marking

Packet classification occurs by marking packets using either IP Precedence or the DiffServ Code Point (DSCP) [46]. IP Precedence utilizes the three precedence bits in the IP version 4 header's Type of Service (ToS) field to specify class of service for each packet. Six classes of service may be specified. The remaining two classes are reserved. The DSCP replaces the ToS in IP version 6 and can be used to specify one of 64 classes for a packet. In a Cisco router, IP Precedence and DSCP packet marking can be performed explicitly through IOS commands or IOS features such as policy-based routing (PBR) and committed access rate (CAR) can be used for packet classification [53].

PBR [54] is implemented by the QoS Policy Manager (QPM) [51]. PBR allows for the classification of traffic based on access control list (ACL). ACLs [55] establish the match criteria and define how packets are to be classified. ACLs classify packets based on port number, source and destination address (e.g., all traffic between two sites) or Mac address. PBR also provides a mechanism for setting the IP Precedence or DSCP providing a network the ability to differentiate classes of service. PBR finally, provides a mechanism for routing packets through traffic-engineered paths. The Border Gateway Protocol (BGP) [56] is used to propagate policy information between routers. Policy propagation allows packet classification based on ACLs or router table source or destination address entry use with IP Precedence.

CAR [57] implements classification functions. CAR's classification service can be used to set the IP Precedence for packets entering the network. CAR provides the ability to classify and reclassify packets based on physical port, source or destination IP or MAC address, application port, or IP protocol type, as specified in the ACL.

Congestion Management

Congestion Management features are used to control congestion by queuing packets and scheduling their order of transmittal using priorities assigned to those packets under various schemes. Giroux and Ganti provide an overview of many of the classic approaches [58]. Cisco's IOS implements four queuing and scheduling schemes: first in first out (FIFO), weighted fair queuing (WFQ), custom queuing (CQ) and priority queuing (PQ). Each is described in the following subsections [53].

First In First Out (FIFO)

In FIFO, there is only one queue and all packets are treated equally and serviced in a first in first out fashion. FIFO is the default queuing mechanism for above E1 (2.048 Mb/s) Cisco routers and is the fastest of Cisco's queuing and scheduling schemes.

Weighted Fair Queuing

WFQ provides flow-based classification to queues via source and destination address, protocol or port. The order of packet transmittal from a fair queue is determined by the virtual time of the delivery of the last bit of each arriving packet. Cisco's IOS implementation of WFQ allows the definition of up to 256 queues.

In IOS, if RSVP is used to establish the QoS, WFQ will allocate buffer space and schedule packets to guarantee bandwidth to meet RSVP reservations. RSVP is a signaling protocol, which will be discussed later in this section, the largest amount of data the router will keep in queue and minimum QoS to determine bandwidth reservation.

If RSVP is not used, WFQ, like CQ (see Custom Queuing), transmits a certain number of bytes from each queue. For each cycle through all the queues, WFQ *effectively* transmits a number of bytes equal to the precedence of the flow plus one. If no IP Precedence is set, all queues operate at the default precedence of zero (lowest) and the scheduler transmits packets (bytewise) equally from all queues. The router automatically calculates these weights. The weights can be explicitly defined through IOS commands.

Priority Queuing

In Cisco's IOS, PQ provides four queues with assigned priority: high, medium, normal, and low. Packets are classified in to queues based on protocol, incoming interface, packet size, or ACL criteria. Scheduling is determined by absolute priority. All packets queued in a higher priority queue are transmitted before a lower priority queue is serviced. Normal priority is the default if no priority is set when packets are classified.

Custom Queuing

In Cisco's IOS, CQ is a queuing mechanism that provides a lower bound guarantee on bandwidth allocated to a queue. Up to 16 custom queues can be specified. Classification of packets destined for a queue is by interface or by protocol. CQ scheduling is weighted round robin. The weights are assigned as the minimum byte count to be transmitted from a queue in a given round robin cycle. When a queue is transmitting, the count of bytes transmitted is kept. Once a queue has transmitted its allocated number of bytes, the currently transmitting packet is completed and the next queue in sequence is serviced.

Traffic Policing and Shaping

Policing is a non-intrusive mechanism used by the router to ensure that the incoming traffic is conforming to the service level agreement (SLA). Traffic Shaping modifies the traffic characteristics to conform to the contracted SLA. Traffic shaping is fundamental for efficient use of network resources as it prevents the drastic actions the network can take on non-conforming traffic, which leads to retransmissions and therefore inefficient use of network resources. The traffic shaping function implements either single or dual leaky bucket or virtual scheduling [59, 60, 61].

Signaling Mechanisms

End-to-end QoS requires that every element in the network path deliver its part of QoS, and all of these entities must be coordinated using QoS signaling. The IETF developed RSVP as a QoS signaling mechanism.

RSVP is the first significant industry-standard protocol for dynamically setting up end-to-end QoS across a heterogeneous network. RSVP, which runs over IP, allows an application to dynamically reserve network bandwidth by requesting a certain level of QoS for a data flow across a network. The Cisco IOS QoS implementation allows RSVP to be initiated within the network using configured proxy RSVP. RSVP requests the particular QoS, but it is up to the particular interface queuing mechanism, such as WFQ, to implement the reservation. If the required resources are available and the user is granted administrative access, the RSVP daemon sets arguments in the packet classifier and packet scheduler to obtain the desired QoS. The classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream. If either resource is unavailable or the user is denied administrative permission, the RSVP program returns an error notification to the application process that originated the request [62].

Study Results

The case study was intended to evaluate the feasibility and production implementation options for the proposed DR solution. Our design sought to minimize communications requirements through data shadowing, exploitation of file commonality in file updates, network traffic shaping and to ensure system stability through IP QoS. Our prototype sought to measure the impact of each of the communication limiting techniques. The measurements were carried out in three distinct evaluations.

- The first evaluation was to determine the effect of block level updates verses file updates.
- The second evaluation was to determine the level of network bandwidth efficiency reasonably achievable.
- The third and final evaluation was establishing a configuration that supports the required QoS.

The test environment is presented next. Followed by the evaluations carried out and the issues they revealed. This section concludes with the results of the evaluations.

The Test Environment

A test environment was configured as shown in Diagram 3 and was constructed in as simple a manner as possible to reduce the cost of the evaluation. The entire test environment took about five days to install and configure, given that the infrastructure was already in place. Operating Systems and applications are loaded on the DR servers and updates are made manually. The baseline testing took about 90 days to gather the data. In the test environment, the OS was Solaris 7 and the applications were Oracle, PVCS and local configuration management database applications. This was the most labor-intensive part of the test setup. One of the production servers was used to

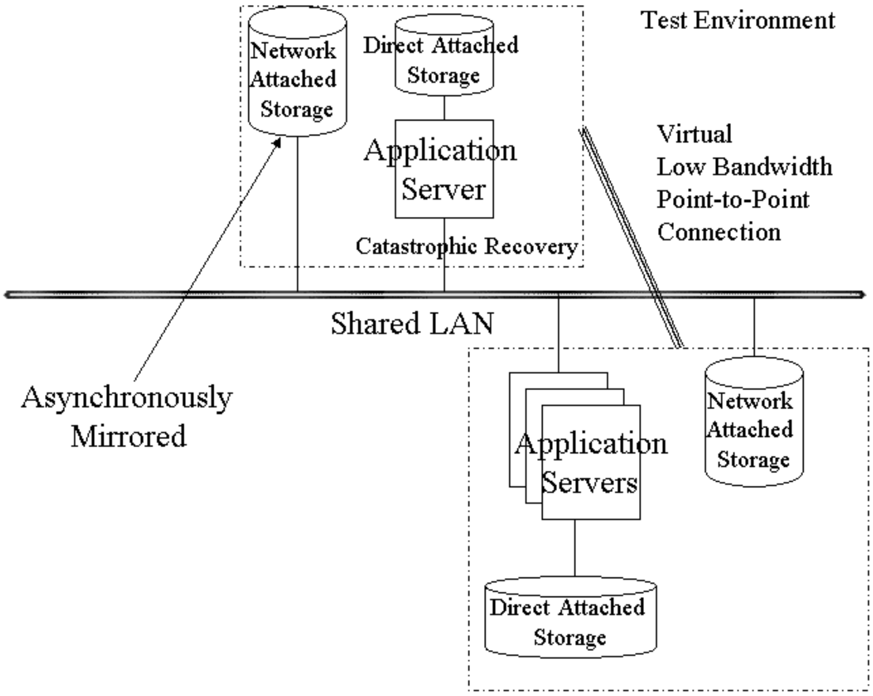

Diagram 3: Test environment.

maintain the heartbeat between the test systems; and initiate and monitor the asynchronous mirror updates. An available Network Appliance filer was setup to store the backup data. The primary data used in the test was restricted to NAS accessed data. This allowed the use of only one commercial data-shadowing product, reducing the cost and complexity of the test. The commercial data shadowing product chosen was SnapMirror [63] from Network Appliance. The filer and SnapMirror software were configured in approximately a day.

SnapMirror uses the snapshot [64] facility of the WAFL file system [65] to replicate a volume on a partner file server. SnapMirror is used in this study to identify changes in one of the production data volumes and resynchronize the remote volume with the active. After the mirror synchronization is complete, the backup data is at the state of the primary data at the instant of the snapshot. The remote mirrors are read-only volumes and are not accessible except by the source file servers. When a fail-over is to occur, the backup volumes are manually switched from a read-only standby state to read-write active state and rebooted. After the reboot, the backup filer is accessible by the DR server and the remote data can be mounted. SnapMirror and Network Appliance filers were chosen for this test based on their current use in the production environment, the availability of a filer to use in the test, and their support of block level updates allowing a determination of the impact of a block level verses file level update policy. The amount of data used in the test was constrained by the available disk storage on the filer, 120 GB.

The production application servers, the production NAS filers, the DR application server and the DR filer were connected to the shared production Gigabit Ethernet LAN. The production and DR filers along with the production application servers also have a second interface which is attached to a maintenance LAN running switched fast Ethernet. The asynchronous mirroring was tested on the production LAN to look for impacts and then reconfigured to run over the maintenance LAN. The heartbeat and synchronization initiation was carried out over the production LAN.

Evaluations

The first evaluation was to determine the effect of block level updates verses file updates. This test consisted of mirroring asynchronously approximately 100 GB of production data for 52 days and measuring the volume of block level updates required for the synchronization. The mirror synchronization was initiated daily at 3 pm, just after the peak load (see Chart 1). Chart 2 shows the weekday daily change rate as a percentage of total data monitored. The mean percentage daily rate of change was 2.32% with the minimum daily rate of change being 1.12% and a maximum daily change rate of 3.69%.

The sizes of the files that were modified over the update period were summed. This test was accomplished by running a perl script over the snapshot data used by SnapMirror. Block level updates show a reduction of approximately 50% of data required for transfer verses uncompressed copying of the modified files.

The second evaluation was to determine the level of network bandwidth efficiency reasonably achievable. The mirrored data was traffic shaped at the data source using a leaky bucket algorithm provided with the SnapMirror product. The data shadowing traffic was measured at the interface of the DR filer during the synchronization process. A threshold value (the hole in the bucket) of five megabits/second was set creating the virtual low bandwidth connection depicted in Diagram 3 within the LAN's Ethernet channel.

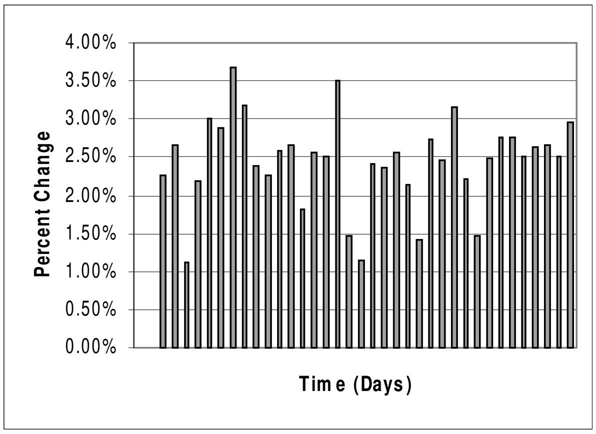

Chart 2: Daily data change rates.

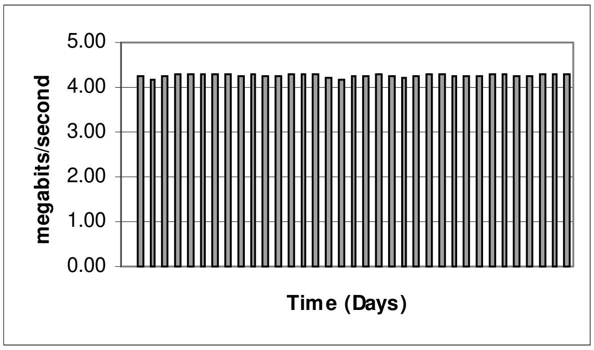

Chart 3: Weekday network throughput.

The asynchronous mirroring occurred over the production LAN where there is significant excess bandwidth. No other IP QoS mechanisms were used at this point in order to see if a constant load could be achieved and what impact the addition of this constant load would place on the Data Center filers and on the LAN. The rate of five megabits/second was selected as it was expected that a low bandwidth channel of less that five megabits/second would be required to update the production DR site given the current one terabyte requirement. The mean effective throughput for the test was 3.99 megabits/second.

When looking at the data, it became obvious the overhead of determining the updates to the volume and read/write times were significant during periods of low data volume change which occur every weekend. In addition, the larger the data transferred in the update, the higher the network throughput. Removing data from each Saturday and Sunday yields an increase in the mean throughput to 4.26 megabits/second, which gives a bandwidth efficiency of 85%. Chart 3 graphs the weekday throughput of the test. As noted in chart 3, the throughput is consistent implying efficient network utilization. Throughput and peak network loads were measured before, during and after the synchronization on weekdays. Throughput, as expected, was increased proportionally to the network traffic added. Peak network loads were unaffected.

The second evaluation demonstrated two key capabilities. First, the data could be successfully synchronized between the primary and a remote DR site over a low bandwidth channel, in this case five Mbps. Secondly, the data required for this synchronization could be throttled to efficiently use a low bandwidth channel or provide a minimal impact on a shared higher bandwidth channel.

The third and final evaluation was establishing a configuration that supports the required QoS. The issue is to ensure that shaped data transmitted from the source filer is transmitted to the DR filer without additional delays. If the network cannot support the data rate of the source filer, the network acts as an additional queue and introduces delay. This delay introduces jitter into the shaped data that may prevent the synchronization of the data within the fidelity of the DR requirements. For example, if the DR requirement is to synchronize data hourly, the additional delay may cause the synchronization to take more than one hour and what is the result? Does the current synchronization fail and the next initiation of synchronization start, which could result in never successfully synchronizing the data? The proposed solution is two fold. First, initiation of a data synchronization does not occur until the completion of the previous data synchronization. If a synchronization has to wait, it is initiated as soon as the previous synchronization completes. This check was added to the heartbeat, but was also discovered to be a feature of the SnapMirror product.

Secondly, to prevent the additional delays in the network, IP QoS mechanisms can be used to provide a guarantee of adequate bandwidth based on the traffic shaping threshold. As previously described, many configuration options could be used to meet the QoS requirements. In the case study, the SnapMirror product was reconfigured to asynchronously mirror over the maintenance network. Custom queuing was enabled on the interface to the source filer and configured to guarantee 5% of the 100 Mbps link to the mirror process. The maintenance network is primarily used for daily backup to tape, thus its traffic is bursty and heavily loaded during off-peak hours (8 pm-6 am). Network traffic continues to be measured at the interface of the DR filer and has continued to operate around the 4.26 Mbps level. An excerpt from the IOS configuration used in the test follows:

1. interface serial 0
2. custom-queue-list 3
3. queue-list 3 queue 1 byte-count 5000
4. queue-list 3 protocol ip 1 tcp 10566
5. queue-list 3 queue 2 byte-count 95000
6. queue-list 3 default 2.

Queues are cycled through sequentially in a round-robin fashion dequeuing the configured byte count from each queue. In the above excerpt, SnapMirror traffic (port 10566) is assigned to queue one. All other traffic is assigned to queue two. Entire packets are transmitted from queue one until the queue is empty or 5000 bytes have been transmitted. Then queue two is serviced until its queue is empty or 95000 bytes have been serviced. This configuration provides a minimum of 5% of the 100 Mbps link to the SnapMirror traffic.

Issues

Four issues arose during the proof-of-concept implementation. The first was ensuring data integrity. What happens if the communications line is lost, primary servers are lost, etc. during remote mirror synchronization?

The SnapMirror product was verified to ensure data integrity. Upon loss of network connectivity during a mirror resynchronization, the remote filer data remained as if the resynchronization has never began. However, an issue arose ensuring database integrity. This is a common problem when disk replication occurs via block level writes. The problem is the database was open during the snapshot operation so transactions continue. Furthermore, the redo logs were based from when the database last performed a "hot backup" or was restarted; thus, could not be applied to the backup. A solution to accomplish database synchronization is to actually shutdown the database long enough to take the snapshot. The database and the redo logs are then restarted. In the case study environment, the entire process takes about four minutes. Liu and Browning [34] provide details covering the backup and recovery process used. An alternative would be to purchase a disk replication package specific for the database.

The second issue was licensing. Several commercial products required for the DR functions use a licensing scheme based on the hostid of the primary system. The license key used for the primary installation could not be used for the backup installation and proper additional licenses had to be obtained and installed.

The third issue arose in the QoS testing and results from excess bandwidth and a reliance on production data to test the design. The maintenance network is lightly loaded during prime shift, from 6 am to 8 pm. The daily synchronization is initiated at 3 pm and normally completes in about 75 minutes. Since the maintenance network has excess bandwidth during the synchronization, it is possible that the custom queuing configuration has no effect as is indicated by the data.

This demonstrates the point that the configuration of bandwidth reservation through IP QoS mechanisms is only required on the local LAN when the local LAN does not have excess bandwidth capacity greater than that of the low bandwidth connection used for backup site communications. In most cases, the bandwidth reservation is insurance that the required bandwidth will always be available enabling the low-bandwidth link to be fully utilized. The issue of excess bandwidth in the testing environment is further evidenced when a synchronization was attempted without traffic shaping enabled. Intuitively, the peak load induced when a transfer of greater than two GB is initiated through a five Mbps queue would slow the transfer down. However, it did not as queue one's data continued to be serviced as long as there was no other network traffic.

The resolution of this testing issue is more difficult. In order to get valid test results for the quantity and types of changes, a production data volume were used. This requires non-intrusive testing on the production LANs. While testing on the maintenance network during backups would be useful to this project, it may also prevent production work from completing and has not currently been undertaken.

The final issue was security. Security of the remote servers, the remote mirror and communications is a topic, which must be further addressed. In the test, standard user authentication provides security on the remote servers and remote filers. Additionally, a configuration file, /etc/snapmirror.allow located on the primary filer, provides a security mechanism ensuring only the backup filers can replicate the volumes of the primary filers. The communication channels were over existing secure links. These secure links may not be available in the final target DR site.

Data Evaluation

The final task of the test is to determine the communication requirements to enable a stable geographically distributed storage update policy over a low bandwidth link. Since not all the data in the Data Center could be used in the prototype due to storage limitations, a linear estimation was used to predict the time required to perform the data synchronization. The assumptions of the model are:

- The amount of data changed is related to the amount of data in use.
- The amount of transferred data directly contributes to the length of time required for data synchronization.
- The relation between these two factors and the time required for data synchronization is linear.

Using the results from the sample data, 85% network efficiency allows a maximum of 13.84 GB of data to be transferred per 24 hours over a dedicated 1.544 Mbps T1 link. Under the assumption of the 3.69% maximum daily change rate, a maximum data store of 375 GB is supported by this design with a 24-hour synchronization policy. Under the assumption of the 2.32% mean daily change rate, a maximum data store of 596 GB can be supported. In order to support the required one TB, a minimum of a 4.22 Mbps link is required for the maximum daily change rate and a minimum of a 2.65 Mbps link is required for the mean daily change rate.

Summary and Future Work

This paper proposes a design that is the integration of several existing HA and network efficiency techniques, disk replication products and current IP QoS mechanisms, to establish an off-site DR facility over a low-bandwidth connection. The paper evaluates an approach to minimizing the communication requirements between the primary and backup site by relying on block level updates by the disk replication products to exploit file commonality in file updates; network traffic shaping and data shadowing to enable efficient network communications; and IP QoS mechanisms to insure that adequate bandwidth is available to ensure efficient usage of the low bandwidth link and that data synchronization can occur within the constraints of the DR requirements.

The proof-of-concept test developed for the case study demonstrated the functionality of the design over a reasonably low bandwidth connection of five Mbps and also demonstrated that a dedicated T1 link was insufficient given a 24-hour update cycle of one TB of data with the derived set of usage parameters. The proof-of-concept also demonstrated several other points about the design. First, the gain from the exploitation of file commonality can be significant but is of course usage dependent. In general, data replication products do not support block level updates and if they do, within a single product line. A more generic solution appears to be the exploitation of commonality at the file abstraction level where data compression and the integration of security mechanisms such as Internet X.509 [41] can be used for additional reductions in required bandwidth and increased security. Secondly, traffic shaping the data was demonstrated to be a highly effective method to efficiently use the available communications on low-bandwidth links. Finally, as stated in the previous section, the testing of the bandwidth guarantees is incomplete, difficult to measure and only required when excess bandwidth is not available or more generally put, as an insurer of available bandwidth. Bandwidth reservations are most likely to be required when communications are over a heavily used or bursty WAN.

The next steps for this project take two distinct tracts. The first involves adding additional remote disk capacity, securing an appropriate remote link with a SLA of a minimum of five Mbps and testing additional disk replication products to support the full data set required at the DR site. The second is investigating the feasibility of providing an enhancement that offers support for asynchronous mirroring of only the modified areas of raw data in a compressed, secure manner, exploiting file commonality and further reducing bandwidth requirements. An enhancement or extension to the Network Data Management Protocol (NDMP) is being explored as a possible solution.

Author Info

Kevin Adams is a lead Scientist for the Submarine Launched Ballistic Missile program at the Dahlgren Division of the Naval Surface Warfare Center. He has been involved with Unix system administration since 1987. Kevin holds a B.S. in Computer Science from James Madison University and an M.S. in Computer Science and an M.S. in Electrical Engineering from Virginia Tech. Reach him via U. S. Mail at NSWCDD; Code K55; 17320 Dahlgren Road; Dahlgen VA 22448-5100. Reach him electronically at AdamsKP@nswc.navy.mil.

References

[1] Gigabit Ethernet Alliance, 10GEA White Papers, Gigabit Ethernet:1000BaseT, http://www.10gea.org/GEA1000BASET1197_rev-wp.pdf, November 1997.

[2] Alvarado, M. and P. Pandit, "How Convergence Will End the SAN/NAS Debate," *DM Review*, http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=3016, February 2001.

[3] Blackburn, D. and D. Driggs, *Sun Sets Agenda on Availability with new SunUP Program*, Press Release, Palo Alto, CA, http://www.sun.com/smi/Press/sunflash/1999-02/sunflash.990209.3.html, February 9, 1999.

[4] Kleiman, S. R., S. Schoenthal, A. Row, S. H. Rodrigues, and A. Benjamin, "Using NUMA interconnects for highly available filers," *IEEE Micro*, Vol. 19, Issue 1, pp. 42-48, Jan-Feb 1999.

[5] McDougall, R., "Availability – What It Means, Why It's Important, and How to Improve It," *Sun Blueprints Online*, http://www.sun.com/solutions/blueprints/1099/availability.pdf, October 1999.

[6] Scott, D., J. Krischer, J. Rubin. *Disaster Recovery: Weighing Data Replication Alternatives*, Research Note T-13-6012, Gartner Research, June 15 2001.

[7] Sun Microsystems White Paper, *Sun Ray1 Enterprise Appliance Overview and Technical Brief*, August 1999.

[8] Sun Microsystems Technical White Paper, *Sizing Sun Ray Enterprise Servers*, January 2000.

[9] Sun Microsystems Technical White Paper, *Integrating Sun Ray 1 Enterprise Appliances and Microsoft Windows NT*, January 2000.

[10] Wood, A., "Predicting client/server availability," *IEEE Computer*, Vol. 28 Issue 4, pp. 41-48, April 1995.

[11] Barr, J. G., *Choosing a High-Availability Server*, Docid: 00018376, Faulkner Information Services, 2001.

[12] Marcus, E. and H. Stern. *Blueprints for High Availability: Designing Resilient Distributed Systems*, pp. 5-6 & 298-300, John Wiley & Sons, Inc., 2000.

[13] Patterson, D. A., G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," *Proceedings of the Conference on Management of Dat*a, pp. 109-116, Chicago, Illinois, June 01-03, 1988, United States.

[14] Hiles, A., "Surviving a Computer Disaster," *IEEE Engineering Management Journal*, Vol. 2 Issue 6, pp. 271-274, Dec. 1992.

[15] Powers, C. B., "Preparing for the worst," *IEEE Spectrum*, Vol. 33 Issue 12, pp. 49-54, Dec. 1996.

[16] Jorden, E., "Project Prioritization and Selection: The Disaster Scenario," *HICSS-32, Proceedings of the 32nd Annual Hawaii International Conference on*, Systems Sciences, 1999.

[17] Kahane, Y., S. Neumann, and C. S. Tapier, "Computer Backup Pools, Disaster Recovery, and Default Risk," *CACM*, Vol. 31, No. 1, January 1988.

[18] Gracia-Molina, H. and C. A. Polyzois, "Issues in Disaster Recovery," *IEEE Compcon Spring 1990, Intellectual Leverage, Digest of Papers, Thirty-Fifth IEEE Computer Society International Conference*, pp. 573-577, 1990.

[19] King, R. P., N. Halim, H. Garcia-Molina, and C. A. Polyzois, "Overview of Disaster Recovery for Transaction Processing Systems," *Distributed Computing Systems, Proceedings, Tenth International Conference on*, pp. 286-293, 1990,

[20] Wiesmann, M., F. Pedone, A. Schiper, B. Kemme, and G. Alonso, "Database Replication Techniques: A Three Parameter Classification," *Proceedings of the Nineteenth IEEE Symposium on Reliable Distributed Systems,* pp. 206-215, 2000.

[21] Green, R. E., "Safety First," *IEEE Information Technology 1990, Next Decade in Information Technology, Proceedings of the Fifth Jerusalem Conference on (Cat. No. 90TH0326-9)*, pp. 593-595, 1990.

[22] Schneider, F. B., "Implementing Fault Tolerant Services Using the State Machine Approach: A Tutorial," *Computing Surveys*, Vol. 22, Issue 4, pp. 299-319, December 1990.

[23] Pfister, G. F., *In Search of Clusters: The Coming Battle in Lowly Parallel Computing*, Prentice Hall, 1995.

[24] Laprie, J. C., J. Arlat, C. Beounes, and K. Kanoun, "Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures," *IEEE Computer*, Vol. 23 Issue 7, pp. 39-51, July 1990.

[25] Milojicic, D. S., F. Douglis, Y. Paindaveine, R. Wheeler, and S. Zhou, "Process Migration," *ACM Computing Surveys*, Vol. 32, No. 3, pp. 241-299, September 2000. 241-299.

[26] Alsberg, P. A. and J. D. Day, "A Principle for Resilient Sharing of Distributed Resources," *Proceedings of the Second International Conference of Software Engineering*, pp. 627-644, October 1976.

[27] Wong, B., "Storage Area Networks: A Blueprint for Early Deployment," *Sun Microsystems Blueprints Online*, http://www.sun.com/blueprints/0101/Storage.pdf, January 2001.

[28] Kovatch, M., "Fiber Channel-Arbitrated Loop: How Does It Compare With Serial Storage Architecture?" *NSWCDD White Paper*, http://www.nswc.navy.mil/cosip/aug98/tech0898-2.shtml .

[29] Rodriguez, E., M. Rajagopal, and R. Weber, *Fibre Channel Over TCP/IP (FCIP)*, Internet Draft RFC, http://www.ietf.org/internet-drafts/draft-ietf-ips-fcovertcpip-11.txt, 06/19/2002 (expires December 2002).

[30] Microsoft Corporation, *Common Internet File System*, http://msdn.microsoft.com/workshop/networking/cifs .

[31] Sandberg, R., D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network File System," *Proceedings of the Summer 1985 USENIX Conference*, pp. 119-130, Portland, OR, June 1985.

[32] Hitz, D., J. Lau, and M. Malcolm, *File System Design for an NFS File Server Appliance*, Network Appliance, Inc., Technical Report (TR) 3002, http://www.netapp.com/tech_library/3002.html .

[33] Rosenblum, M. and J. K. Ousterhout, "The Design and Implementation of a Log-Structured file system," *ACM Transactions on Computer Systems (TOCS)*, Vol. 10, No. 1, pp. 26-52, Feb. 1992.

[34] Liu, J. and J. Browning, *Oracle7 for UNIX: Backup and Recovery Using a NetApp Filer*, Network Appliance, Inc., Technical Report (TR) 3036.

[35] Bhide, A., E. N. Elnozahy, and S. P. Morgan, "A Highly Available Network File Serve," *USENIX Conference Proceedings*, USENIX, Dallas Texas, pp. 199-206, January 21-25, 1991.

[36] Khalidi, Y. A., J. M. Bernabeu, V. Matena, K. Shirriff, and M. Thadani, *Solaris MC: A Multi-Computer OS*, Sun Microsystems Laboratories, SMLI TR-95-48, November 1995.

[37] Vogels, W., D. Dumitriu, K. Birman, R. Gamache, M. Massa, R. Short, J. Vert, J. Barrera, and J. Gray, "The Design and Architecture of the Microsoft Cluster Service," *IEEE*, pp. 422-431, 1998.

[38] Veritas White Paper, *Veritas Cluster Server v 2.0 Technical Overview*, http://eval.veritas.com/downloads/pro/vcs20_techover_final_0901.pdf, September 2001.

[39] Brunt, B., F5 Networks White Paper, *Achieving High Availability: Quest Software's SharePlex & F5 Network's BIG-IP/3-DNS*, http://www.f5networks.com/solutions/whitepapers/WP_SharePlex_F5.pdf .

[40] TurboLinux White Paper, *Cluster Server 6*, http://www.turbolinux.com/products/tcs/cluster6_whitepaper.pdf, October 2000.

[41] Housley, R., W. Ford, W. Polk, and D. Solo, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, RFC 2459, IETF, January 1999.

[42] Dias, D. M., W. Kish, R. Mukherjee, and R. Tewari. "A Scalable and Highly Available Web Server," *IEEE Proceedings of COMPCON 1996*, pp. 85-92, 1996.

[43] Iyengar, A., J. Challenger, D. Dias, and P. Dantzig, "High-Performance Web Site Design Techniques," *IEEE Internet Computing*, pp. 17-26, March-April 2000.

[44] Cardellini, V., M. Colajanni, and P. S. Yu, "Geographic load balancing for scalable distributed Web systems," *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Proceedings, Eighth International Symposium on*, pp. 20-27, 2000,

[45] Bhatti, Saleem N. and Jon Crowcroft, "QoS-Sensitive Flows: Issues in IP Packet Handling," *IEEE Internet Computing*, pp. 48-57, July-August 2000.

[46] G. Armitage, *Quality of Service in IP Networks*, pp. 67-70, 84-87, 105-112, Macmillan Technical Publishing, 2000,

[47] Cisco IOS Documentation, *Quality of Service Solution Guide*, Implementing DiffServ for end-to-end Quality of Service, Release 12.2, pp. 371-392, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt7/qcfdfsrv.pdf .

[48] Braden, R., L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*, RFC 2205, September 1997.

[49] Cisco Internet White Paper, *Designing Service Provider Core Networks to Deliver Real-Time Services*, http://www.cisco.com/warp/public/cc/pd/rt/12000/tech/ipra_wp.pdf, January 2001.

[50] Cisco IOS Documentation, *Internetworking Technology Overview*, Chapter 49 Quality of Service Networking, Release 12.2, http://www.cisco.com/univercd/cc/td/doc/ cisintwk/ito_doc/qos.pdf .

[51] Cisco IOS 12 Documentation, *Using QoS Policy Manager*, Chapter 1, Planning for Quality of Service, http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/qos/qpm21/qpm21ug/ugintro.pdf .

[52] Cisco Internet White Paper, *Supported Devices and QoS Techniques for IOS Software Releases*, http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/qos/pqpm20/qprodev.pdf, July 2002.

[53] Cisco IOS Documentation, *Quality of Service Solutions Configuration Guide*, Release 12.2, pp. 1-84, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/qcfbook.pdf .

[54] Cisco Internet White Paper, *Policy-Based Routing*, http://www.cisco.com/warp/public/cc/techno/protocol/tech/plicy_wp.pdf, 1996.

[55] Cisco Internet White Paper, *CiscoWorks2000 Access Control List Manager 1.4*, http://www.cisco.com/warp/public/cc/pd/wr2k/caclm/prodlit/aclm_ov.pdf, 2002.

[56] Lougheed, K., Y. Rekhter, *A Border Gateway Protocol (BGP)*, RFC 1163, IETF, June 1990.

[57] Cisco IOS Documentation, *Committed Access Rate*, Release 11.12, http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/car.pdf .

[58] Giroux, Natalie and Sudhakar Ganti, *Quality of Service in ATM Networks: State-of-the-Art Traffic Management*, Prentice Hall, pp. 48-59, 101-109, 246, 1999,

[59] Butto, M., E. Cavalero, and A. Tonietti, "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks," *IEEE Journal on Selected Areas of Communications,* Vol. 9, No. 3, April 1991.

[60] Dittman, L., S. Jocobson, and K. Moth, "Flow Enforcement Algorithms for ATM Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April 1991.

[61] Gallasi, G., G. Rigolio, and L. Fratta, "ATM: Bandwidth Assignment and Enforcement Policies," *Proceedings of the IEEE Globecom*, paper 49.6, Dallas Texas, November 1989.

[62] Cisco IOS Documentation, *Quality of Service Solutions Configuration Guide*, Release 12.1, Signaling Overview, pp. 243-260, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt5/qcfsig.pdf .

[63] Brown, K., J. Katcher, R. Walters, and A. Watson. *SnapMirror and SnapRestore: Advances in Snapshot Technology*, Network Appliance, Inc., Technical Report (TR) 3043.

[64] Marchi, M. J. and A. Watson, *The Network Appliance Enterprise Storage Architecture: System and Data Availability*, Network Appliance, Inc., Technical Report (TR) 3065.

[65] Hitz, D., *A Storage Networking Appliance*, Network Appliance, Inc., Technical Report (TR) 3001.