ARCHITECTING AVAILABILITY AND DISASTER RECOVERY SOLUTIONS

Tim Read, Sun N1 Availability Engineering

Sun BluePrints™ OnLine — April 2006

Part No 819-5783-10
Revision 1.0, 3/31/06
Edition: April 2006

Please
Recycle

Table of Contents

Introduction 1
About This Document 1
Overview 2
Defining Availability Objectives 2
Improving Availability 2
Protecting Against Disasters 3

Service Availability 4
Service Availability Solutions 4
Single, Resilient Servers 4
Home-Grown Availability Solutions 5
Sun Cluster Software 5
Comparing Availability Solutions 7
Coping With Local Disasters 8
Quorum Devices and Three-Site Configurations 8
Volume Management State Replica Majority 9

Disaster Recovery Solutions 10
Backup And Recovery From Tape 10
Using Snapshots to Lower the Impact of Tape Backups 11
Backup and Recovery Performance 11
Remote Disk Mirroring 12
Host-Based Replication 13
Restrictions on Using Sun StorEdge Availability Suite Software With Parallel Databases 14
Using Point-in-Time Software for Fast Recovery 14
Storage-Based Data Replication 14
Data Replication Infrastructure 15
Transaction Monitors and Application Servers 17
Replicating Databases 18
Comparing Disaster Recovery Solutions 19
Metro Clusters 20

Integrating Availability and Disaster Recovery Using Sun Cluster Geographic Edition Software 21
Partnerships 21
Replicating Cluster Data to Protect a Service Against Disasters 22
Using Sun Cluster Geographic Edition Protection Groups 24

The Performance Impact of Separating Data Centers 26

Summary 30
About the Author 30
References 30

Chapter 1
Introduction

Companies world-wide are facing increasing legal requirements for both protecting and retaining data for auditing purposes. This business obligation is forcing IT departments to ensure that their availability and disaster recovery systems are sufficient to comply with these demands. In addition, as a result of the global economy, internal and external customers expect higher levels of service availability. Together, compliance and increased user demand are creating more pressure on tight IT budgets. Many IT departments try to minimize costs by combining an availability solution with a disaster recover strategy.

IT departments typically run four broad classes of service in the data center: mission critical, business critical, business operational, and administrative services. The services that fall into each category are normally agreed upon between the business units and the IT department by determining the importance of various business processes and how the processes map to IT systems. Each class, and possibly individual services within a class, has a service level agreement (SLA). The SLA defines different levels of protection against failure, whether caused by hardware or software issues, operator error, data loss or corruption, or other disasters. Additional solutions are required to protect the actual data to makes sure it does not become unavailable because it is deleted or corrupted.

Mission-critical services require technical solutions that include both a service availability and a disaster recovery component as part of a full business continuity plan. The Sun Data Center Reference Architecture provides a best practice for designing many of the components required for business continuity, including local area networks, storage area networks, system management, security, provisioning, and clustering. See “References” on page 31 for the location of Sun Data Center Reference Architecture documents.

About This Document

This BluePrint article is intended for data center architects. It describes some of the technologies available to meet service availability requirements and recommends how the technologies can be used to meet the varying levels of service found in the data center. It also highlights areas for concern when trying to combine both availability and disaster recovery solutions.

The article discusses the options for meeting the SLAs for mission and business critical services with particular reference to the Sun Java™ Availability Suite. Where multiple solutions exist, the underlying complementary technologies such as disk mirroring, data replication, transaction monitors, and database replication techniques are examined to highlight the trade-offs of using certain hardware and software combinations. The article addresses the following topics:

- Service availability and various solutions for achieving high availability
- The advantages and disadvantages of currently available disaster recovery solutions
- Integrating availability and disaster recovery solutions
- The performance impact of separating data centers of long distances

The broader topic of business continuity involves the consideration of more than just system availability and disaster recovery. This article does not cover any aspects of the disaster planning required for telecommunications, staffing, or physical infrastructure, such as buildings, desks, etc.

Overview

When creating a business continuity plan, companies must trade off the cost of the additional infrastructure, e.g., hardware, software, telecommunications, buildings, etc., required against the risks and costs associated with a prolonged outage. As a result, systems that are critical to the business and those for which there is a legal requirement are the top priority.

Defining Availability Objectives

A recovery time objective (RTO) and a recovery point objective (RPO) need to be defined for each component of an application service, taking into account any interdependencies that might affect other applications. The RTO is defined by how quickly applications must be available in order for the business to resume functioning. For example, a shipping company's customers may tolerate a shipping delay of a day or two. However, a financial services organization requires extremely high availability to reduce lost transactions to the smallest extent possible. RPO is the point in time the data must be restored to in order for business to resume, measured by how much data an organization can afford to lose for particular application types. For example, a day's worth of shipping data can probably be reconstructed from an order entry system, but a day's worth of lost medical scans at a hospital can mean the difference between life and death.

When deciding on these objectives it is important to understand the volume of data the application service uses together with the maximal rate of change of that data so that replication technologies can be designed to keep up the growth in data. These figures coupled with the RPO and RTO that represent the business objectives help the IT department determine the feasibility of meeting their recovery goals, as well as sizing and costing appropriate telecommunications capacity. For more details on RPO and RTO see the white paper *Data Replication Strategies* by Jay Orcutt. See "References" on page 31 for the location of this document.

Improving Availability

Service availability is measured by a user's ability to use the service. This could mean entering data or satisfying a query. The service itself might consist of many components—the user interface itself is only the final piece in a long chain of software that generally constitutes a service.

For the service to be available, all the constituent pieces must be available. Rapid recovery of individual service elements is the key issue. Outages can result from hardware failure (component or power failures, etc.), or because of software failures such as operating system panics, application crashes, etc. Availability can also be affected by network connectivity failures.

Most, if not all of these types of failure can be either masked through components redundancy, e.g., by multiple network connections, dual paths to storage, etc., or by having a standby server (that may form part of a cluster) ready to take over the workload.

The key distinction is that the failures are local and usually only affect individual components or systems, while the primary data source is still intact. The fact that the data is usually not lost or compromised means that high availability technology like Sun™ Cluster software can simply try and restart the service. The system on which the service is restarted depends on whether its original hosts is still available. This service is only restarted once the Sun Cluster software determines that the application is stopped successfully on the original host.

The technologies for addressing availability are discussed in “Service Availability” on page 4.

Protecting Against Disasters

A disaster can be considered to be any event resulting in a large number of simultaneous failures, such that the protection offered by a cluster or local availability solution is unable to provide continuous service. The most common causes are fires and floods but also include natural disasters, wars, and other such events.

To protect against disasters businesses need a business continuity plan that includes contingencies for staffing, telecommunications, buildings, etc., as well as the critical system that run the business. Any secondary site that houses alternative computing and data resources needs to be sufficiently far from the primary site to ensure it is not subject to the same event. This might mean that the secondary sites needs to be tens or even hundreds of kilometers away. Once this location is chosen, critical data must be replicated to the secondary site to allow the business to restart critical services should the need arise.

The pivotal IT concern when a disaster strikes is the state of the data and what hardware resources are available to host the service. If the data is inaccessible at the primary site, or becomes corrupted, then restarting the service on the remote site is most likely to restore critical services in a timely fashion. However, this decision requires a human assessment of all factors, many of which cannot be measured or quantified by a computer. Therefore, moving services to a secondary site should be a manual process that involves a joint decision by all of the relevant business units and management staff.

The technologies for addressing disaster recovery are covered in “Disaster Recovery Solutions” on page 10.

Chapter 2
Service Availability

Service availability should not be confused with server availability. Clustered hardware solutions can provide alternative servers on which a service can be hosted in the event of hardware failure, however, this does not guarantee that the service is able to run. The cluster software cannot overcome situations where the application service is mis-configured, is out of storage space, or where the data is corrupted. Therefore, a distinction must be drawn between platform availability and service availability. The latter is of more concern to a business unit and their customers. IT departments can maximize service availability with well trained staff that are implementing carefully tested and documented procedures using effective change control practices.

The broad range of possible SLAs results in a diverse set of possible technical solutions. As these SLAs become more demanding, e.g., 24 x 7x 365 with 99.999 percent availability, the cost in providing the solution becomes correspondingly greater. As a result, business units must be able to justify the cost of the protection afforded to their services.

Service Availability Solutions

The sections below primarily focus on those services that mandate the use of a high availability framework such as that provided by the Sun Cluster software. However, alternative options are considered together with any implicit limitations. The following topics are discussed:

- Single, resilient servers
- Home-grown availability solutions
- Sun Cluster software

A simple definition of availability is the time for which a service is accessible by its users expressed as a percentage of the total time it supposed be accessible. This calculation therefore excludes planned outages for system maintenance and upgrades but includes time where the system is down through some combination of hardware and software failures. It is important to recognize that availability should not be confused with reliability. A highly available system can still be highly unreliable if it has many brief outages. Conversely, a highly reliable system might not be highly available if it has one prolonged outage within the period under consideration. This distinction can often be overlooked or forgotten when designing solutions.

Single, Resilient Servers

Single, resilient servers are best suited to applications with an RTO ranging from hours to days and a low RPO. These applications are generally those that are already duplicated and possible serving static data or data that changes infrequently. This is the lowest cost availability solution, and is easy to implement and administer. However, any extended problems with the single server can translate into long outages. At a minimum, servers should include the following:

- A server with built-in automated recovery features.

- Redundant components where ever possible.
- An operating environment that includes predictive healing technologies. Predictive self-healing is used to facilitate a simplified administration model wherein traditional error messages intended for humans are replaced by binary telemetry events consumed by software components that automatically diagnose the underlying fault or defect. The results of the automated diagnosis are used to initiate self-healing activities such as administrator messaging, isolation or deactivation of faulty components, and guided repair.
- A properly sized and configured backup and recovery environment.

Home-Grown Availability Solutions

Less critical applications such as administrative and business operation services, such as a web server responsible for internal user or non-corporate pages and staff directories, can achieve an adequate level of availability using a combination of the following:

- A server with built-in automated recovery features.
- An operating environment that includes predictive healing technologies.
- An properly sized and configured backup and recovery environment or a remote backup system.
- A alternative server capable of handling a pre-determined amount of the original workload should the primary platform become unserviceable for a prolonged period.
- A set of documented and tested plans and procedures for handling the outages that might occur.

An alternative option is to connect a smaller secondary server to the main storage. This server is configured to use the features of Solaris™ Volume Manager (or VERITAS Volume Manager) to import the metasets (or disk groups), either automatically or manually, from the primary server should it become unavailable. Despite avoiding the cost and complexity of a full clustering environment, the home-grown availability solution approach is not recommended. Companies determined to employ this mechanism should ensure, as a minimum, that:

- Procedures or mechanisms exist to ensure data is not corrupted by any subsequent activity on the primary node.
- Someone who has the skills to carry out the procedures is available both when failures occur and to maintain the scripts as the operating environment and application software are upgraded.
- The hardware combination employed is thoroughly tested. Testing should be carried out with the system under load and failing in various scenarios, e.g., with failed networks, disks, I/O paths, etc. The system must be demonstrated to work successfully without bugs, data corruption, and without contravening any operating specifications, e.g., cable lengths, etc.

Sun Cluster Software

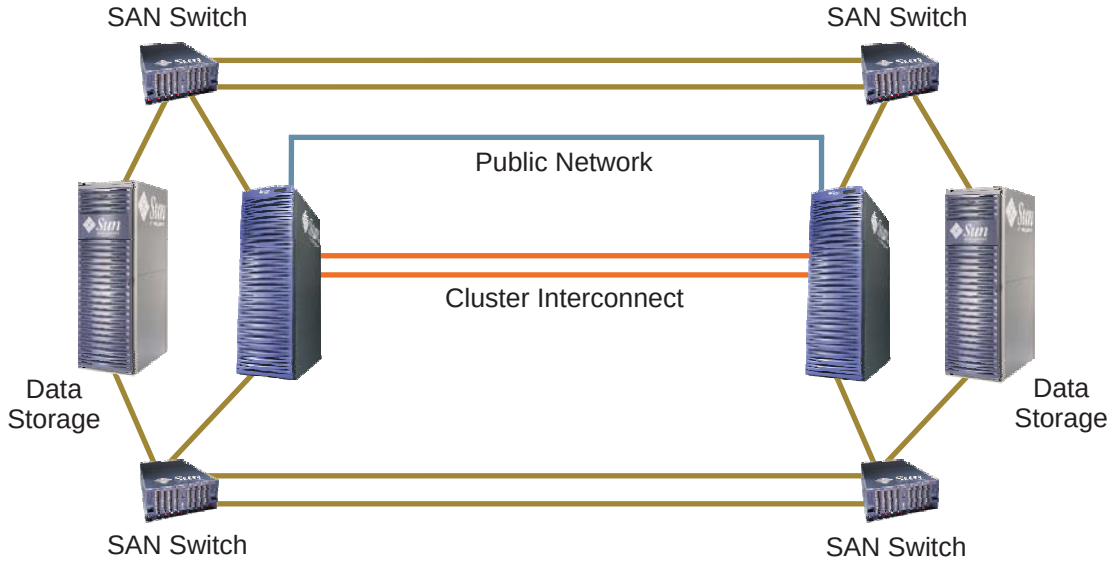
As availability requirements increase, a more robust, automated high availability framework is needed. Services that are business- or mission-critical can benefit from the features delivered by Sun Cluster software, such as:

- A robust resource group framework that starts, stops, and fails over services in an orderly manner. Resources in the Sun Cluster environment can be placed into groups, called resource groups, so that

they can be managed as a unit. A resource group is migrated as a unit if a failover or switchover is initiated on the resource group.
- The ability to detect and react to a range of hardware, software, and service faults.
- The ability to set up service dependencies to help ensure that service components are started and stopped in the correct sequence.
- The capability to bind services to project groups within the Solaris™ Resource Manager software to align resource utilization to business needs.

All of this functionality requires one or more additional servers configured with extra networks to carry the traffic generated by the cluster heartbeat, global file system, and scalable service, as illustrated in Figure 1. As a result, it makes sense to consolidate more than one service onto the cluster to efficiently utilize resources. In this consolidated environment, SLAs can be more successfully met if the services are bound to Solaris Resource Manager projects.

Figure 1. Two Node Cluster with SAN-Attached Data Storage



Furthermore, deploying a vendor availability solution, such as Sun Cluster software, to meet a service's availability objectives includes these added benefits:

- Full support from Sun, when used with certified hardware and operating system combinations
- Fully tested environment, having undergone numerous load and fault injection tests through the Sun Cluster Automated Test Environment (SCATE) software test harness.
- Extensibility through the use of the generic data service (GDS). The GDS is a mechanism for making simple network-aware applications highly available or scalable by plugging them into the Sun Cluster Resource Group Management framework. This mechanism does not require the coding of an agent which is the typical approach for making an application highly available or scalable.

Although the Sun Cluster software automates the responses to a range of failure scenarios, it does not address data availability issues. This is regardless of whether or not the data is protected by some form of RAID technology. If data is deleted or corrupted on a RAID protected device, there is not, generally

speaking[1], any other on-line copy to fall back on. Data must be recovered from tape or point-in-time copies on disk. This can be time consuming and can impact the measured availability of the system.

The importance of SCATE testing cannot be understated. SCATE is a powerful tool developed by Sun to test the Sun Cluster software and associated data services against a range of faults. These tests include panicking nodes, killing processes, performing quorum reservation or disk fencing, etc., in configurable and yet repeatable sequences. This allows new revisions of the Sun Cluster software, application software, and system patches to be extensively tested for correctness of operation and to help ensure against regressions in functionality.

Comparing Availability Solutions

Table 1 below compares the cost, achievable server availability levels, recovery times, complexity, and overall risk of the three potential availability solutions discussed in the previous sections. The indicative availability figures quoted below are generate from field measurements and from component mean time between failure (MTBF) numbers.

Table 1. Comparison of Availability Solutions

Method	Cost	Availability Level	RTO	Management Complexity	Overall Risk
Single, Resilient Server	Low — no additional hardware or software needed	Average — a single server should achieve 99.9 percent availability	Medium — reboots on large servers can take up to 30 minutes. Serious hardware problems can cause even longer outages.	Low — this is a single server and hence the baseline for management complexity.	Medium — problems that cannot be overcome by the server's RAS features can cause extended outages.
Home-Grown Availability Solution	Medium — an additional, dedicated server may be needed	Good — availability levels in excess of 99.9 percent should be achievable	Low — a standby server is available to switch the load to, minimizing outage time to that of problem detection and service restart.	High — any home grown software needs maintenance and testing. Corner case[a] problems can result in a risk of data corruption.	High — corner case problems together with on-going software support issues make this solution a much greater risk than an vendor based one.
Vendor Availability Solution	Medium — a minimum of two or more servers are needed	Good — availability of 99.97 percent or more should be achievable	Low — most problems are automatically detected within a few minutes. Outages times are dominated by service restart times.	Medium — additional training on cluster management is needed for operations staff. Good procedural documentation is important for maintaining availability levels.	Low — due to 7 x 24 support and on-going development by the vendor, the solution is likely to have a greater level of robustness.
Legend	Low/Good	Medium/ Average	HIgh		

a.Extreme cases with combinations of failures or unusual or unexpected conditions, including cascading failures, exhasted resources, etc.

1.Most databases offer recovery mechanisms that allow deleted data to be recovered under certain circumstances without the need to restore from tape. However, certain cases of data corruption may make this difficult, if not impossible.

Coping With Local Disasters

When a clustered solution is installed within a single data center the components are commonly deployed across the entire facility. This assumes that the dimensions of the data center are within the standard Ethernet and Fibre Channel specifications for their respective connections. This approach affords the system some level of protection against localized fires or floods that would otherwise destroy the entire cluster. Such arrangements are still vulnerable to scenarios that can cause half of the servers and the quorum device to fail instantaneously. The reason for this is described in the next section.

Quorum Devices and Three-Site Configurations

Distributed systems, such as those implemented using the Sun Cluster software, rely on heartbeat messages sent over the cluster interconnect networks to determine whether the peer nodes are functioning or not. The aim of this process is to help ensure that a single, safe working set of cluster nodes exist that can provide the configured services. Nodes that join a cluster are required to register a key on a special reserved part of the quorum device. When this safe working set is in doubt, the cluster framework uses a majority mechanism to decide which nodes should continue. If all the heartbeat links are severed or the messages do not get through for any reason, it is impossible for an individual cluster node to determine whether one or more remote nodes are really *dead* or whether they have just lost the ability to communicate. To overcome this impasse a tie-breaking mechanism is used. The node with the lowest cluster node ID in each partition (where a partition is a group of cluster nodes that can still communicate with each other) attempts to remove the registration keys of the nodes not in its partition. Here the race is biased towards potentially larger partitions succeeding by giving them a head start and keys are removed atomically using SCSI-2 operations. The node that performs this operation successfully forms a new cluster with the other nodes in its partition. The remaining nodes panic through lack of quorum.

To work in multi-node clusters, the quorum disk is given a number (N-1) of *votes* according to how many nodes (N) it is connected to. Thus, on a four node cluster, a quorum disk connected to all nodes has three votes. When the cluster, or partition, has a majority of the possible votes, it continues to run the Solaris™ Operating System (OS) and hence the application. When an event happens on a four node cluster, with a four-way connected disk, that requires a *vote* count to be taken, four votes are needed by any partition to continue. This could be one or more nodes plus the quorum device. This quorum mechanism resolves the situations where an even, i.e., 50/50 split has occurred and there is no other way of choosing which half should continue. The quorum device is only important when reconfiguration events happen. If it fails it does not affect the running cluster unless an event happens before it can be replaced.

Even with a working quorum device, it is possible that a disaster could cause half of the cluster nodes to become unavailable, with a 50 percent chance of making the quorum device unavailable too. If this happens the remaining node fails through loss of quorum. Although a quorum device cannot be *mirrored,* multiple quorum devices can be specified. Unfortunately, these configurations actually result in a further decrease in availability rather than an increase and at the cost of increased management complexity. This is due to the increased chance that one or more of the quorum devices could have failed. The only way to help ensure automatic fail-over is to place a quorum device, or an additional cluster node, in a third location that is unlikely to be subject to a disaster that compromises either of the other two locations. There are no other possible technical solutions that can satisfactorily overcome this problem automatically.

Volume Management State Replica Majority

Solaris Volume Manager uses replica databases to store information about the RAID devices that are configured on the system. Data integrity is established by requiring a majority of these replicas to be available when making changes to or importing/exporting a metaset. While this approach is extremely robust in protecting data from corruption, it suffers from similar drawbacks (splits) to those described above.

Configurations that have an even split of storage resources between two location have the Solaris Volume Manager state replicas evenly distributed too. Any event that causes the instantaneous loss of half of the cluster nodes and half of the storage causes metasets currently held on those failed nodes to be imported as *read-only.* This is because their lack of state replica majority means the system cannot risk importing them read/write for fear of data corruption. This outcome is avoided only by configuring storage so that no disaster can result in the instantaneous loss of 50 percent or more of the replicas and half of the cluster nodes. To achieve this, some storage should be installed in a third site that is unlikely to be impacted by any disasters that occur at the other two sites.

An alternative solution is to add an extra state replica database to one LUN in each metaset, where that LUN is located in the same site as the quorum device. Therefore, in scenarios where the site without the quorum is lost, the remaining site can continue automatically and without intervention. Unfortunately, the contrasting scenario that results in the lost of the quorum device results in a little more work to re-establish replica majority.

Chapter 3
Disaster Recovery Solutions

Mission-critical and some business-critical services must continue to meet their SLAs even in the face of a major disaster at the primary data center, e.g., stock market trading where downtime can cost millions of dollars per hour. In order to maintain these services in the event of a disaster, a secondary IT infrastructure must be available at a site sufficiently remote from the primary site so that it is not affected by the same disaster. The time to recover and restart the service is then the sum of the time it takes to decide to enact the disaster recovery plan and switch to remote site operations plus the time it takes for the service to be brought up on the system at the secondary site.

For less critical services where data is not continuously replicated to the secondary site, the time to bring up the service includes the additional time component of restoring data from tape or a point-in-time disk copy, if required.

Regardless of the technique employed, it is important to monitor and address any changes to the service and, in particular, to address any changes in data volume. If, for example, the primary site allocates more storage, but this change is not propagated to the data continuity solution on the secondary site, this mismatch can cause the service to fail to start on the secondary site in a disaster situation.

The following sections begin by discussing the basic infrastructure required by all but the simplest disaster recovery solution. Each of the alternatives are then detailed in turn, examining the trade offs between cost, the RTO, the RPO, and software changes needed to implement the solution. This chapter cover the following topics:

- Backup and recovery from tape
- Host-based replication
- Storage-based replication
- Data replication infrastructure
- Replicating databases
- A comparison of the various disaster recovery solutions
- Metro clusters

Backup And Recovery From Tape

Although RAID technology is designed to provide high levels of data availability through redundancy of the underlying components used to store data, it cannot protect against data loss if the data is deleted (accidental or otherwise) or corrupted. Backing up the data to disk or tape on a regular basis helps ensure that data can be recovered to an earlier point in time.

Once backup of the data is finished, the tapes can be cloned, i.e., copied to new media to allow them to be stored off-site in a disaster recovery location. This approach is the least expensive of all of the options discussed in this chapter. Alternatively, if a second tape library is accessible over the network in the disaster recovery site, the tapes can be cloned directly to the remote location. This mechanism can make

large network—local area network (LAN) or storage are network (SAN)—demands for the duration of the cloning process and its impact on other services sharing that infrastructure should not be ignored or underestimated.

Although backup and recovery from tape is a robust approach to creating usable, point-in-time copy of the data, it is only really applicable as the primary disaster recovery mechanism for non-critical services, i.e. services with RPOs where data loss and longer RTOs are acceptable. Recovery from tape can also serve as a method of last resort for critical services. Two examples illustrate the issues with this method of disaster recovery.

In the first scenario, tape backups are started at midnight and are complete by 2:00 am. The media is then cloned and the copy is physically transported to a remote site and delivered at 9:00 am. In the second, scenario, the media is cloned directly to the remote site, completing by 4:00 am. However, in both cases, if disaster strikes at 4:00 pm that day, an entire day's transactions are lost. It also takes roughly three times as long to restore the data as it does to back it up, so services in this example are not available again until approximately 11:00 pm. This is based on two hours to back up, six hours to restore and one hour to recognize and react to the disaster, starting from 4:00 pm.

The amount of data loss can be reduced by creating incremental backups during the day. However, there is always a finite level of data loss and a relatively prolonged recovery time associated with tape backup and recovery.

Using Snapshots to Lower the Impact of Tape Backups

SAN-based tape backups can stream data to multiple devices in parallel and, as a result, have the potential to make a sizeable impact on service performance. This is because data traffic to tape is competing for Fibre Channel I/O bandwidth with the production I/O requests. This can be avoided by using snapshot technology such as: the host-based Sun StorEdge™ Availability Suite software, the storage-based Sun StorEdge™ Data Snapshot software for the Sun StorEdge™ 6000 family or the storage-based Sun StorEdge™ 9900 ShadowImage software for the Sun StorEdge™ 9900 series. Snapshot technology enables a point-in-time copy of the data to be created. This *snapshot* can then be backed up instead of the production data.

Backup and Recovery Performance

Backup and recovery performance is determined to a large extend by a combination of four factors: raw tape throughput, file system type, the data stored in the file system, and the method for creating the backup. The fastest backup and recovery mechanism allows both the disk storage and tape to stream data in an uninterrupted fashion. This only occurs when creating or restoring a physical copy, i.e., raw disk blocks, of the underlying data or when moving very large files. Where there is a requirement to create a logical copy of the data, i.e., through a file system, the process is inhibited by the need to look-up or restore file system metadata. This is accentuated if the file system contains a large amount of small files.

Remote Disk Mirroring

Solaris Volume Manager is often used to provide local RAID-protected storage for servers. These logical devices are usually constructed using a combination of striping (RAID 0) and mirroring (RAID 1) technologies. A stripe of mirrors is most commonly used because of its enhanced recovery characteristics after a physical disk has failed.

While most mirrors have only two underlying, local, logical components, it is also possible to maintain a third remote mirrored component. The CPU cycles required to perform double or even triple mirrors are relatively small, typically less than 5 percent. However, the performance penalties to the application are entirely dependent on the I/O profile and the latency of the link between the two sites.

Remote disk mirroring solutions incur some additional capital and recurring costs. The operational costs include charges for the rental of the dark fibre and any notional costs associated with power, cooling, and space usage in the second data centre while capital expenditure may be needed for extra Dense Wave Division Multiplexors (DWDM) or SAN switches.

A remote, third mirror need not be physically connected to, or associated with, a physical server. A disaster recovery plan need only specify the procedure for making such an association should the need arise. Indeed, no actual server need ever be present in the second location, if an appropriate server can be procured or rented at short notice from a supplier. In each case, the time to recover is entirely dependent on the time it takes to associate the remaining mirror with a server running a similar operating environment and patch level as the server at the primary site. If the server is already on-site, this could be as little as a few minutes. Only the Solaris 9 update 7 (and above) Operating Systems allow the use Solaris Volume Manager metasets to create shared disk sets outside of the Sun Cluster software. These can be imported into the new system using the `metaimport` command. Systems using earlier versions of the Solaris OS need to re-create the metaset on the new target server. When this is performed correctly, using the same Solaris Volume Manager commands used to create the original set, the data is not destroyed[1].

Third mirrors have the advantage that the data they contain is up to date and do not require any changes to applications in order to function. Data that is committed to the third mirror is protected in the event of a disaster at the primary site, providing an effective solution for services with RPOs that require zero data loss. However, the disadvantage of this solution is that deleted or corrupted data is immediately propagated to all of the components of the mirror. This issue can be essentially rectified by creating frequent snapshots of the data and using these snapshots to restore corrupted or deleted data.

Companies such as SunGard can provide out-sourcing services to accommodate third-site mirrors for organizations that do not want to implement this type of solution in-house.

1.The critical success factor is that the `metaset`(1m) command to add the LUNs to the set on the new host should not change the LUN partitioning. Therefore, the partition table required by Solaris Volume Manager must be compatible between operating system release and patch levels of the two hosts. For more information, see the `metaset`(1m) manual pages and the associated commands.

Host-Based Replication

Sun StorEdge Availability Suite software consists of two complementary components: point-in-time copy and remote mirror software. The suite can be used to create a disaster recovery environment that offers some additional protection against deleted or corrupted data.

The remote mirror software works at the Solaris OS kernel level to intercept writes to underlying logical (Solaris Volume Manager and VERITAS Volume Manager, but not their cluster volume management counterparts) devices as well as to physical devices, such as disk slices and hardware RAID protected LUNs. It then forwards these writes on to one or more remote Solaris OS-based nodes connected through an IP-based network. This data transfer can be configured in one of two ways and is illustrated in Figure 2.

- *Synchronous mode replication*—causes the initial write to wait until the local bitmap is updated and the remote write is committed to a remote storage device by the receiving Sun StorEdge Availability Suite kernel module. This mode injects latency into the application's write performance. When this additional latency becomes significant compared with the standard, non-replicated write latency, the application suffers some form of performance degradation or lowering of throughput.
- *Asynchronous mode replication*—allows the initial kernel write to return immediately without waiting for an acknowledgement. In this mode, only the additional latency of writing to the bitmap is added to the application's write performance. However, this means that when a disaster strikes, some data might not be replicated to the remote site(s). If the data forms part of a database log, one or more transactions that are considered complete on the primary site can be incomplete on the secondary site. Therefore, asynchronous replication should only be used where data loss is tolerable or in conjunction with synchronous replication for critical components, such as database online redo logs, control files, and rollback segments.

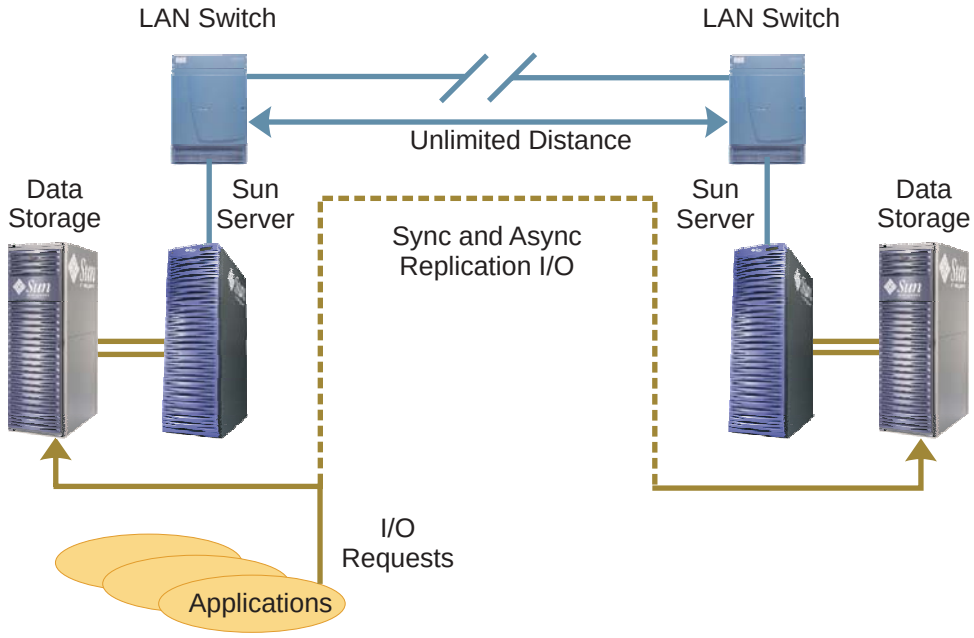
Both modes add extra latency to write operations because the bitmap indicating changes for the replicated volume must be updated before the actual data writes to local disk or transmits across the network. To minimize the impact of this latency, the bitmap volumes should be placed on separate LUNs to avoid contention.

The remote mirror software is designed to tolerate network failures. If inter-site connectivity is lost, the primary site goes into *logging* mode. Changes to the data are marked in the bitmap so that the secondary (mirrored) volumes can be resynchronised quickly. The bitmap is cleared once the connectivity is restored. In logging mode, write ordering is no longer preserved.

The logging mechanism can be initiated manually as well. In addition, the direction of synchronization can be reversed to allow a primary site to be recovered from the disaster recovery site. This allows disaster recovery plans to be tested on a regular basis.

In order to ensure maximal remote mirroring performance, host TCP parameters including: tcp_max_buf, tcp_cwnd_max, tcp_xmit_hiwat, tcp_recv_hiwat should be tuned. For more details see *Sun StorEdge™ Availability Suite 3.2 Remote Mirror Software Configuration Guide*. See “References” on page 31 for the location of this document.

Figure 2. Data Replication Using Sun StorEdge Availability Suite Remote Mirroring



Restrictions on Using Sun StorEdge Availability Suite Software With Parallel Databases

The way the remoter mirror software works precludes its use for replicating data held in parallel databases such as Oracle® Parallel Server (OPS) or Oracle Real Application Cluster (RAC). This is because data is written to the volume managers or global devices concurrently from multiple nodes, making it impossible for any single kernel to intercept and forward all of the writes performed.

Using Point-in-Time Software for Fast Recovery

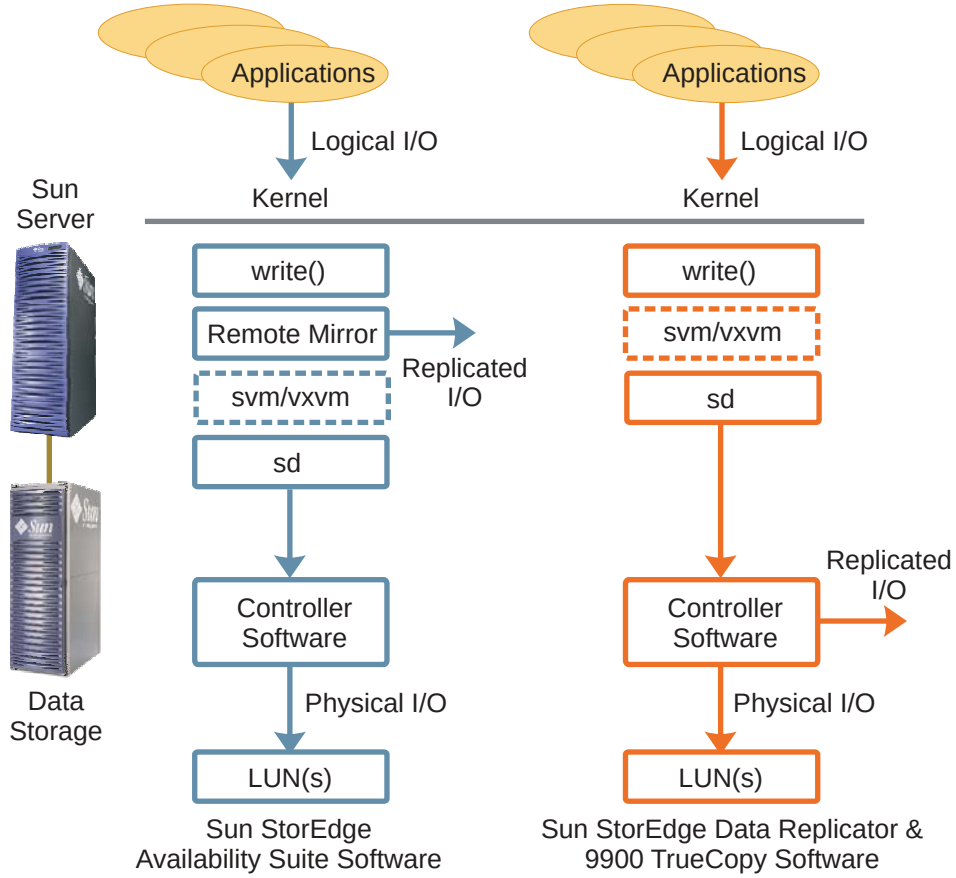
Synchronous replication using the remote mirror software has the same pitfalls as remote disk mirroring. Any deleted or corrupted data is immediately propagated to the remote site. A robust recovery strategy uses regular disk-based point-in-time copies, such as the point-in-time software within Sun StorEdge Availability Suite software, to minimize the amount of data loss if one or more files are deleted. This comes at the cost of increasing the amount of disk storage needed to house the snapshots, although the expense can be somewhat minimized by using lower performance, higher capacity drives such as SATA disks.

Storage-Based Data Replication

The Sun StorEdge 6000 family and 9900 series perform data replication on the CPUs or controllers resident in the storage systems. The functionality of both the Sun StorEdge Data Replicator and the Sun StorEdge 9900 TrueCopy software is broadly similar to that of Sun StorEdge Availability Suite software described above, i.e., synchronous and asynchronous modes, but the software operates at a much lower level, as depicted in Figure 3. Consequently, storage-based replication software can replicate data held by applications such as Oracle OPS and Oracle RAC even though the I/Os to a single LUN may be issued by several nodes concurrently. The software features are also independent of the operating environments of

the connected servers, enabling a single replication technique to be used across heterogeneous servers and thus lowering management and administrative overhead.

Figure 3. Comparing How I/O is Replicated



The Sun StorEdge 9900 ShadowImage software provides snapshot capabilities similar to that of Sun StorEdge Availability Suite software, and can also be combined with Sun StorEdge 9900 TrueCopy software to rapidly restore data if it is deleted or corrupted.

The Sun StorEdge 9900 family has the capability to replicate data over Fibre Channel or an ESCON serial link. In contrast, the Sun StorEdge 6920 system can replicate data either over Fibre Channel or through an IP network. Many other enterprise-class storage sub-systems provide storage-based replication facilities, e.g., EMC Symmetrix arrays combined with Symmetrix Remote Data Facility (SRDF).

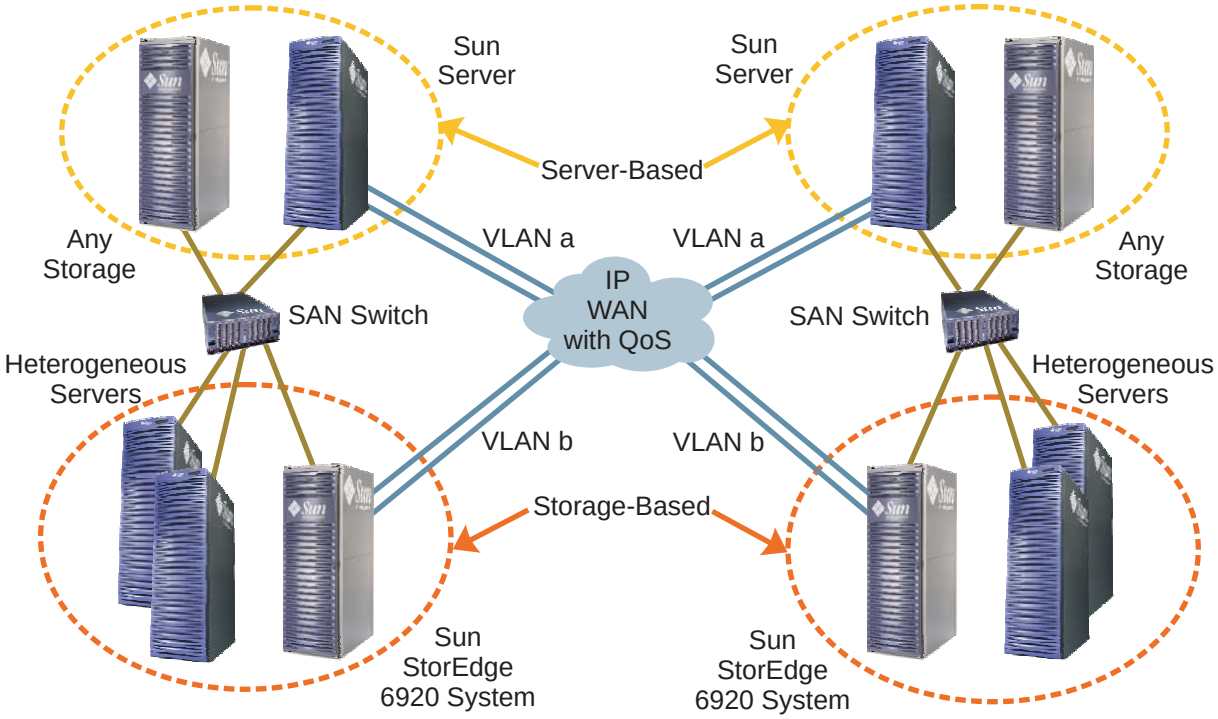
Data Replication Infrastructure

There are a wide range of techniques available for creating copies of application data at one or more remote sites. The appropriate mechanism for a particular application service depends on a combination of the service's RPO and RTO, weighed against the cost of meeting these objectives. Most of these mechanisms require some level of permanent communications infrastructure between the sites involved.

Disaster recovery solutions consist of a primary data center that is backed up by one or more secondary sites. These secondary sites are located a sufficient distance away so as to be unaffected by disasters that might strike the primary. The distance can vary from a few kilometers to several thousand kilometers depending on business needs, budgetary constraints, regulatory requirements, and availability of suitable facilities. However, most have a basic IP network infrastructure connecting the facilities and some may also have a Fibre Channel SAN. It is very important that the IP network and SAN are properly implemented and configured for any solution that supports data replication.

The first objective is to create an infrastructure that is resilient and highly available (it should also be secure, but that discussion is beyond the scope of this document). This requires the core LAN or SAN to have redundant components on each site that are connected through links following independent paths, so as not to present a single point of failure, as shown in Figure 4. Once this requirement is met, the switches must be configured to guarantee a quality of service (QoS) level for IP or Fibre Channel traffic.

Figure 4. Server- and Storage-Based Replication Through a Common IP Network

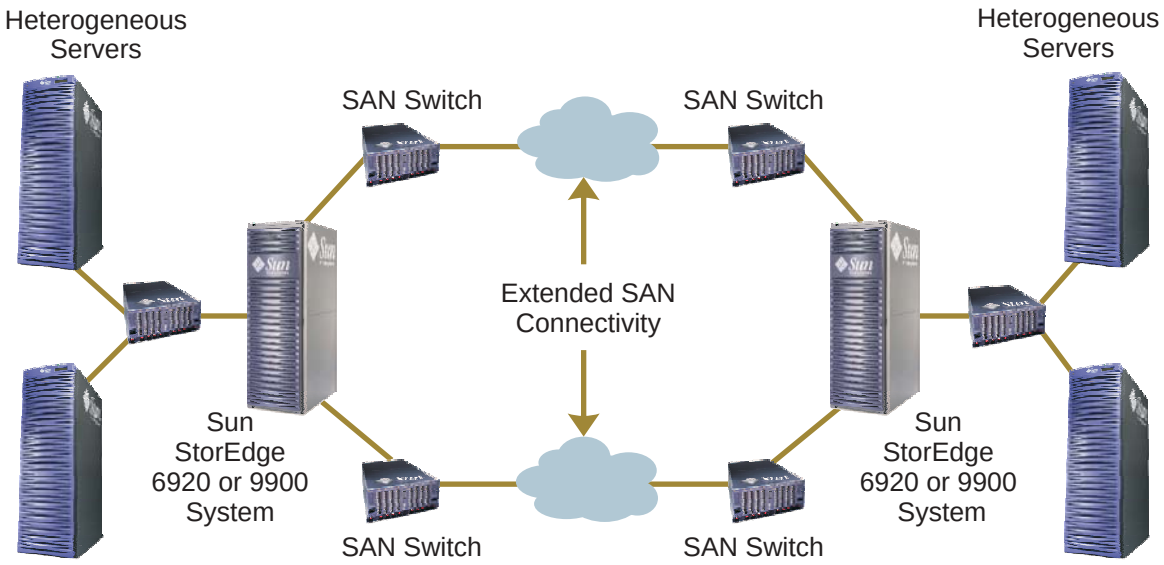


In a LAN environment, traffic is typically separated using virtual LANS (VLANs). As a result, one or more dedicated VLANs should be configured to carry IP data replication traffic. Each VLAN should be prioritized with a QoS guarantee.

For host-based data replication, resilient network connections can be configured with a Solaris IP Multi-Path (IPMP) group that connects to separate LAN switches in each data center. For storage-based data replication, such as that provided by the Sun StorEdge 6920 system, the storage system is connected to the LAN through two connections from separate storage controllers (SIO) cards.

When the storage-based replication uses Fibre Channel connectivity, rather than an IP network, the outbound Fibre Channel connection should connect to an independent SAN switch to help ensure resilience and be configured into zone for security, as illustrated in Figure 5.

Figure 5. Storage-Based Replication Through an Extended SAN Infrastructure



If the SAN is extended more than 10 kilometers, sufficient buffer credits must be allocated to the links carrying the replication traffic. The Sun Data Centre Reference Architecture Implementation SAN design incorporates these requirements. See “References” on page 31 for the location of this document.

Transaction Monitors and Application Servers

Application servers (e.g., Sun Java System Application Server software) and transaction monitors (e.g., BEA Tuxedo) include messaging and transaction processing APIs, among their many other features, that allow developers to write services to perform highly customized, distributed transactions, honed to match their specific business needs. The transactional capabilities include full two-phase commit protocols for distributed transactions. By writing such applications, organizations can build in their RPO and RTO trade-offs.

For example, an application server service might allow transactions to replicate across widely separated sites to provide both higher availability and disaster recovery. The service could be coded to handle network outages to the primary or secondary systems. For example, the service could determine whether transactions should be queued if they cannot be completed on the primary or whether the service should be suspended until the primary becomes available again. Alternatively, the transactions might be queued to the primary site while they continue to complete against the secondary (or vice versa).

Using an application server or transaction monitor requires considerable investment if applications are not created to use these APIs. This, coupled with the cost of implementing and testing the changes often restricts this approach to all but the most critical, legacy applications. Further development, support, and

maintenance of this code is then an on-going responsibility. However, new application developments that are architected to uses these APIs can benefit from these facilities.

Replicating Databases

Relational database management software (RDBMS) portfolios from IBM, Oracle, and Sybase include a wide range of tools to manage and administer data held in their respective databases: DB2, Oracle, and Sybase. The data replication tools include IBM DB2 DataPropagator, Oracle Data Guard, and Sybase Replication Server. These tools differ from products like Sun StorEdge Availability Suites software and Sun StorEdge 9900 TrueCopy software because the database tools work at a transactional level rather than at the disk block level. The RDBMS software is designed to handle logical changes to the underlying data, e.g., insert a new row or decrement all prices by 5 percent, and has intimate knowledge of which low level writes constitute a complete, atomic change. In contrast, a block-level replicator knows only about a stream of I/O requests, without recognizing where the transactions begin and end. As a result, database replication offers considerably greater flexibility and lower network traffic than a corresponding block-based replication solution.

For example, Oracle Data Guard software uses the concept of primary and standby databases. Either or both of primary and standby databases can be configured as RAC clustered databases, as required. The production workload runs against the primary database, with one or more standby databases configured for disaster recovery purposes. The standby databases can be used simultaneously for reporting purposes.

Oracle Data Guard software includes three replication configuration options:

- *Maximum protection mode.* In maximum protection mode no data is lost because the redo information needed to recover each transaction is written to the local log and one or more remote redo logs before the transaction is considered complete, or committed. Any fault prevents this write from occurring, causing the primary database to shut down.
- *Maximum availability mode.* If the consequences of maximum protection mode are too restrictive, then maximum availability mode offers an alternative option to deal with failures. Here, the primary database continues, but changes are stored until they can be passed on to the standby database. Once connectivity is re-established, the software reverts to continuous replication. In maximum availability mode, data is only lost when a prior failure prevents the resynchronization of the primary online redo log with the secondary standby redo log(s).
- *Maximum performance mode*. In this mode, transactions are committed through the local primary redo log, which are then asynchronously replicated to the secondary standby redo logs. When sufficient network bandwidth exists, this option approximates to maximum availability. However, data is lost if a disaster occurs in the primary location before all of the outstanding transaction data is replicated.

The performance of Oracle Data Guard, together with that of the other RDBMS replication packages, is governed by the bandwidth and latency of the links between the primary and secondary sites and is discussed in “The Performance Impact of Separating Data Centers” on page 27.

Because RDBMSs work on transactional changes to the underlying data, it is much simpler to revert to an earlier consistent state. As with other continuous replication technologies, the database replication software also succumbs to the possibility of immediately propagating accidental data deletion or corruption. However, software options such as Oracle's Flashback Technology enables these changes to be reversed relatively quickly without requiring major bulk restores of data.

Comparing Disaster Recovery Solutions

Table 2 compares the benefits and drawbacks of the mechanisms described in this chapter. A high RPO indicates the service can survive with data that may be out of date, while a low RPO indicates the data must be as current as possible, if not completely up to date.

Table 2. Comparison of Disaster Recovery Solutions

Method	Cost	Recovery Point Objective (RPO)	Recovery Time Objective (RTO)	Performance Impact
Tape Backup and Restore	Low	High — but dependent on when the last usable backup was created	High — typically three times the time it takes to back up the data	Low — though the backup process might have some impact depending on the precise mechanism
Remote Mirror	Medium	Medium — due to vulnerability to deletion or corruption	Low — so long as a suitable host and software environment are available to restart the service on the remote site	Medium — but dependent on the distance to the remote mirror
Synchronous Host-Based Data Replication	Low — because additional storage space is required to provide complete protection against data deletion or corruption, and low cost storage can be used	Low — but only if coupled with snapshot techniques to guard against loss through corruption or deletion	Low	Medium — so long as sufficient inter-site network bandwidth is available and its latency is low enough to meet peak application I/O demands
Asynchronous Host-Based Data Replication	Low — because additional storage space is required to provide complete protection against data deletion or corruption, and low cost storage can be used	Medium — but only if coupled with snapshot techniques to guard against loss through corruption or deletion	Medium — so long as a suitable host and software environment are available to restart the service on the remote site	Medium — so long as sufficient inter-site network bandwidth is available to meet peak application I/O demands
Synchronous Storage-Based Replication	Medium — because additional storage space is required to provide complete protection against data deletion or corruption	Low — but only if coupled with snapshot techniques to guard against loss through corruption or deletion	Low — so long as a suitable host and software environment are available to restart the service on the remote site	Medium — so long as sufficient inter-site network bandwidth is available and its latency is low enough to meet peak application I/O demands

Method	Cost	Recovery Point Objective (RPO)	Recovery Time Objective (RTO)	Performance Impact
Asynchronous Storage-Based Replication	Medium — because additional storage space is required to provide complete protection against data deletion or corruption	Medium — but only if coupled with snapshot techniques to guard against loss through corruption or deletion	Medium — so long as a suitable host and software environment are available to restart the service on the remote site	Medium — so long as sufficient inter-site network bandwidth is available to meet peak application I/O demands
Transaction Monitors/ Application Servers	High — because application software might need to be re-written to use these technologies	Low — if two-phase commit type logic is used, no transactions should be lost	Low	Low — because only transactional information has to be passed between sites
Database Replication (but only addresses database data)	Medium — dependent on ISV licensing costs	Configurable	Variable	Low
Legend	Low	Medium	High	Variable

Metro Clusters

Recent improvements in optical technology are decreasing the cost of long distance, high bandwidth connectivity. Consequentially, the ability to cluster systems across hundreds of kilometers using Dense Wave Division Multiplexors (DWDM) and SAN connected Fibre Channel storage devices is technically and financially feasible. As a result, more organizations are proposing cluster deployments that try to combine availability and disaster recovery by separating the two halves of the cluster and storage between two widely separated data centers. The physically separated cluster nodes work identically but have the added benefits of protecting against local disasters and eliminating the requirement for a dedicated disaster recovery environment.

Despite the wide separation of the nodes, this approach still effectively creates a *local* cluster, with all the issues related to quorums. Performance of the applications also suffers as a result of the increased distance between the cluster nodes. This reasons for the performance impact are described in detail in “The Performance Impact of Separating Data Centers” on page 27.

Chapter 4

Integrating Availability and Disaster Recovery Using Sun Cluster Geographic Edition Software

Deploying a disaster recovery solution does not preclude the use of availability solutions at the individual data centers. The Sun Cluster Geographic Edition software provides a framework that allows one or more data services to be moved between a primary Sun Cluster environment and secondary environment in a controlled fashion. This chapter is an introduction to the key features of the Sun Cluster Geographic Editions software:

- Partnerships
- Data replication
- Protection groups

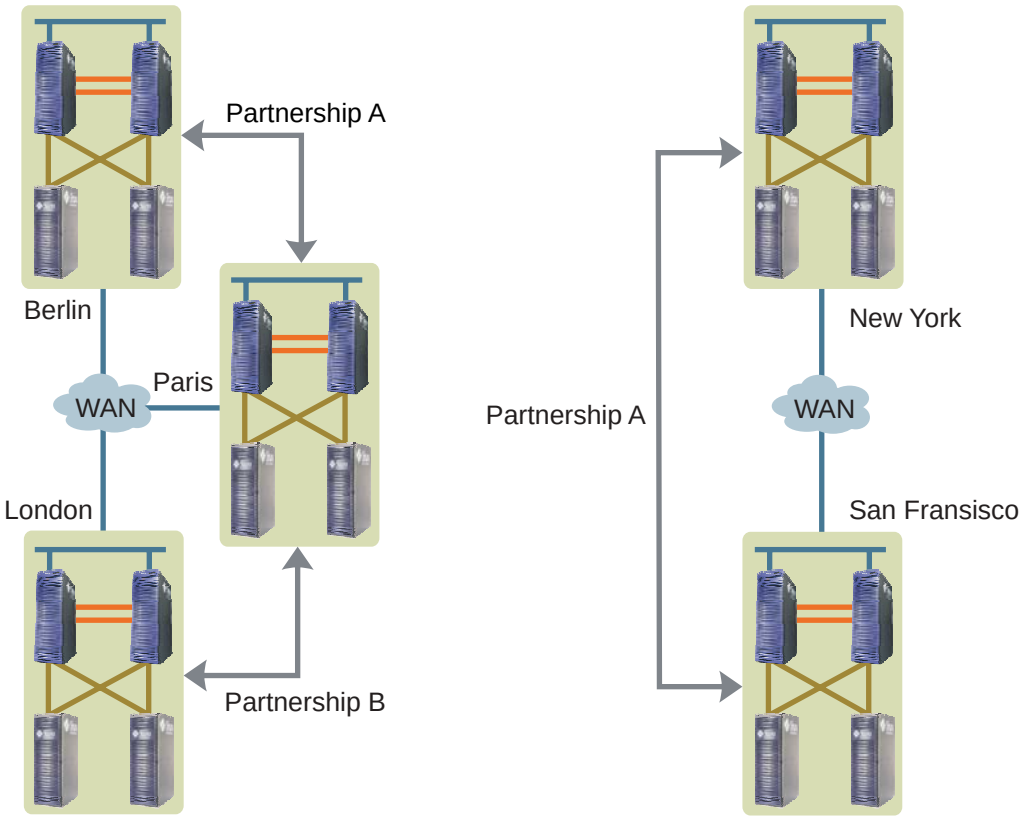
Sun Cluster Geographic Edition software is a layered extension to the Sun Cluster software that combines the data replication capabilities of Sun StorEdge Availability Suite software or Sun StorEdge 9900 TrueCopy software to provide an integrated disaster recovery solution. It enables controlled migration of production services from a primary cluster to one or more secondary clusters either in the event of a disaster or as part of a planned procedure. Data is continuously replicated from the primary cluster to the secondary cluster either synchronously or asynchronously (or a combination of both) depending on the RPOs of the application services supported by the clusters.

Partnerships

The Sun Cluster Geographic Edition software enables partnerships to be formed between clusters to provide mutual protection against disasters. The clusters in a partnership monitor each other by sending heartbeat messages to each other in a similar fashion to the nodes within a single cluster. Unlike local clusters, the clusters in a partnership use the public network for these messages, but support additional, plug-in mechanisms as well.

Two partnership topologies are supported: N+1 and pair, as illustrated in Figure 6. The overriding rule is that a cluster cannot form more than one partnership with the same cluster. The term cluster refers to both single- and multi-node Sun Cluster implementations. Using Sun Cluster software even in a single node implementation means that services benefit from the strong start/stop control that the resource group manager provides.

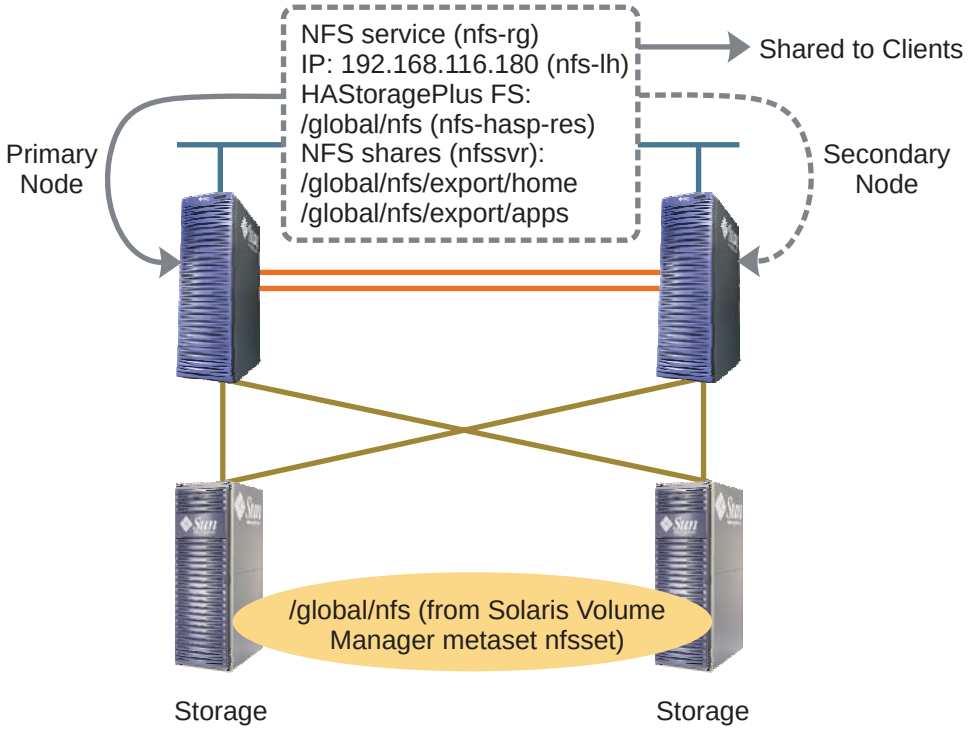
Figure 6. Sun Cluster Geographic Edition Topologies: N+1 or Paired



Replicating Cluster Data to Protect a Service Against Disasters

The Sun Cluster software uses the concept of resource groups to describe a logical collection of IP addresses, file systems, and application software that comprise some or all of an application service. The simple NFS example below consists of a resource group containing one logical IP address, one failover file system, and an NFS service sharing two directories in that file system. The IP address is described as a *logical* address because it moves between the cluster nodes as the service is mastered on one, then the other, or others if this is a multi-node cluster. Unlike logical addresses, the physical addresses are assigned permanently to each individual cluster node.

Figure 7. Simple Cluster with One NFS Service Resource Group (With Resource Names)

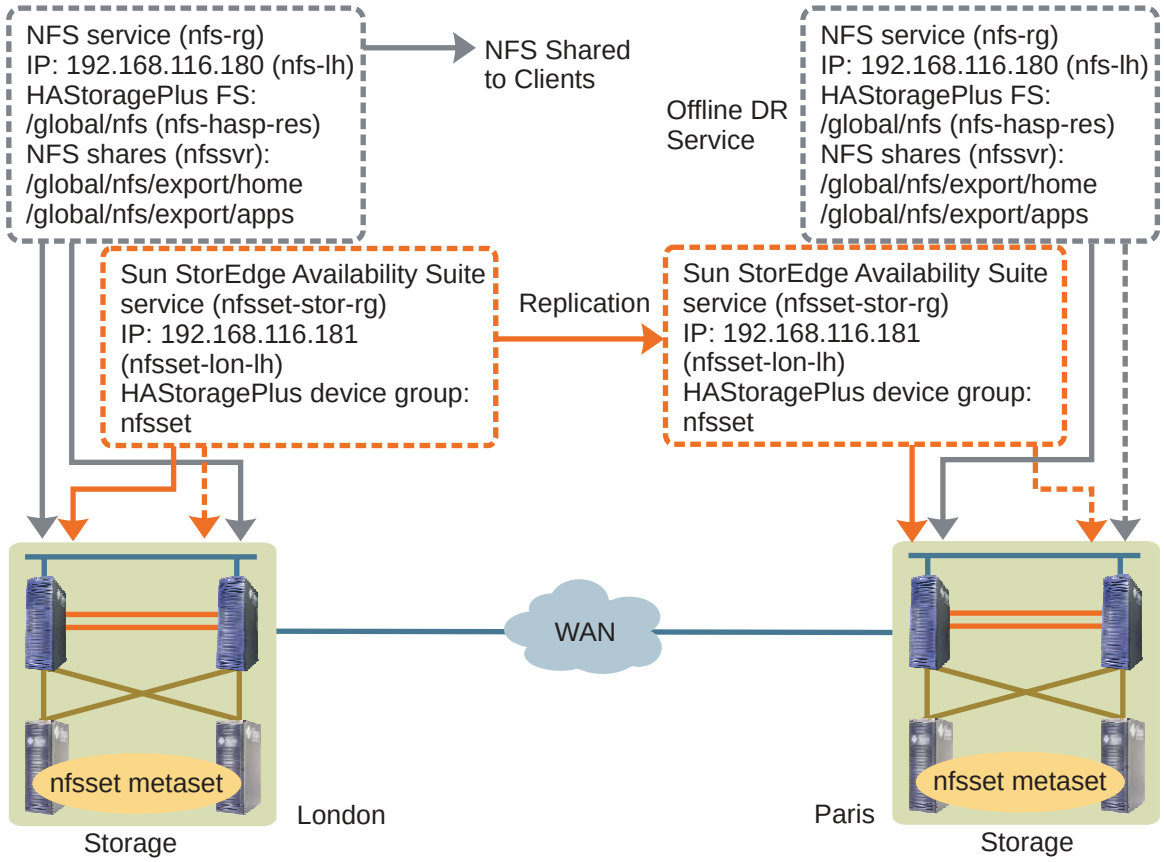


In order to protect the example NFS service above[1], data replication must be configured. If Sun StorEdge Availability Suite software is employed, an additional lightweight resource group must be created to collocate the IP address used for the host-based replication and the data being replicated. The Sun Cluster software's global file system has the concept of a primary I/O path. This is the path through which all I/O passes to disk. A global file system is mounted on all cluster nodes, enabling services to write to it from non-primary nodes. An NFS service sharing for example, `/global/nfs/data` could be hosted on node A while the primary I/O path for this could be through node B. If `/global/nfs/data` is replicated, the data passes from the node of the primary I/O path, i.e., node B. Thus, the replication service for a data set needs a logical IP address in a resource group because it is required to move between nodes and the node on which it is on at any point in time might not be the same as the logical host of the service. The additional light-weight resource group also means that the NFS resource group can be taken offline without necessarily stopping the replication.

Each device group, e.g., Solaris Volume Manager metaset or VERITAS Volume Manager device group, that replicates data using Sun StorEdge Availability Suite software requires an individual lightweight storage resource group. Sun Cluster software uses these resource groups to start and stop the replication process if services, and hence their data, migrate between clusters. This in turn requires logical host names (nfsset-lon-lh and nfsset-par-lh in Figure 8) that follow the source of the data that is replicated. In contrast, Sun StorEdge 9900 TrueCopy software does not need these additional resource groups because the replication occurs in the storage unit rather than at the host level.

1.Sun StorEdge Availability Suite software can be used to replicate local, highly available local, and global file systems.

Figure 8. Cluster with an NFS Service Replicating Data to Another Cluster

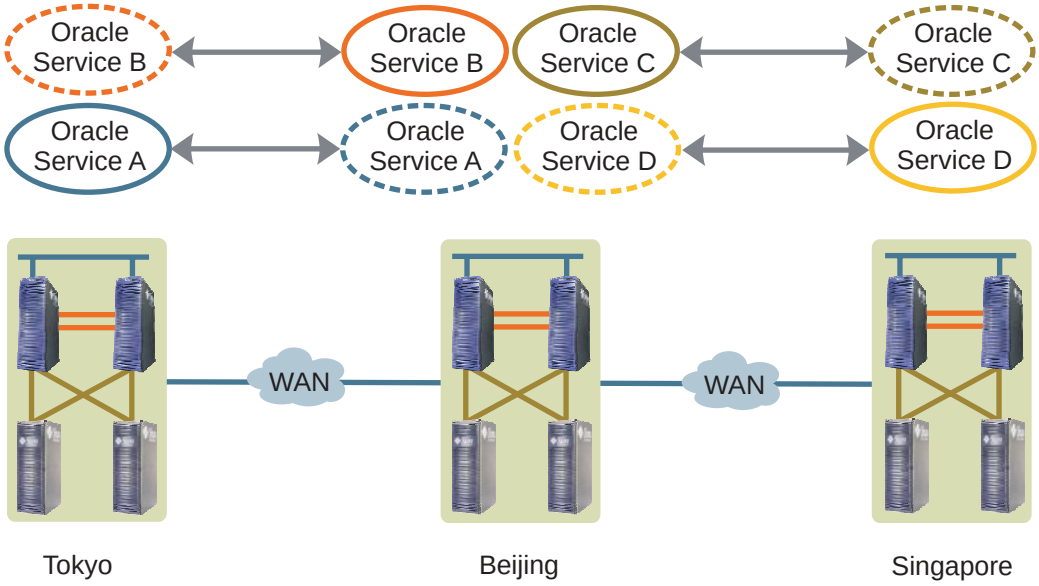


A configuration using Sun Cluster software and data replication software alone allows two clusters to provide a disaster recovery service. However, all of the monitoring and the processes to migrate services safely between clusters needs to be implemented separately.

Using Sun Cluster Geographic Edition Protection Groups

Sun Cluster Geographic Edition software provides a framework for monitoring and managing the controlled migration of services between the primary and a secondary site. Cluster resource groups, together with the data they rely on, are described in terms of protection groups. Each protection group is assigned to a primary cluster, as well as a secondary cluster that is used in the event of a disaster. Therefore, any cluster in a partnership can act as a primary host to some protection groups and a secondary to others, as depicted in Figure 9 with Oracle services. The benefit is that hardware resources can be better utilized, rather than remaining completely idle.

Figure 9. Four Oracle Protection Groups Split Across Three Active Clusters



Under normal conditions, a protection group can be migrated between its host cluster through an administrative switch-over. This process uses the Sun Cluster resource *stop* method to shut down the application in a controlled fashion. After the application is stopped, the associated file systems can be unmounted, the resource group is set to an offline status, and the data replication is stopped. On the second cluster, the process is reversed—the direction of the data replication is reversed, the resource group is started and it mounts the necessary file systems before the application service is started through the Sun Cluster resource *start* method. This allows disaster recovery procedures to tested and practiced to help ensure the procedures are realistic for actual disasters.

If the heartbeat messages fail, the Sun Cluster Geographic Edition software notifies the system administrator of the potential problem either through E-mail, the Sun Cluster Geographic Edition graphical user interface (GUI), or through a pager. Unlike its local Sun Cluster counterpart, Sun Cluster Geographic Edition software does not attempt to automatically fail over the service to the secondary site. This is because most IT departments have significant processes to follow before deciding whether to restart the service on a secondary site.

If a real disaster occurs and the decision is made to migrate the service to the secondary site, it impossible to effect a controlled shutdown of the service on the primary cluster. In these circumstances a forced take-over is used to restart the service on the secondary site. This process should not be considered lightly. There is a significant risk of data corruption if the service is still active on the primary site. The forced take-over overrides the controls that normally protect data integrity. This is another reason why the take-over process is a manual one rather than an automatic one.

Chapter 5
The Performance Impact of Separating Data Centers

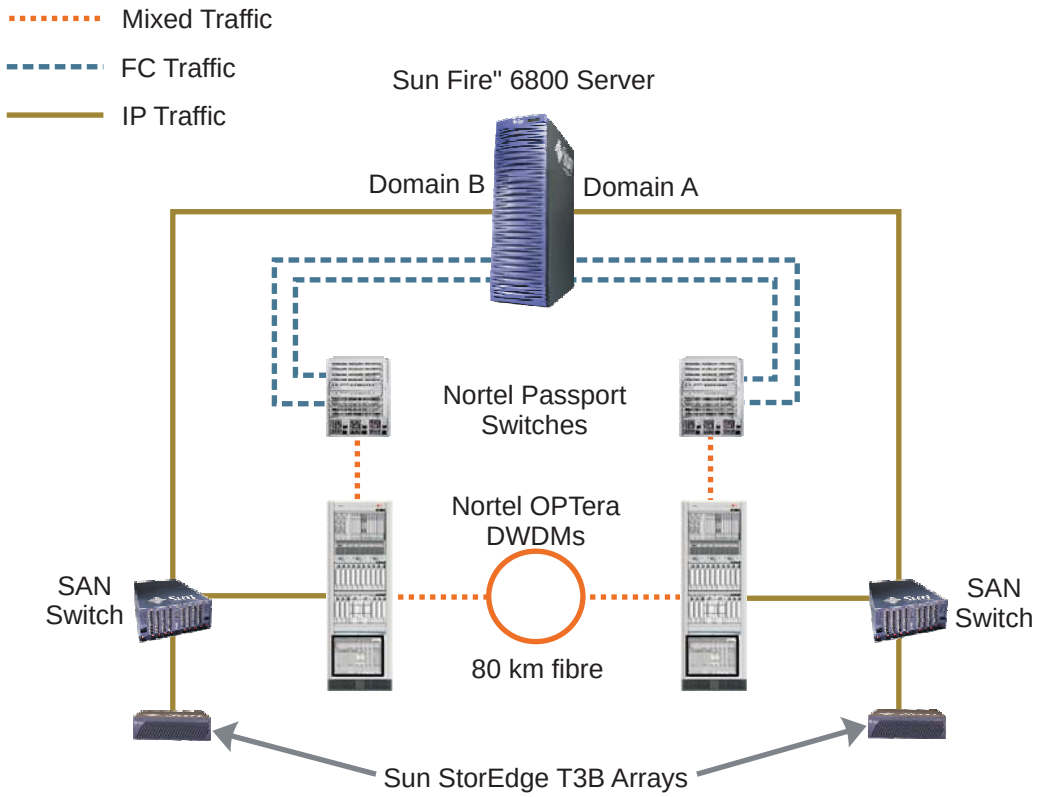
Multiple, geographically separated data centers are often used to provide disaster recovery capabilities. Mission-critical data is replicated either synchronously or asynchronously between sites depending on the sensitivity of the business to data loss (in the event of a disaster). But this insurance comes at a price. Each additional meter of separation injects an extra 5 nanoseconds of I/O latency. Every physical storage I/O requires two SCSI transactions that must be completed serially and acknowledged before the I/O is considered complete—one transaction to set up the command and one to send the data. Although some SAN switches allow the SCSI set up command to be acknowledged locally, the data transfer and acknowledgement must physically occur before the I/O returns as complete in the operating system.

Bandwidth bottlenecks can be removed to a great extent. Unfortunately latency bottlenecks can only be overcome to a point, e.g., the speed of light in glass is 5 nanoseconds per meter. For example, attempting to replicate transactions synchronously over 1000 kilometers inevitably adds a delay of a minimum of 2/100th of a second (1000 [kilometers] x 1000 [meters per kilometer] x 2 [out and back] x 2 [trips] x 5 [nanoseconds per meter]) to the transaction. Additional factors, such as other traffic on the communications link, forwarding delays through the communications equipment involved, and the possibility that the communications links follow an indirect route, could increase the latency to 50 to 100 milliseconds. The data also needs to physically write to a disk at the standby site, possibly adding a further 5 to 10 milliseconds. Given these facts, a basic delay of approximately 0.1 seconds might be more realistic. However, the absolute number is not that important. Instead, it is the relative increase over the latency of the application before it is replicated that determines the perceived performance penalty.

For example, a database service performs well on a system with locally attached storage, but which is known to be sensitive to the response time of single-threaded, sequential writes to its online redo logs. If the typical response time for this locally attached LUN is 2 milliseconds, then configuring a mirror 100 kilometers away adds a minimum of 2 milliseconds to the service time, or a 100 percent increase. This can result in a substantial, and unavoidable, drop in database performance. Here, the underlying technology makes no difference. Both mirroring and replication are subject to the same basic law of physics.

These characteristics are demonstrated by tests run at Sun's Global Storage Centre in Linlithgow, Scotland. The hardware used for the tests is shown in Figure 11.

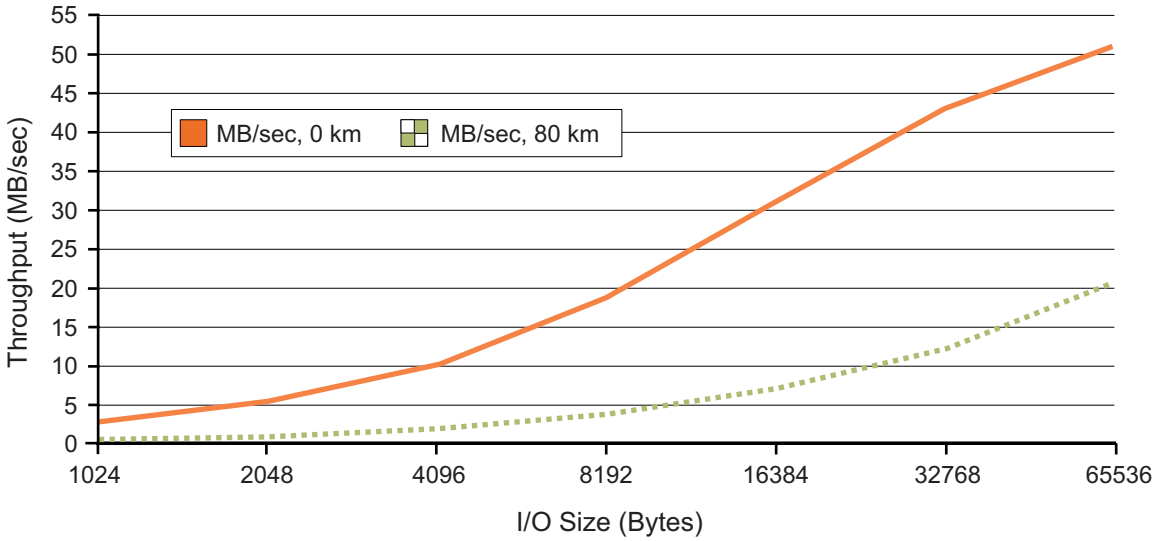
Figure 10. Hardware Set-up for Testing I/O Throughput Over Extended Distances



The inter-switch link ports on the SAN switches are configured with sufficient buffer credits to help ensure that the switch is not be the limiting factor. In this case, 55 additional buffer credits are configured. When only the standard 12 are configured, the extended distance performance degrades even further.

The throughput is then measured for both local (0 kilometers) and remote (80 kilometers) I/O, the results of which are illustrated in Figure 11. The tests measure throughput to raw disk (`/dev/rdsk/cXtYdZs0`) as opposed to the file system to avoid the effects of file system page cache that can distort the results. This approach equates to applications that open file system data files using the `O_SYNC` flag, e.g., Oracle for its online redo logs.

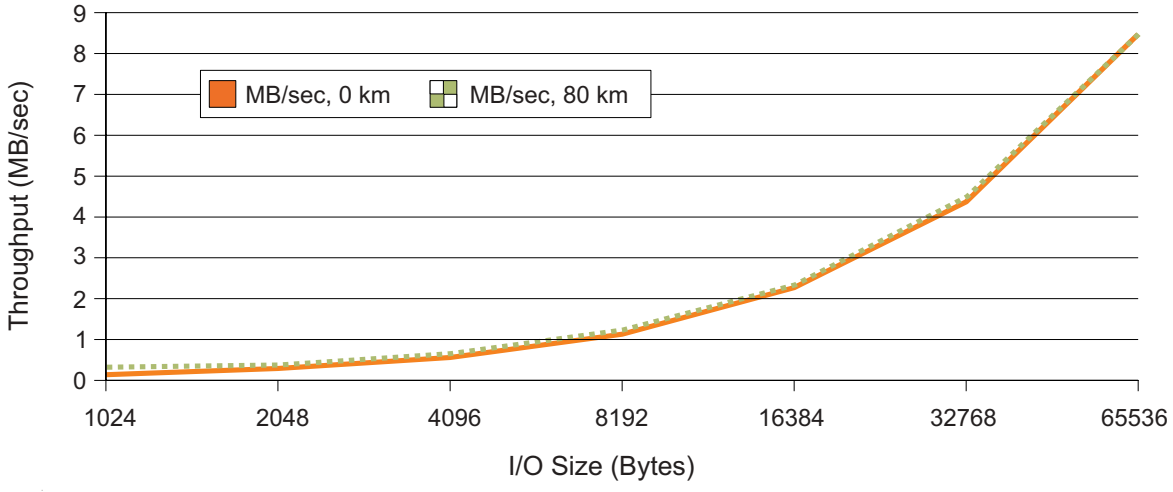
Figure 11. Raw, Single-Threaded, Sequential Write I/O Throughput



Single threaded, sequential raw writes show significant degradation at 80 kilometers. This is because the local storage array has a response time of 1 millisecond, whereas the remote operations consume a minimum of 2.6 milliseconds. If the array can (asynchronously) de-stage the data from cache faster than the requests are sent, then the cache never becomes full and the additional latency I/O remains the dominant factor.

In contrast, single threaded, random writes do not degrade nearly as much, as illustrated in Figure 12. Under load, the array has a service time of approximately 5 milliseconds. The extra 1.6 millisecond latency (80 x 1000 x 2 x 2 x 5ns) is hidden by the time it takes to de-stage the data from cache to disk. The ability to mask this extra latency is heavily dependent on any cache the storage sub-system contains and how effectively it can sustain the flow of data through the cache. If the cache is unable to keep up with the write request rate, then random write degradation may well tend toward that of sequential writes. The exact amount of degradation depends on precisely how much extra latency, caused by the extended distance, is passed on to the write. Furthermore, the performance of multi-threaded, random writes is again determined by whether the storage array's cache can keep up with the I/O demand. When the array is driven to saturation point, the additional latency is once again hidden by the de-staging time because it is typically substantially larger (8 to 10 milliseconds) than the extra latency (1 to 2 milliseconds) caused by the extended distance.

Figure 12. Raw, Single-Threaded, Random Write I/O Throughput



Single-threaded sequential write and multi-threaded random write performance characteristics are important when considering the processes that databases use to record transactions and checkpoint (or flush) their buffer cache respectively. Single-threaded write performance is the factor that determines write performance to a transaction log file, e.g., Oracle's online redo log. This in turn governs transaction commit time and thus perceived application performance. Multi-threaded random write performance governs the time it takes to flush a database's buffer cache. Increasing the time for a flush can result in greater contention for free buffer pages, thus lowering application performance once more.

It is worth noting that remote reads are never performed by a system that is correctly configured. Reads are always fulfilled from a *preferred* (local) mirror that is used unless there is a fault on this component.

Chapter 6
Summary

Data center architects should clearly understand the differences between the goals of creating availability and DR solutions in order to employ the appropriate combinations of hardware and software technologies to meet the RPO and RTO of each application service.

Availability solutions range from the availability inherent in single systems with redundant components, to home-grown availability solutions, to solutions available from vendors, with vendor solutions such as Sun Cluster software providing higher availability with less risk.

Depending on the RPO and RTO, cost-effective DR can be implemented using backup and recovery from tape, data replication, database replication, and metro clusters. For the highest level of availability and DR, a combination of clustering and data replication, such as provided by Sun Cluster Geographic Edition software, can be employed to build a solution that meets both availability and DR needs while providing the flexibility to allow disaster recovery plans to be tested before a real disaster occurs.

However, as this papers illustrates, the distance between primary and secondary sites incurs additional latency on I/O writes. This latency needs to be considered when designing long-distance DR solutions for applications that are latency sensitive.

About the Author

Tim Read has worked in the UK computer industry since 1985. He joined Sun in 1990, in a pre-sales role, as a Systems Engineer. He was responsible for the first Sun Cluster HA 1.0 deployments in UK and now works as Staff Engineer for Sun's N1 Availability Engineering group. Tim has authored a number Blueprints including co-authoring “Designing Enterprise Solutions with Sun Cluster 3.0". Tim holds a B.Sc. in Physics with Astrophysics from Birmingham University in the UK.

References

For information on the Data Center Reference Architecture Implementation see:
`http://www.sun.com/service/refarch/datacenter.html`

Oracle® Data Guard Concepts and Administration 10g Release 1 (10.1) Part No. B10823-01

Sun StorEdge Data Services V.3.0.0 release, Part No. 817-0122. Search for 817-0122 in
`http://docs.sun.com`

Sun StorEdge™ Availability Suite 3.2 Remote Mirror Software Configuration Guide, Part No. 817-3753-10,
`http://www.sun.com/products-n-solutions/hardware/docs/pdf/817-7380-10.pdf`

Data Replication Strategies, Jay Orcutt,
`http://www.sun.com/storage/white-papers/data_replication_strategies.pdf`

Sun™ Cluster Software — Quality by Design for Advanced Availability, Technical White Paper, `http://www.sun.com/software/cluster/wp-advancedavail/wp-advancedavail.pdf`

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` Web site enables you to access Sun technical documentation online. You can browse the archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`