Understanding the Benefits of Implementing Oracle RAC on Sun Cluster Software

Kristien Hens and Michael Loebmann

Sun BluePrints™ OnLine— January 2005

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 819-1466-10
Revision 1.0, 1/3/05
Edition: January 2005

Please
Recycle

Understanding the Benefits of Implementing Oracle RAC on Sun Cluster Software

Editor's Note – This article is the complete second chapter of the Sun BluePrints™ book, "Creating Highly Available Database Solutions: Oracle Real Application Clusters (RAC) and Sun Cluster 3.x Software," by Kristien Hens and Michael Loebmann, which is available from `www.sun.com/books`, amazon.com, and Barnes & Noble bookstores.

Oracle Real Application Cluster (RAC) rightly presents itself as the leading high availability (HA) technology provider with regard to commercial databases. A well-considered data center architecture plays a decisive role in this context. Configuring the underlying hardware that enables a demanding technology like RAC to work properly is a challenge. Sun has substantial experience in this area to help you develop an architecture that will support HA solutions.

In solutions that implement Oracle RAC and Sun™ Cluster software, the flexibility and power of Sun's cluster solution can add structure and maintainability to various underlying hardware components. Sun Cluster software offers a number of valuable features that set it apart from other cluster managers:

- Remote Shared Memory (RSM) functionality interconnects such as Scalable Coherent Interconnect (SCI) that improve clustering performance
- Failure fencing algorithms that ensure the integrity of data
- Customized agents (non-cluster-aware applications controlled by the Sun Cluster framework) that can be deployed with Oracle RAC on the same cluster, enabling the consolidation of different applications on the same cluster
- Detailed configuration matrix that provides a smooth and stable basis for RAC
- Predeveloped HA agents that greatly simplify HA solutions

This chapter contains the following sections:

Using Oracle on the Solaris Operating System

The Solaris™ Operating System (Solaris OS) provides a feature-rich platform for the Oracle database. Various products for resource management, volume management, and file system management provide features that enable you to configure systems to provide the required levels of performance, availability, and manageability. This section focuses on the availability and performance benefits of using an Oracle database on the Solaris OS.

Availability

More than ever, systems must provide high levels of availability. The reliability and recoverability of the underlying storage plays a critical role in supporting these requirements. To ensure the reliability and recoverability of a system, we recommend that you focus on minimizing restart times and using simple, well-tested configurations.

The amount of time required for a file system to recover to a known state after an unexpected outage is referred to as its *restart time*. When a system crashed or a power outage occurred in early implementations of UNIX® file systems (UFS), the file system was left in an unknown state. As a result, consistency checking occurred at boot time, a situation that often caused restarts to take a very long time (for example, three or more hours for a 50-gigabyte file system). This situation improved when *logging* was introduced to prevent the file system structure from becoming corrupted during a power outage or a system failure. During logging, file system metadata changes made to on-disk data were logged in separate sequential rolling logs so that the system maintained an accurate picture of the file system state in the event of a power outage or system crash. This functionality enables you to check the file system log and correct the last few updates as necessary, rather than scanning

the entire file system. A logging file system can mean the difference between mounting a heavily populated file system in twenty seconds instead of three hours without a log.

Starting with the Solaris 7 OS, the Solaris UFS has an option to enable logging. We recommend the use of this feature for any system that must provide high levels of availability. Because of the small number of file creations, deletions, and size changes, there is little performance overhead for file system logging when it is used for Oracle tablespaces. For this reason, it makes sense to enable logging with all file systems used for Oracle applications.

Performance

This section describes the functionalities that have the greatest impact on file system performance and compares them with the performance achieved when the raw disk path is used.

Note – In general, the fewer software layers between an application and the hardware, the better the performance. But this also leads to higher complexity and less robustness. The performance versus availability trade-off will always exist.

Bypassing the File System Cache With Direct I/O

An Oracle database caches tables in its own cache, which is held within the database shared global area, known as the *database block buffer cache*. That cache is sized at database startup according to the `db_cache_size` instance parameter. Database reads and writes are cached in a block buffer cache so that subsequent access to the same blocks do not need to reread data from the OS.

The Solaris OS also buffers reads through its file system cache, which means that by default, each read is potentially cached twice: once by the Solaris OS and once by Oracle. In addition to double caching, there is also CPU overhead for the code that manages the OS file system cache. A more efficient way for Oracle to perform I/O is to have the OS bypass its file system cache and read blocks from disk straight into the Oracle block buffer cache. To do this, use a file system feature known as direct I/O. Direct I/O uses a code path similar to that used when the raw disk path is used, but rather than bypassing the file system completely, it bypasses the file system cache. This has the advantage of retaining the regular file system administration model and provides an efficient code path similar to that for a raw disk.

Since the Solaris OS version 2.6, an option for direct I/O access for the UFS file system is provided. Enable the direct I/O feature by mounting the file system with the `forcedirectio` option as follows:

```
# mount -o forcedirectio /dev/dsk/c0t1d0s2 /db1
```

To enable `forcedirectio` on a file system that has already been mounted, use the following command:

```
# mount -o remount,[no]forcedirectio /dev/dsk/c0t1d0s2 /db1
```

The direct I/O path provides the following substantial performance improvements:

- Eliminates double buffering
- Provides a small, efficient code path for reads and writes
- Removes memory pressure from the OS

The Solaris OS uses the virtual memory system to implement file system caching, and as a result, many page-in operations result when data is read from disk. With direct I/O enabled, database reads do not need to involve the virtual memory system. Because using direct I/O means we no longer use the file system cache, be careful to correctly size the database block buffer cache. This is also a good reason to run 64-bit Oracle on 64-bit Solaris: it enables you to create a much bigger database buffer cache.

Enabling the Direct I/O Feature for Log Files

The system writes to the database transaction log synchronously during update transactions, and the log holds the data that are updated during each transaction. In the event of a failure, the transaction can be rolled back, leaving the data in the tables in their initial state before the transaction began. Therefore, each transaction involves writing to the transaction log. If the log is improperly configured, it can easily become a bottleneck.

The performance of the I/O subsystem servicing the database transaction log requires special attention. The I/O performance of the transaction log file directly impacts the update performance of the entire database instance. Because Oracle batches multiple transaction, many small updates can be rolled into fewer large writes. The writes are completed in a synchronous mode; therefore, the database can be sure that transaction log information has reached nonvolatile storage. Batching writes can provide a significant performance improvement because the cost of performing a large I/O is often not much greater than that of a small I/O.

By default, Solaris file systems break synchronous writes into 8-kilobyte units, so a single 64-kilobyte write will be performed as eight 8-kilobyte synchronous writes. Therefore, regardless of the size of the Oracle I/O, logs are written to as individual 8-kilobyte transactions, indirectly limiting the log throughput to the number of synchronous I/Os the underlying device can perform.

The direct I/O feature eliminates this behavior, allowing large writes to complete as a single I/O. Enabling direct I/O for the transaction log allows the Oracle log writer to efficiently batch log file writes, eliminating the log file as a bottleneck and allowing scalable throughput. As of the release of the Solaris 8 1/01 OS, direct I/O eliminates single writer locks on entire files, referred to as concurrent direct I/O.

Overlapping Data Writes With Asynchronous I/O

Asynchronous I/O is a mechanism to achieve I/O concurrency for greater performance on multidisk storage. The Oracle database writes data though central database writer processes. The database writers optionally use the Solaris asynchronous I/O mechanism for issuing writes to the OS, allowing multiple overlapping outstanding writes. The alternative slower mechanism, used when asynchronous I/O is disabled, issues individual writes by using the `pwrite()` system call at the expense of limiting the concurrency of database table writes.

To achieve maximum performance, a high degree of concurrency is required so that multiple overlapping write I/Os can be queued. You can configure the `pwrite()` database writer to allow a small amount of concurrency by using multiple processes to issue writes (up to 10 database writer processes can be started). Although this approach provides a small amount of concurrency, it does not provide the same level of write and update performance as the full asynchronous I/O implementation.

If you are using Oracle 9*i*, asynchronous I/O is enabled by default. If you have to enable it on another version of Oracle, set the Oracle instance parameter `disk_async_io`. When set to `true`, the database writer issues writes by using the `aio_write()` system calls and will overlap up to 3072 writes. The concurrent writes are handled by OS threads, allowing a single Oracle database writer process to issue asynchronous writes.

Reducing System Calls With Kernel Asynchronous I/O

Another minor optimization for asynchronous I/O is available when the raw disk path is used. When using raw disks, the Solaris OS uses kernel threads to manage the asynchronous I/O requests, lowering the amount of time spent context-switching and lowering the number of system calls required. This optimization can sometimes provide a slight performance improvement for the raw disk path.

Assessing the Impact of Using the Single Writer Lock

Solaris implements a per-file reader/writer lock in accordance with POSIX semantics. When writes are performed synchronously, the reader/writer lock ensures consistent write ordering by writing data to disk in the order in which they were issued from the application. The Solaris OS guarantees ordering for synchronous writes by allowing only one writer to a single file at any time. For the duration of a write, no other reads or writes can occur. This can significantly impact Oracle database performance because Oracle uses synchronous I/O for table writes.

The overhead of the single writer lock varies. For a system with little or no write/update component, there is little overhead. For update intensive workloads, there can be substantial overhead. Oracle products are designed to work correctly with raw disks, which means that Oracle it organizes the order by which data are is read and written before handing them to the OS. Therefore, the OS does not need to do this organization, thereby allowing further optimizations within the OS.

Direct I/O that eliminates the single writer lock (known as concurrent direct I/O) was available in the Solaris 8 1/01 OS. This allows the UFS to perform nearly as well as raw disks when used with the Oracle database.

Understanding the Value of Oracle and Sun Mature Technologies

In addition to the benefits of using Oracle with the Sun OS, there is inherent value in using mature technologies, such as those produced by Oracle and Sun. The technology used to support HA is not new, and Oracle and Sun have worked together for years to create products that successfully support HA. In 1991, Oracle released the Oracle Parallel Server (OPS) for production sites. Four years later, in 1995, Sun released its Sun Cluster product. The effort and time these two companies have invested in developing products and services that work together to support their customers' HA requirements has produced mature technologies that are in many ways superior to other products in the marketplace.

Note – For a more detailed discussion on the effort Oracle and Sun have put into creating products that support HA, the evolution of cache fusion and Sun Cluster 2.x software, and ways in which early advancements have affected the current state of database technology, refer to Appendix A of this book.

Understanding the Robustness of Sun Cluster Software

Sun Cluster 3.x software protects against single hardware or software failures such as node crashes or service interruptions. The software monitors the status of hardware and software components and initiates appropriate actions if a problem or failure occurs. For better reliability and performance, Sun Cluster 3.x software is tightly integrated with the Solaris OS. This integration speeds up error detection times and makes the software stack more robust. Depending on the type of failure to which it is responding, Sun Cluster 3.x software will either fail over the affected services to another node in the cluster or try to restart them. In all cases, the highest priority is to maintain data integrity and minimize application service downtime regardless of the failure. This requirement drives the layout of the infrastructure and all of the algorithms in the product.

The robustness of Sun Cluster 3.x software is one of the major reasons that it has achieved success in many areas. First, Sun Cluster software supports an extensive certification matrix of hardware and software. This matrix is the result of thorough testing, which ensures that a proposed configuration offers the requested functionality. Second, most of the Sun Cluster framework is implemented as kernel-level modules; a very high priority can therefore be given to detecting failed paths and nodes. In addition, the internal algorithms provided by Sun Cluster software make it virtually impossible for data or cluster database inconsistencies to exist.

Certification

Sun Cluster 3.x software supports a wide range of hardware from Sun, including servers and storage devices. In addition, the product is positioned to leverage the wide range of high-speed interconnects that are becoming very prominent in the industry. All the new-generation hardware from Sun will be supported by Sun Cluster 3.x software. New servers and storage are certified after development. Therefore, the range of possible configurations that are thoroughly tested and guaranteed to work in a Sun Cluster environment is always expanding. The Sun Cluster Open Storage Program (OSP) is an example of the types of programs Sun offers to provide fully tested and fully qualified products.

Understanding the Benefits of the Sun Cluster Open Storage Program

The Sun Cluster OSP enables storage vendors to qualify their hardware for use with Sun Cluster software. OSP provides these benefits:

- **Enhanced choice and investment protection.** You can now choose from a wider range of storage arrays to deploy with Sun Cluster software. If you have standardized on arrays other than Sun StorEdge arrays, you can now use third-party arrays with your mission-critical Sun Cluster deployments.
- **Reduced risk because of higher quality and joint support.** The storage you choose through the Sun Cluster OSP has been rigorously tested by the Sun Cluster Automated Test Environment (SCATE) software, and test results have been jointly reviewed and approved. In addition, joint support ensures that call facilitation occurs between Sun and the storage vendor through TSA Net. For more information about TSA Net, refer to `http://tsanet.org`.

When you are developing an HA solution, it is important to understand how the data path components in a cluster configuration can impact the availability of services. For example, data path components are Fibre Channel drivers and multipathing software. It is obvious that these components must be fully tested and fully certified with all possible storage options.

Sun's data path technology contains I/O path components to help you architect, implement, and manage mission-critical services. The components that are available in the SPARC® version of the Solaris OS include the UFS, Solaris™ Volume Manager software, and Sun StorEdge Traffic Manager software (previously known as multiplex I/O [MPxIO]). In addition, Sun storage management products are designed with Storage Management Specification Initiative (SMI-S) to operate in a heterogeneous environment. The Sun Cluster OSP offers these technology components to storage vendors, along with the automated test suite SCATE and test specifications, to support interoperability with a storage array.

SCATE-certified arrays provide an interoperability matrix from which you can choose thousands of cluster configurations with confidence when designing an HA service. Sun provides user documentation, white papers, and best practices recommendations you can use to implement and manage an HA architecture. Sun also works with various storage vendors to provide post-sales support.

For more information about the Sun Cluster OSP, the joint interoperability matrix, and joint support for storage devices with Sun Cluster software, refer to `http://www.sun.com/clusters`.

Understanding the Benefits of SCATE Software

SCATE software is a fully automated modular test suite that validates the rate, robustness, and recovery (R^3) framework availability model. The suite includes a large set of tools for qualifying the HA aspects of hardware and software in a clustered environment. SCATE software includes an innovative state-of-the-art test environment with comprehensive functional, load, and fault capabilities to verify end-to-end functionality of the entire cluster. SCATE software's automated test suites and tools can qualify data services, server platforms, network interface cards (NICs) for public network and private interconnects, and storage devices. These storage devices include arrays, host bus adapters (HBAs), switches, and hubs. It is currently used by the Sun Cluster Software Engineering group to perform internal tests before releasing new versions and updates of Sun Cluster software.

Sun also distributes the suite to a select group of independent hardware vendors (IHVs) and independent software vendors (ISVs) to help them test and qualify their products in a Sun Cluster software environment. Using SCATE software, product vendors and product groups within Sun (servers, storage, software, and the like) can qualify their products to work with Sun Cluster software. In addition, the SunService[SM] service can use SCATE software to identify problems with specific components or configurations, and customers can use it to validate a cluster during various phases of development or deployment. You can use it to verify that changes to existing cluster configurations, such as OS upgrades, applications upgrades, and patches, will not introduce new problems.

In addition to running the tests that are included as part of the SCATE software public distribution, Sun Cluster Software Engineering runs additional tests by injecting faults at the source code level. These tests are primarily aimed at helping customers deliver on their service level guarantees. Additionally, SCATE software features tools to validate the integration of Sun Cluster software with the following products, providing improved choice for clustered environments:

- Solaris Operating System Volume managers (Solaris Volume Manager, VERITAS Volume Manager/Cluster Volume Manager)
- File systems (UFS, Sun StorEdge QFS software, VERITAS file system)
- Sun StorEdge Availability Suite

Kernel-Level Cluster

Structured procedural and object-oriented software paradigms have dramatically improved the speed and quality of software development. The premise of this paradigm is that by defining clean interfaces between data or functional blocks of code, you can build on functions written by someone else so that you avoid reinventing existing code. There is also a tremendous gain in productivity and quality due to the reduction in complexity that naturally occurs in such

environments. Indeed, these paradigms work very well when your software is designed based on the assumption that the platform works. Unfortunately, this same reduction in complexity works against you when you are designing for the case in which the platform doesn't work. This is especially true when the platform has only partially failed. Handling these exceptions is vitally important when providing an HA platform. The intimate integration of Sun Cluster software into the OS platform resulted from the need to be able to quickly and efficiently handle exceptions.

Consider a simplified I/O software and hardware stack that contains the following items:

- Application
- File system
- Block device
- SCSI protocol
- HBA
- Wire
- Disk

For data to be read from or written directly to the disk, the data must be transferred through several layers. You can add redundancy at several places in this stack to help improve the resilience against faults in the stack below. For example, you might add a logical volume manager between the file system and block device. In some sense, each layer in the stack shields its upper layer from faults below it. You can add redundancy by including timeouts and retries. As you add layers to the stack, you also add to the timeouts and retries that might be required when a low-level exception occurs. Some faults might be transient, and the timeout/retry cycle will effectively compensate for the faults. For other faults, take immediate action at the application layer and do not wait even a few minutes for all the accumulated timeout/retry cycles.

Another problem with this layered approach is that the interfaces are standardized and some of the standards are quite old. This results in an abstraction of the actions of the lower stack layers. For example, a reservation conflict is one of the faults you would expect to find in a cluster that preserves data integrity for cluster members. Sun Cluster software uses SCSI-2 reserve/release reservations or SCSI-3 persistent group reservations (PGR) for fencing the data from nodes that are not participating in the running cluster. However, from an application developer's perspective, a `read`(2) or `write`(2) system call does not return an error code that indicates a reservation conflict. Somewhere along the stack, the fault will be translated into something like an I/O error. The context of the fault is lost. Further, if the application developer doesn't actually check the return code and implement the proper error handling, the program could continue with unknown consequences.

The intimacy between Sun Cluster software and the OS helps solve this problem in a consistent and timely manner. Sun Cluster software activates a failfast mode in the device driver for all shared disks. If a SCSI reservation conflict is detected, then the node will panic. This is detected by the other nodes in the cluster, and the application will be restarted on a surviving cluster node.

To prevent latent reservation conflicts, Sun Cluster software periodically polls the failfast-enabled drives to ensure that they have not been fenced away from the node. This polling enables the application to be restarted on a surviving node even if the application was not actively attempting I/O to the shared disks.

The fact that most of the Sun Cluster components run in the kernel makes it virtually impossible for the cluster functionality to be impaired by userland processes that run wild. Moreover, having most communication between the cluster components taking place in the kernel ensures that context-switching is reduced to a minimum, thus reducing fault detection quickly and accurately. For example, the cluster transport facility reacts swiftly when it detects that another node is dead. Any true HA solution should be based, at least partly, on a kernel-level cluster.

In addition to delivering efficient fault detection, the tight integration of Sun Cluster software with the Solaris kernel enables a tightly coupled cluster to seamlessly use core Solaris system services such as file systems, devices, and networks, while maintaining full Solaris software compatibility for existing applications. The software provides continuous availability and enables application service management through features such as global file services, global networking services with internal load balancing, and global devices that provide continuous availability to the customer.

Advanced Data Protection

In any HA solution, HA can never be an aim in itself and must be achieved by guaranteeing data consistency at all times. If even a slight amount of data could be corrupted, it is always better to give up a node or even the entire cluster. This might seem contradictory, but consider the situation in which a cluster node detects the death of another cluster node and continues functioning without going through some kind of reconfiguration. Then consider what would happen if only the interconnect paths have failed and the other node continues functioning. Because an HA cluster usually consists of data that are accessible through all nodes, the situation we're considering could lead to two incarnations of the HA application running on both nodes simultaneously and accessing the same data without being aware of each other's existence and without data synchronization. As you can imagine, that can lead to severe data corruption, which would require you to restore the data, even if the physical data are protected by RAID mechanisms. A restoration takes much more recovery time than giving up one of the cluster nodes or, in extreme cases, the entire cluster, and waiting for manual intervention. Further, the situation would be even worse if the corrupted data went undetected for some time, and hours, days, or even weeks of production were lost. Availability is about reducing recovery times.

For this reason, we firmly recommend that you use clustering to guarantee the integrity and consistency of data in the cluster configuration, membership information, and shared data access at all times.

Chapter 4 describes the tight quorum algorithm and data fencing strategies in Sun Cluster software that you can use to protect data.

Understanding Options for Accessing Memory Remotely

As the IT industry moves toward horizontally scaling application architectures, the demand for low-latency and high-bandwidth remote memory access protocols will most likely increase. In response to these requirements, Sun will leverage its experience with RSM, while embracing emerging standards-based APIs and new interconnect technologies to increase scalability and reduce the overall cost of deploying these kinds of architectures.

RSM-Capable Interconnect

The RSM implementation in Sun Cluster software enables a user process on a given cluster node to directly access memory on another cluster node. RSM accesses the other node's memory through memory-based interconnect hardware such as SCI. RSM on top of SCI is a mature and safe choice that is currently deployed in many mission-critical environments.

Because of SCI's high data rate and low latency, you can use it with Sun Cluster software or with the Sun high-performance-computing (HPC) cluster tools to improve the performance of most clustered applications. For commercial applications, such as parallel databases, that use Sun Cluster 3.x software, SCI is an ideal high-performance option. HA features such as link striping and hot swapping are designed to satisfy the most demanding data center requirements.

Using RSM, SCI can transform data between nodes by using the power of the hardware to replace the communications overhead of TCP/IP. Further, SCI cluster interconnects offer latencies that are much lower than Gigabit Ethernet (GBE). If your interconnect is the bottleneck, the price/performance of SCI can help get more overall system balance and, therefore, investment protection of the server investment. Bandwidth for SCI also exceeds that for GBE, and with the support of dual SCI NICs for failover and performance, SCI offers excellent bandwidth capacity. The combination of SCI's high data rates and low latency produce an overall superior clustering performance.

Compared with the traditional default protocol, Universal Datagram Protocol, which runs on all interconnects, RSM enjoys all the advantages of a low-latency protocol: low overhead, proven improved scalability, and, especially important for supporting

HA, a reliable protocol. The hardware offloading and bypassing of the networking stack means more saved CPU cycles, which in turn means that more CPUs are available for Oracle database processing. Thus, you get more useful work done with your CPUs, leading to significant cost savings.

With Oracle RAC, which shares data among different instances, data are synchronized through the cluster's private interconnect. With Sun Cluster software, this private interconnect can be based on SCI interfaces. In this case, Oracle RAC can use the RSM functionality of SCI to bypass the TCP/IP stack and write directly to the cache of other instances.

RSM is an excellent solution for applications that keep state in memory and need to access data from a remote node. Information about the possible future of remote memory access protocols is presented in "Looking at the Future of Oracle and Sun Cluster" on page 20.

Understanding Options for Storing Data

Sun Cluster software offers options for storing the data files of an Oracle RAC database. Traditionally, you could only use raw devices. Because raw devices still provide the best performance result, we highly recommended that at a minimum, you place the redo logs on a raw device. If you prefer managing file systems, you could also choose to use a shared QFS file system or a Network Attached Storage (NAS) solution. The following sections provide a brief overview of these options.

Raw Devices

Many customers are reluctant to use raw devices because they are concerned that they will lose flexibility or convenience. In our opinion, the use of raw devices is by no means inferior to using a file system.

Customers often give us the following reasons for deciding not to use of raw devices. Most of these beliefs are misconceptions, and those that are valid have simple solutions, as described below.

- Raw volumes are not visible.
- Raw volumes are difficult to size and design.
- Fixing full tablespaces is difficult with raw volumes.
- Backing up raw volumes is difficult.
- Oracle performs better when accessing data from file systems.
- The future of raw volumes is uncertain.

The following paragraphs explain how to easily resolve issues involved with the use of raw devices and dispel some of the misconceptions people have regarding their use.

- **Raw volumes are not visible.** The UNIX `ls` command is always cited as a method for providing information about the size of the Oracle database. However, you can really only gain accurate information about the number of blocks written by Oracle, the degree to which a raw device is occupied by Oracle data, and the size of a tablespace by using tools like the Oracle Enterprise Manager (OEM) or other third-party products. Because you can use these types of tools on raw devices or file systems, the way you choose to store your Oracle files is irrelevant.
- **Tablespaces composed of data files that are based on raw volumes are difficult to size and design.** It is true that in most cases Oracle data files are almost as big as their underlying raw device, and thus the use of the autoextend option is difficult or even impossible. However, we recommend that you always plan, as accurately as possible, the size and average increase of tablespaces. This recommendation applies whether you are using file systems or raw devices. An advantage of using a file system is that it enables you to put several tablespaces in one file system. With raw volumes, however, a one-to-one correlation of data file and volume would make sense, so raw volumes become slightly more difficult to use in an environment with many data files.
- **Fixing full tablespaces is difficult with raw volumes.** When tablespaces become full, you need to enlarge either the raw volumes or the file system. Customers often incorrectly think that administering raw volumes is more difficult than administering file systems. However, enlarging an entire file system involves expanding a volume with Solaris Volume Manager or VERITAS Volume Manager (VxVM) commands and then growing the entire file system. It is actually easier to enlarge raw volumes, which don't require you to perform the second step and allow you to grow only the volume containing the full or nearly full tablespace.
- **Backing up raw volumes is difficult.** You can back up file systems or raw devices with commercial tools like Oracle Recovery Manager (RMAN). For features built into the Solaris OS, the `dd` command does for raw volumes what the `cp` command does for file systems.
- **Oracle performs better when accessing data from file systems.** Oracle inherently expects to encounter raw devices when accessing data, and it is optimized for this case. In other words, the introduction of a file system prevents Oracle from efficiently reading and writing data. The Solaris OS offers a proven file system option called direct I/O, which circumvents the file system cache, thus enabling you to achieve almost the same performance with file systems as you would with raw devices. However, when performance is an issue, using raw volumes is the preferred method for running Oracle databases on the Solaris OS.

- **The future of raw volumes is uncertain.** Oracle tends to use as few third-party technologies (for example, file systems, volume managers, and cluster software) as it can. With the advent of Oracle 10*g*, Oracle has a feature called Automatic Storage Management (ASM), which deals with only raw devices but adds a layer of manageability to them.

In a solution implemented with Oracle RAC and Sun Cluster software, we recommend that you use the following options with regard to raw volumes:

- **Solaris Volume Manager software.** As of the Solaris 9 9/04 OS and Sun Cluster 3.1 9/04, Solaris Volume Manager software has multi-owner diskset functionality. Disksets can be taken on all nodes needing direct access to the volumes in these disksets. If you are running this version of the Solaris OS and want to use raw devices, using Solaris Volume Manager software is the preferred method for taking disksets on nodes that need direct access. Solaris Volume Manager software is included with the Solaris OS and offers the same ease of management as VxVM.
- **Partitions on raw `/dev/did` devices**. If your storage hardware supports both disk redundancy through hardware RAID 1 or RAID 5 and path redundancy through Sun StorEdge Traffic Manager, Hitachi Data Systems (HDS) Dynamic Link Manager, or EMC PowerPath, you can choose not to use any volume manager. This simplifies cluster reconfiguration because volume managers always introduce an extra layer that has to be reconfigured.
- **VxVM with the Cluster Volume Manager license.** This extra VxVM license enables you to create shared diskgroups, which means each cluster node can directly write to the volumes in this diskgroup. Use this option when you want to use a volume manager to manage raw devices and you are using a version of the Solaris OS that is older than Solaris 9 9/04 OS.
- **Oracle 10*g* Automatic Storage Management (ASM).** This new feature in Oracle 10*g* enables a dedicated Oracle instance to manage raw devices and eliminates the restriction that there can be only one data file for each raw device. ASM allows you to create mirrors and stripes over raw `/dev/did` partitions, Solaris Volume Manager, or VxVM volumes.

File Systems

If you want to use file systems for your Oracle data files, you have the following options:

- **Shared QFS.** QFS is a high-performance file system tuned to the needs of applications that work with a relatively small number of big files, such as databases. It can be shared among different servers, in which case every server can directly read from and write to the file system by means of the Storage Area Network (SAN).

Use shared QFS if you want near-raw performance and want the look and feel of a UNIX file system.

- **Network Attached Storage.** NAS is hard-disk storage that has its own network address instead of being attached to a dedicated server. A NAS device is attached to a local area network (LAN) and is assigned an IP address. File requests are mapped by the main server to the NAS file server. Oracle has strict requirements for NAS devices and supports only devices that have been validated for use with Oracle databases. In the Solaris OS, file requests are mapped by the main server to the NAS file server with the NFS protocol.

Supporting Enterprise Continuity Solutions

In addition to the data protection solutions Sun offers to help you manage risk, prepare for disasters, and improve business continuity, Sun and Nortel Networks have developed the Enterprise Continuity Solution. This offering is a geographically dispersed, near-instantaneous, data and application protection solution. An active/active data continuance configuration, Enterprise Continuity helps ensure virtually nonstop protection of mission-critical data and applications over distances of up to 200 kilometers. It also supports Oracle RAC and is the first architecture to support Oracle RAC in a campus cluster scenario over distances up to 200 kilometers.

Architecting and implementing a successful business continuance solution that protects against nearly any event imaginable involves the following activities:

- Ensuring data availability and consistency
- Ensuring server and application availability
- Networking enterprise components
- Leveraging skilled people and tested processes

The following sections describe how these activities support an Enterprise Continuity solution.

Data Availability and Consistency

In the global e-commerce environment, businesses rely on computing functions and electronically stored data for day-to-day operations. The information derived from interpretation of these data can often translate into competitive advantages. The first step in developing an HA Enterprise Continuity solution is to ensure that data are

protected from corruption and failures and are consistent between sites. Consolidating storage resources onto SANs can help make these tasks easier and more cost effective.

SANs completely separate the server and storage network from the LAN. A SAN is a separate network, behind the server, that is isolated from the LAN and is optimized to move data between disks, tape devices, and servers. SANs use Fibre Channel for network transport and move data to and from disks with the SCSI-3 protocol (serial SCSI). Combining the high-speed Fibre Channel transport with existing SCSI protocols enables high bandwidth, high capacity, high availability, and manageable data storage networks.

Managing and Protecting Data

Managing and protecting data in an Enterprise Continuity solution primarily entails the following activities:

- Protecting data
- Consolidating storage resources
- Sharing data
- Providing data consistency

You can use various Sun solutions to perform these tasks. For more information about these activities and the tools available to accomplish them, refer to `http://www.sun.com/solutions/`.

Server and Application Availability

Once data are separated and protected, the second step in designing an Enterprise Continuity solution is to provide HA servers and application services. The globally networked economy that exists today demands powerful information systems that are available to customers, business partners, and employees around the clock and around the world. When a company's application is down, business costs such as lost productivity, reduced sales and profits, poor customer service, and decreased customer loyalty can be substantial. Wherever advanced systems availability and productivity are critical for competitiveness, HA applications are required.

To provide continuous application availability, you must protect end users from failures of both server hardware and application software.

Protecting Against Server Failures

To protect against individual server failures, two types of redundancy are required:

- **Component redundancy.** This type delivers fault-resilience and tolerance by automatically failing over to redundant components within a single server.
- **Server redundancy.** This type enables a secondary server to take over if the primary server experiences a failure.

Sun servers offer a high degree of component redundancy and other reliability, availability, and serviceability (RAS) features that make them excellent for serving HA applications. Server redundancy is implemented with Sun Cluster 3.x software. For more information about Sun Cluster software, refer to "Understanding the Robustness of Sun Cluster Software" on page 7.

We recommend that, in addition to Sun servers and software, you use quorum devices to help the cluster decide which nodes form a new subcluster in case of any failure. The availability of the quorum device is key in a disaster situation. For common two-site infrastructures, if the site containing the quorum device is lost, manual procedures must be followed to bring the cluster back online in the other data center. In a three-site infrastructure, a quorum device is in the third site, so the loss of one site does not affect the majority of quorum votes.

Ensuring Application Availability

Application downtime or poor performance is unacceptable in a world of 24×7 access, e-commerce, the Internet, and application service providers. When customers can switch suppliers at the click of a button, high levels of service are vital. Sun Cluster 3.x software helps maintain high levels of application availability and scalability. Because data are now accessible by multiple hosts, processing can be distributed across more than one server. Applications can be load-balanced across a number of specialized servers, and database applications can be consolidated and maintained separately.

Enterprise Components

Once the cluster applications, computing, and storage aspects of an Enterprise Continuity solution have been addressed, you can focus on connecting all of them. The underpinning of any successful data service is a scalable, available, and secure network. Such a network is divided into access and transport segments. The transport segment is crucial for HA because it provides a method for separating data centers across a significant distances. In particular, the transport network needs to significantly exceed the 10-kilometer limit associated with Fibre Channel protocol.

In addition, any complete networking plan needs to consider how users will be re-routed in the event of a failure, and how a system will be recovered in the event of disaster. The following sections briefly address these issues.

Accessing Data From Distant Servers With an Access Network

At this point, data are separated from servers, and servers and applications are set up to deal with failure situations. However, it is not sufficient to simply have secondary servers available for emergencies. Applications must also automatically redirect client requests to the applicable failover servers; you can do that by adding a mediation service to the HA data.

Long distance application clusters require public network access across distances. In traditional local cluster implementations, applications fail over locally to a physically separate, colocated server. This secondary server is also directly accessible through the LAN, thereby avoiding redirection of user data traffic. Now that the failover server has been moved to the other side of the MAN, it is essential to ensure that user requests can reach the failover server and that replies are returned across the MAN to the user. The access network specific to the Enterprise Continuity solution consists of aggregation and load-balancing devices for the data network, and Fibre Channel fabric switches for the storage network.

Transporting Data Over Large Distances With a Transport Network

If all the data processing equipment is at a single site, a failure of the site caused by man-made or natural disasters can put a company out of business. Disaster protection procedures that include the ability to recover at a remote location need to be in place. A disaster recovery strategy depends on several factors, including the allowable time a system can be down and how back-level the data can be following the recovery operation.

SAN internetworking provides an infrastructure that enables new disaster recovery solutions to protect data over long distances. One of these solutions is the synchronous mirroring of data over a SAN/Dense Wave Division Multiplexer (DWDM) infrastructure to the hot-stand-by system by site. When the primary site fails, a remote system takes over, using the mirrored volumes. Single-mode long wavelength GBICs support distances up to 10 kilometers. SANs can also utilize WANs and MANs to extend the distance between production and recovery sites. This extended fabric feature enables native Fibre Channel connectivity up to 200 kilometers, using DWDM at nearly 200 megabytes per second.

The Enterprise Continuity solution employs an optical transport network to meet the distance requirement to extend Fibre Channel. The optical network uses WDM technology, which permits the native transport of different protocols such as Fibre Channel and Gigabit Ethernet across the same fiber, maximizing bandwidth with a minimal number of network devices and connections. This technology enables the crucial distance extension for clustering and SANs in the Enterprise Continuity solution.

Skilled People and Tested Processes and Products

Technology alone will not address all aspects of continuous service availability. To ensure the highest levels of business continuity, businesses must invest in three essential components: people, processes, and products. A well-trained staff armed with thoroughly tested architectures and procedures and a robust Enterprise Continuity infrastructure are the best defense against detrimental site outages.

Looking at the Future of Oracle and Sun Cluster

In September 2003, Oracle Corporation launched the newest release of its database product, Oracle10*g*. This release focuses on grid computing and targets both enterprise and midmarket customers. Oracle 10*g* includes its own clustering and volume management software, not only for platforms like Linux and Windows but also for UNIX platforms.

While the future of these products and their functionality is not certain, we anticipate that in many cases, the configuration of Oracle 10*g* and Sun Cluster software will continue to be a superior solution to one without Sun's clustering software. Furthermore, the Sun Cluster implementation on the x86 platform deserves closer attention in terms of the Oracle 10*g* port. In addition, the advent of RSM in connection with InfiniBand should provide new dimensions in terms of performance and reliability.

Better, faster, and less expensive interconnects will play a major role in future data center architectures. Together with new Solaris OS capabilities, these enhancements will allow internode communication to bypass traditional protocol stacks and other mechanisms that increase latency.

Further, the availability of User Direct Access Programming Library (uDAPL) in the Solaris 10 OS, coupled with Oracle's planned support for uDAPL in 10*g* RAC, positions Sun customers to benefit from the open-standards-based, lower-cost Remote Direct Memory Access (RDMA) interconnects that are on the horizon.

Note – uDAPL is a standard API that provides a userland transport framework to enable applications to make efficient use of the RDMA capability in InfiniBand (IB) interconnects. uDAPL is defined by the Direct Access Transfer (DAT) collaborative `www.datcollaborative.org`, which provides test programs, specifications, and design documentation.

The uDAPL APIs on top of IB offer functionality similar to that of RSM/ Reliable Data Transport (RDT) on SCI. The uDAPL enables applications like Oracle RAC to leverage a single source code base on both Linux and Solaris platforms, thereby reducing development, testing, and system integration time.

In addition, uDAPL is designed to be transport independent, which is very important because it means application developers can write to uDAPL and plug in different kinds of RDMA capable hardware transports, including IB and future RDMA interconnects over Ethernet. By standardizing on uDAPL, developers can take advantage of any hardware transport that satisfies their needs.

About the Authors

Kristien Hens is currently working as a Technology System Engineer, with focus on storage and cluster technologies. Before that she worked for four years for Sun Educational Services. She has taught storage and Sun Cluster courses to customers and Sun engineers. She has also developed and taught the courses 'Oracle 9i for Sun Engineers' and Oracle 9i RAC on Sun Cluster 3.x Workshop'. She is co-author of three Sun BluePrints OnLine articles.

Michael Loebmann has been the local senior systems engineer for the ISV Oracle in Germany for three years.His tasks include both the exchange of technical knowledge between Oracle and Sun and customer consultation with regard to competition. He has been working in the computer field since 1990, Oracle specifically since 1993. He now has over nine years of experience with Oracle products, especially in Oracle Datawarehousing and parallel Oracle databases (OPS and RAC). His specialities include Oracle Performance Tuning on Solaris and the investigation of database performance on storage subsystems.

Ordering Sun Documents

The SunDocs^{SM} 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 `docs.sun.com` 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`