Introduction to SunTone™ Clustered Database Platforms

Ted Persky and Richard Elling—Enterprise Engineering

Sun BluePrints™ OnLine - March, 2002

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-4514-10
Revision 1.0, 03/18/02
Edition: March 2002

Please
Recycle

Introduction to SunTone™ Clustered Database Platforms

This article presents the benefits of SunTone™ Clustered Database Platforms, describes the services of the management server, compares Oracle configurations, and covers tuning system parameters. This article presents options and recommends best practices for storage arrays, boot disk mirroring, and back up and recovery.

Relating it to Sun's "Three Big Bets," this article uses the soon-to-be announced Clustered Database Platform 280/3 (CDP 280/3) as a reference platform for describing benefits derived from providing Sun customers with preinstalled, "ready-to-deploy" clustered database systems.

This article contains the following topics:

Introduction

Sun identifies the integrated platform (stack) as one of its “Three Big Bets” as it moves forward implementing best practices for the data center. Agreement is widespread in the industry that a need exists for integrated stacks, however, the question of exactly what constitutes an integrated stack, particularly in clusters and high availability, remains to be answered. The integrated stack elevates the process of procuring large application or database servers to its next logical level. The integration and reliability of hardware and software components need to be seamless.

Let’s compare the data center’s availability with some familiar examples. As is observed in the reliability of modern telephone switches, system downtime is not an option. To transform the data center to the next logical level, Sun Microsystems provides SunTone™ platforms. These enterprise servers provide continuous availability, such as the ever-present dial tone you expect when picking up the telephone.

Acquiring and installing a data center server needs to be as simple as ordering telephone service. When you call up the local telephone company to establish service, you need not concern yourself with questions such as:

- How will the telephone line be strung between my house and the central switching office?
- Does the telephone switch have enough capacity to handle the calls I make?
- Is the software in my central office compatible with that used by the customers I call in another state?

All these issues are handled by making one call to the telephone company’s business office.

Similarly, when you purchase a new car, you need not ask the dealer, “By the way, could you please tell me the part number for the bumpers? I want to make sure I include them on the order.” As ludicrous as this statement appears, situations like these are symbolic of the situations that data center managers and system administrators (SAs) have been forced to deal with for years.

Advances in cluster technology within the past decade lend credence to the idea of being able to provide a database or application service akin to the “number of nines” availability and reliability we expect from the telephone dial tone. In addition to database or application availability and reliability, factor in ease of use, installation, ordering, and interoperability.

Reference Platform

To illustrate the benefits of implementing preinstalled, ready-to-deploy clustered database systems, we focus on the soon-to-be released Cluster Database Platform 280/3 (CDP 280/3).

Hardware

The hardware for our reference platform consists of the following:

- two Sun Fire™ 280R servers
- one Sun Fire V120 management server (follow-up to Netra™ t1 AC200 server)
- two mirrored Sun StorEdge™ T3 arrays, all mounted in a 72-inch Sun StorEdge expansion cabinet

Everything ships from the factory already installed and cabled in the cabinet, reducing the time previously required to ensure that all components were connected and communicating properly.

As expected, the entire system ships with redundancy already factored into the design. For example, the cluster has dual, private interconnects between the Sun Fire 280R server nodes, and the public network interface is ready to be configured into a failover group. Each Sun StorEdge™ T3 array contains a hot spare disk to improve data recovery times in case of a disk failure.

The management server ships as a pre-installed JumpStart™ server; each Sun Fire 280R server cluster node arrives with no operating environment installed and is a JumpStart client ready to be installed.

Software

The key software components for our reference platform consist of the following:

- Solaris™ 8 Operating Environment (Solaris OE)
- Sun™ Cluster 3.0 software
- Oracle9i Release 1 with Real Application Clusters (RAC, previously referred to as ORACLE Parallel Server)
- Solaris Volume Manager software (previously known as Solstice DiskSuite™ software)
- VERITAS Volume Manager™ (VxVM software)

Benefits

Sun has thoroughly tested the interoperability of the products (hardware and software) that comprise SunTone Clustered Database Platforms. This pretesting relieves a customer's IT department from the hassle of having to hunt down the certification matrices on the web sites of Sun, ORACLE®, and VERITAS.

Even if, for some reason, a customer needs to call Sun's or ORACLE's support line, the support specialist immediately has a reliable reference of the customer's hardware and software configuration, because the customer is running a SunTone Clustered Database Platform.

Streamlined Configuration

Through a simple, interactive question-and-answer session, an SA can configure the cluster nodes on-demand to support either ORACLE RAC[1] with shared storage managed by VxVM software or High Availability (HA) ORACLE[2] accessing a Global File System maintained by either VxVM software or Solaris Volume Manager software.

The installation process is highly automated and much more simplified than what an SA would typically perform when installing new hardware components, integrating existing hardware, and installing new software.

Custom (site specific) configuration details such as cluster name, cluster node names, Internet Protocol (IP) addresses, time zone, and so on are defined by the SA at the installation site during the installation process.

The installation process is referred to as the cluster installation, and one of the accompanying documents titled the *CDP Installation Guide* provides detailed information. Note that the cluster installation process is streamlined and is far less effort than that required if the customer had to install all the software from CDs.

1. Both cluster nodes actively process transactions.

2. One cluster node is active, the other passive, awaiting any failure on the first.

Pre-Installed Management Server

A typical installation including Solaris OE, VxVM software, and ORACLE would include answering potentially hundreds of questions and loading potentially dozens of software CDs. By including the management server, we move the software installation to the network and are able to pre-answer many of the standard configuration questions.

Updated Patches

The most up-to-date patches are applied to each software product, as of the time of product code freeze. Any SA who has waded through pages of patch reports to construct the correct order in which to apply them would call this a significant benefit, in and of itself.

Careful attention is paid to ensure that all high-priority security patches for each product are applied.

Applied Cluster and Database Packages

Your choice of JumpStart software configuration (RAC or HA ORACLE) determines a suite of additional cluster and ORACLE packages that must be applied and patched in a specific order.

After you install and configure the software, a database appropriate for your choice of JumpStart software configuration cluster (active/active in the case of RAC; active/passive in the case of HA ORACLE) starts. Now it's ready to accept user connections and create application tables.

When done manually, this process requires a fair amount of training and hands-on experience to master; otherwise, the cluster node can reach an unstable state. Here are some estimates of the training required:

- An experienced UNIX® SA would need about five days of class time to learn Sun Cluster 3.0 software.
- A database administrator (DBA) already well versed in Oracle9i would require another five days of training to learn the RAC option.
- After returning from class, even the brightest students would need approximately two weeks each of lab time to feel comfortable undertaking the installation and configuration of a production database cluster.

Using the traditional approach, 240 person-hours are spent preparing for a cluster installation that now can be completed within a matter of hours, using the JumpStart software functionality in CDP 280/3.

An alternative to training in-house employees would be to hire short-term consultants to perform the installation, which can cost as much as or more than the 240 hours spent training employees.

Even if a manager decides to send employees to the 80 hours combined of UNIX operating system and RAC training, it is better for the employees to return to work, being immediately productive on a cluster they know has been installed correctly. In this scenario, the employees acquire additional experience from an installation that is running, rather than suffering through the tedium of verifying every patch, coping with initial node panics, and troubleshooting a myriad of other problems.

After the cluster is successfully up and running, there is no special SunTone Clustered Database Platform training required to configure and maintain the cluster.

Applying Future Patches

After a cluster installation is implemented, an SA can easily:

- apply new Solaris OE patches the same as with any other cluster
- perform ORACLE upgrades as indicated in the Release Notes supplied by ORACLE
- apply high-profile security patches for any included software product, announced after shipment of CDP 280/3

Rebuilding the Cluster

After a cluster installation is implemented, an SA can rebuild the cluster easily and automatically. An SA can use the time saved to gain more experience with the cluster, without the worry of lengthy rebuilds if a mistake is made.

Services

The primary purpose of the management server is to monitor and manage the cluster environment. The majority of the management server functions are beyond the normal management services associated with the Sun Cluster 3.0 software environment. You can supply these services to other systems as computing resources in the management server allow.

The management server has enough CPU power and memory resources to consolidate logging messages from the cluster nodes and to implement the following services:

- Sun Management Center software
- SunPlex™ software
- Sun StorEdge Component Manager™ software
- Network Time Protocol (NTP)
- AnswerBook2™ software
- Sun Ray™ hardware

The Sun Cluster software cluster control panel (ccp) graphical user interface (GUI) is installed on the management server to enable easy access to the cluster node consoles. This GUI enables the SA to execute commands concurrently on all cluster nodes; however, it does not provide a management interface. The ccp works with a terminal concentrator (TC) to provide a seamless connection to serial port consoles. See the following figure for details.

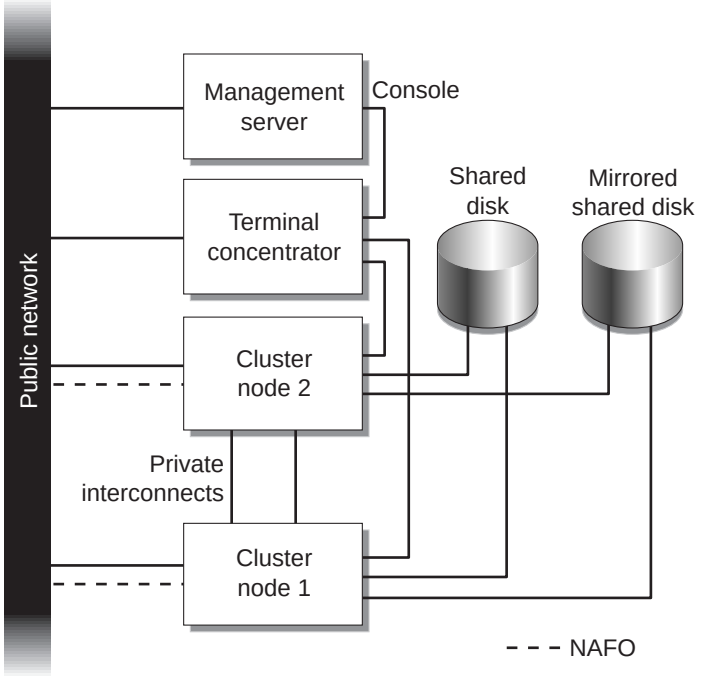

FIGURE 1 Connections Between the Management Server and Cluster Nodes

For increased availability, the management server includes a second internal SCSI disk to mirror the boot drive. Mirroring using the Solaris Volume Manager software takes place automatically during the on-site configuration.

To eliminate the need for a keyboard, mouse, and monitor, a TC connects the cluster nodes to the console port of the management server, thus providing console access to all cluster nodes.

Configuring the Management Server

You can configure the management server as a central repository for logs and messages (informational, warning, and error) for each cluster node; this configuration is similar to Sun Fire™ server domain logging by the System Service Processor.

Modify the `/etc/syslog.conf` files on each cluster node to route system messages to the management server's SYSLOG service. This routing allows easy correlation of cluster node events over time to improve cluster management. In addition, you can review cluster node events on the management server when a cluster node is down.

It is important to note that the management server itself does not constitute a critical component in the operation of a cluster. All cluster operations continue without impact if the management server fails. The default configuration for Sun Cluster 3.0 software has no dependency on it. In case of a management server failure, implement an alternative, operational procedure to access the system messages and console devices of servers to maintain serviceability of the cluster components.

The Sun Management Center software has a log file filter that allows you to define patterns of messages that should be highlighted or ignored, and to quickly identify key messages in the midst of more routine informational messages. This log file filter enables easy analysis of the correlating log files.

Every aspect of managing, securing, planning, and debugging a network involves determining when events happen. Time is the critical element that allows an event on one network node to be mapped to a corresponding event on another. In many cases, these challenges can be overcome by the enterprise deployment of the NTP service. The management server includes the `xntpd(1M)` daemon, which is bundled with the Solaris OE software to provide time synchronization services to all cluster nodes.

Using ORACLE Database Configurations

This section addresses Solaris OE users and groups for accessing ORACLE software and using ORACLE database configurations. You can choose Real Application Clusters or HA ORACLE data services.

Solaris OE Users and Groups

The Solaris OE user `oracle` owns the ORACLE software and database files, whether on raw devices or UNIX® file system (UFS). Although the database components reside on the Sun StorEdge™ T3 arrays, the ORACLE software for each node always resides on that node's local storage, thereby providing redundancy.

The `oracle` user belongs to a primary group of `oinstall` and a secondary group named `dba`. Members of the `oinstall` group are responsible for maintaining the software on each cluster node and upgrading it as necessary. Members of the group `dba` create databases and maintain them.

If needed, an additional Solaris OE user can be defined with membership in group `dba` and not in group `oinstall`, thus giving the user DBA authority but withholding the ability to modify the software.

Real Application Clusters (RAC) Service

Real Application Clusters (RAC) allows two or more cluster nodes to perform transactions against a single database simultaneously. We use the term "active-active" or "scalable" to refer to this type of architecture. Multiple nodes synchronize their accesses to database objects.

Each node starts up an ORACLE database instance—comprised of the necessary background processes—such as the system monitor, process monitor, log writer, and database writer (SMON, PMON, LGWR, and DBWR, respectively). Furthermore, each node maintains its own System Global Area (SGA) in memory, including the Database Buffer Cache, the Redo Log Buffer, and the Shared Pool.

Distributed Lock Manager (DLM) processes run on each instance, synchronizing data block accesses, thus creating a memory-resident repository of lock objects equally distributed among all instances. Each instance is "mastering" a subset of the

distributed resource locks. Background processes that support the DLM include the Global Enqueue Service Monitor (LMON) and the Global Enqueue Service Daemon (LMD). See the following figure.

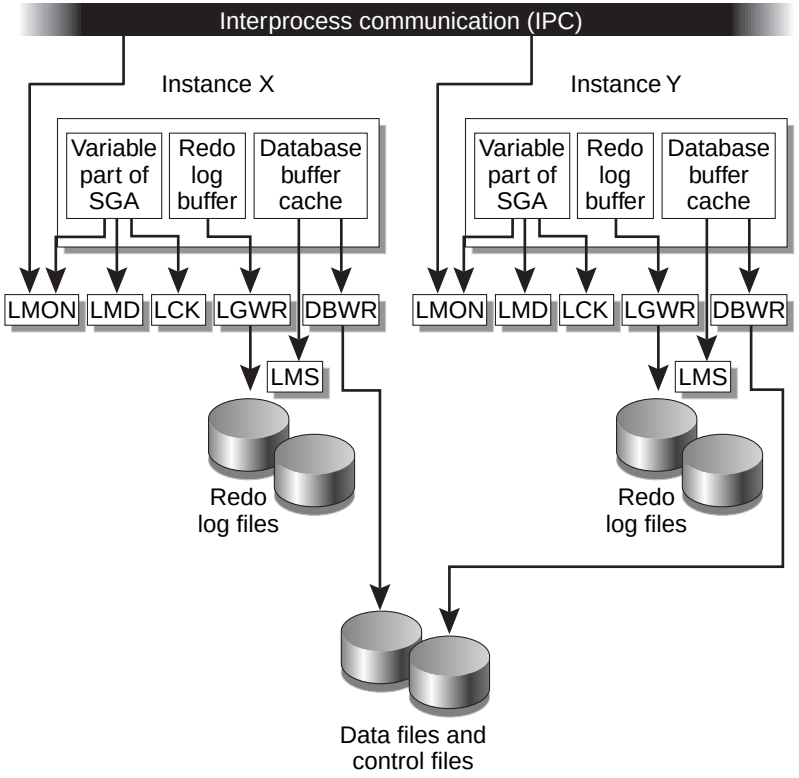

FIGURE 2 Real Application Clusters (RAC) Background Processes and Memory Structures

On shared storage, the database instance of each node is assigned its own redo log files and rollback segments. Redo log files are the way a database recovers to a consistent state, following a system crash. These files record changes made to blocks of any object, including tables, indices, and rollback segments. These files provide a way to guarantee that all committed transactions are preserved in the event of a crash, even if the resulting data block changes are not written to data files. Rollback segments store database “undo” information. For example, they store the information needed to cancel or “roll back” a transaction, if the application needs to do so. Also, rollback segments provide a form of SQL statement isolation. A long-running query against a set of tables must only see their contents as they were at the time the query began.

As with any RAC installation, CDP 280/3 implements shared storage on top of raw volumes, in this case built using VxVM software. Using raw volumes allows each ORACLE instance to access the other instance's redo log files and rollback segments, particularly in the event of a node failure.

Recovering a Database Instance

When a node leaves the cluster, the resources it was mastering need to be remastered on the surviving node. With the improved hashing algorithm introduced in Oracle9i, locks already mastered on the surviving instance are not affected. At the same time remastering is occurring, the SMON process on the surviving node performs instance recovery. All transactions that were performed on the failed instance are recorded in its redo log files, but only those transactions committed prior to the newest checkpoint are guaranteed to be written out to data files.

Because the redo log files for both instances reside on raw devices, the instance performing recovery can access the failed instance's redo log, either committing to the data files those transactions that had committed after the final checkpoint (also known as "rolling forward"), or rolling back those that had not. If it rolls back uncommitted transactions, it does so by reading "before images" found in the failed instance's rollback segments.

SMON also frees up any resources that pending transactions may have acquired. During a roll forward period, the database is only partially available; a surviving instance can only access data blocks it currently has cached. It can not perform any I/O to the database, nor can it ask for any additional resource locks during this period. Rolling back uncommitted transactions can occur in parallel with the creation of new work.

Maintaining Availability

Using ORACLE's Transparent Application Failover (TAF), client connections performing read-only queries can continue on the surviving node, unaware that the original instance has failed. Users might experience longer response times, because the query must be restarted from the beginning against a "cold" database buffer cache on the surviving instance, which cannot commence until it has recovered the failed instance.

Applications performing data modifications must be specially coded to handle a status code returned from the ORACLE Call Interface (OCI), indicating that an instance has failed, recognizing such a return code has occurred, and reconnecting to the surviving instance for further processing of the statement.

In the unlikely event that all instances fail, the first instance restarted after the failure performs instance recovery on the redo log files of all failed instances, including its own, if necessary. This action is known as "database crash recovery."

When the nodes in a CDP 280/3 cluster are first installed via JumpStart software, a general-purpose RAC database is initialized:

- First, the node that VxVM software determines is "mastering" the shared storage creates a series of mirrored, raw volumes, each with Dirty Region Logging™ (DRL) enabled.
- Next it copies each component file of the database from local, file-system based storage out to its corresponding raw volume.
- Using the host name chosen during the installation question-and-answer session, it configures the Oracle listener file, `listener.ora`, and initiates the listener process daemon.
- It defines two service names in the `tnsnames.ora` file for the newly installed database, `orcl1`, for a direct connection into the instance of the first node, and `orcl`, which provides a load balancing scheme.
- Client connections that use a service name of `orcl` connect to the RAC instance with the lightest load observed at the time the connections are requested.
- Finally, it brings up the SGA and background processes that form the `orcl1` instance on the first node.

The remaining cluster node may be thought of as the "slave," with regards to VxVM software. During its initial JumpStart software process, it must wait for the master node to complete the creation and initialization of the raw volumes that form the `orcl` database on shared storage.

Once per minute (for a maximum of 15 minutes), it "wakes up" to see if the volumes are ready. If they are, it begins its own listener process and instance, in this case named `orcl2`. No type of recovery work is necessary. The `orcl2` instance begins to record the modification of database blocks to its redo log files and the prior state of those blocks to its rollback segments.

Of course, this synchronization of nodes against the creation of raw volumes only occurs the first time each node is installed in the cluster. From that point forward, if a node needs to be rebooted, it rejoins the cluster and automatically starts up its Oracle instance and listener daemon.

The `orcl` starter database is based upon a "general purpose" database configuration that ORACLE supplies its customers via the ORACLE Universal Installer™ (OUI). This installer implements a database with an eight-Kbytes block size, useful for either On-line Transaction Processing (OLTP) or Data Warehouse applications. The data center DBA is free to modify initialization parameters, except for block size, to tune the instances on each node. By tuning the instances, the DBA can more effectively support a particular OLTP or warehouse environment, for example, to resize SGA components[1] or to alter the behavior of background processes. At this point the database appears to the DBA no different than one that the DBA might

1. Dynamic resizing of the SGA (for example, no instance restart is necessary) is not configured into CDP 280/3. For details on how to enable Solaris OE Dynamic Intimate Shared Memory to allow dynamic SGA resizing, please see ORACLE Metalink Note No. 151222.1 on `http://metalink.oracle.com`.

have created manually. In fact, the DBA can create additional raw volumes on the storage array and use the ORACLE Database Creation Assistant to build other RAC databases, resulting in one or more instances running on each cluster node, each instance serving a particular underlying database.

HA ORACLE Data Service

HA ORACLE data service allows only one cluster node to host transactions against a database at a time. Using JumpStart software, cluster nodes are started as a "failover resource group," to borrow from Sun Cluster 3.0 software terminology. This approach results in a highly available, ORACLE data service.

ORACLE registers its database and listener services to the cluster via Sun Cluster Resource Group Manager (RGM), which chooses one of the nodes to host them. A storage resource (representing the Sun StorEdge™ T3 arrays) and a logical host name and IP address are registered with RGM. The node chosen to host the ORACLE resources hosts the storage and logical host name too. Client sessions connect to the database service using the logical host name, which is distinct from the physical host name assigned to each node.

When it is necessary to switch ORACLE data services to another node in the cluster, the storage resource changes hosts. The logical IP address "floats" to the other node so that any packets routed to that address are then handled by the subsequent node. Note that this data services switch can occur either by request or by reason of a failed node. In the first situation (perhaps a maintenance window is required on the node currently hosting the services), ORACLE performs a graceful database shut down on the first node and simply restarts on the second node; HA ORACLE data service is unaware that it is now running on a different host.

In the case of node failure, the Sun Cluster software's "heartbeat" on the surviving node determines that the departing node has left the cluster and proceeds to rehost the logical host name, IP address, and storage. ORACLE's SMON process starts up and performs crash recovery in "roll forward" and "roll back" phases. Unlike the case of RAC, the DLM remastering does not occur, because DLM processes are not needed in a configuration such as HA ORACLE data service, where only one instance is alive at any given time. Once again, HA ORACLE data service is not aware that it is restarting on a different host, just that an instance crash occurred. In either case, client connections disconnect and reconnect to the surviving node, after it has completed failover processing.

Because HA ORACLE data service only uses one active instance, only one ORACLE license needs to be purchased. Of course, availability suffers, because any type of cluster switch involves a finite amount of downtime, in addition to the need for clients to reconnect.

It is interesting to note that the term “shared storage” is actually a misnomer in the HA ORACLE architecture, because only one node is accessing a given logical unit at any point. Raw devices are not necessary in this configuration, because there is no concept of a second instance accessing the redo logs of the first. Database files reside on a UFS on the Sun StorEdge™ T3 arrays and are mounted under `/global`, which is visible by either node. Per installer request, this file system can be constructed on top of either Solaris Volume Manager software or VxVM software and is mounted using the direct I/O feature of Solaris OE. The Solaris OE UFS with direct I/O demonstrates nearly the same performance as raw devices, but with the ease of management of a file system. Solaris OE UFS logging is enabled by default to improve file system recovery.

Similar to the RAC configuration, a starter database is furnished; however, in the case of HA ORACLE data service, you choose between an OLTP and a Decision Support System (DSS) version, based upon the configurations ORACLE provided within OUI. Each version features an eight-Kbytes block size; however, the initialization parameter `SORT_AREA_SIZE`, which limits the amount of memory used internally by ORACLE for sorting result sets, is twice as large in the DSS version as it is in OLTP. Also similar to the RAC installation, the DBA can extend and further tune the starter database as needed.

Setting `/etc/system` Parameters

Some parameters must be set in the `/etc/system` file before the system can run Oracle9i and VxVM software. The following table lists and describes the `/etc/system` tunable parameters set for the CDP 280/3.

Parameter	Units	Solaris Default	Tuned Value	Description
maxusers	users	min (1 + MBs, RAM, 2048)	2048	Originally defined the number of users the system could support. Now, Solaris Operating Environment (Solaris OE) sizes mostly based on the physical memory on the system. Subsystems derived from maxusers: maximum processes, quota structures, and size of the directory name lookup cache (DNLC).
msgsys:msginfo_msgmax	bytes	2048	16384	Maximum size of a System V message.
msgsys:msginfo_msgmnb	bytes	4096	16384	Maximum bytes on any one message queue.
msgsys:msginfo_msgmni	queues	50	2200	Maximum message queues that can be created.
msgsys:msginfo_msgtql	messages	40	2500	Maximum messages that can be created. If a msgsnd(2) call attempts to exceed this limit, the request is deferred until a message header is available. Or, if the request has set the IPC_NOWAIT flag, the request fails with the error EAGAIN.
semsys:seminfo_semmni	identifiers	10	100	Maximum semaphore identifiers.
semsys:seminfo_semmns	semaphores	60	2500	Maximum System V semaphores on the system.
semsys:seminfo_semmnu	structures	30	2500	Total undo structures in the system.
semsys:seminfo_semmsl	semaphores	25	300	Maximum System V semaphores per semaphore identifier.
semsys:seminfo_semopm	operations	10	100	Maximum System V semaphore operations per semop(2) call. This parameter refers to the number of sembufs in the sops array that is provided to the semop system call.
semsys:seminfo_semume	structures	10	2500	Maximum System V semaphore undo structures used by any one process.
shmsys:shminfo_shmmax	bytes	1048576	0xffffffffffffffff	Maximum system V shared memory segment that can be created. This parameter is an upper limit that is checked before the system sees if it actually has the physical resources to create the requested memory segment. This parameter creates confusion and many problems for Sun customers. By setting it to a maximum number, the simple check performed at shared segment creation time is disabled. The maximum shared memory segment size is based solely on available memory for shared segments in the system.

Parameter	Units	Solaris Default	Tuned Value	Description
hmsys:shminfo_shmseg	segments	6	32	Limit of shared memory segments that any one process can create.
vxio:vol_default_iodelay	ticks	50	5	The count in clock ticks that utilities pause between issuing I/Os when the utilities are directed to throttle down the speed of their issuing I/Os, yet are not given a delay time. Utilities performing such operations as resynchronizing mirrors or rebuilding RAID-5 columns use this value. Increasing this value results in slower recovery operations and consequently lower system impact while recoveries are being performed. Lowering this value increases the speed at which mirrors are syncrhonized and recovered. The default setting limits the I/O load on the system so that mirror synchronization or RAID-5 recovery does not impact performance. With large disks, the synchronization and recovery time becomes very long, perhaps days for large logical volumes.
vxio:vol_maxio	sectors	512	10240	Controls the maximum size of logical I/O operations that can be performed without breaking up the request. Physical I/O requests larger than this value are broken up and performed synchronously. Physical I/Os are broken up based on the capabilities of the disk device and are unaffected by changes to this maximum logical request limit. File systems do not normally perform such large I/O operations.
vxio:vol_maxioctl	bytes	32768	131072	Controls the maximum size of data that can be passed into the VxVM via an ioctl(2) call.
vxio:vol_maxspecialio	sectors	512	10240	Maximum size of an I/O request that can be issued by an ioctl(2) call. Although the ioctl request itself can be small, it can request a large I/O request. Tuning limits the size of these I/O requests. If necessary, a request that exceeds this value can be failed, or the request can be broken up and performed synchronously.
vxio:vol_rootdev_is_volume	boolean	0	1	Sets a boolean flag value to true (1) once the root file system on the node is encapsulated and mirrored. NOTE: This flag is set by the VxVM boot disk encapsulation process; do not tune it manually.

Mirroring Shared Storage Arrays

The CDP 280/3 includes two Sun StorEdge T3 arrays, one mirrored to the other. A single array contains nine, 36-Gbyte disk drives, eight of which are arranged in a controller RAID-5 configuration (seven data disks plus one parity) with the remaining disk left over as a hot spare. The total amount of usable, shared disk storage is approximately 225 Gbytes (seven disks multiplied by 36 Gbytes each, minus overhead).

This configuration provides availability with reasonable storage capacity for the Sun StorEdge™ T3 arrays. You can change it if desired, as long as the resulting configuration follows accepted best practices. Each array is configured at the factory with a 32-Kbyte data stripe width.

When VxVM software is used to manage the shared storage (in all cases except HA ORACLE data service with Solaris Volume Manager software), the system makes use of DRL to minimize the time required to remirror an array, if one fails. A VERITAS volume constructed using DRL contains an additional log “subdisk” (to borrow from VxVM software terminology), which stores a recovery map and two active maps (one for each node). The region of the volume being modified is marked as dirty in the log and flushed from the RAID cache before the actual writes of the database data take place. Once the data are written to both the primary Sun StorEdge™ T3 array and its mirror, that region is again marked as clean in the log’s maps. Upon a system failure only those regions still marked dirty need to be applied between the primary and its mirror to bring the mirror up-to-date.

Mirroring may be implemented with Solaris Volume Manager software instead, if such a database configuration is chosen for HA ORACLE, as described previously.

As with the rest of the CDP 280/3, the Sun StorEdge™ T3 arrays are reconfigurable, in this case by using Sun StorEdge Component Manager software to access the RAID controller of each array.

Mirroring the Boot Disk

Mirroring the boot disk provides additional redundancy in both the management server and cluster nodes. The management server automatically mirrors its boot disk during the data center installation using Solaris Volume Manager software. The nodes are handled differently, depending upon the type of database installation you chose.

For Oracle9i RAC installations, the root disk on each node is encapsulated and mirrored using VxVM software. This mirroring is the only configuration possible.

For HA ORACLE data service, the boot disk is mirrored using the same software chosen to manage the globally accessible file systems: either VxVM software or Solaris Volume Manager software. Choose either of these, as is appropriate for your system.

Managing Back Up and Recovery

Successful data center operations require good backup, restore, and recovery processes. Good processes are critical when a data center is providing highly available services.

The management server is the focal point of Sun Cluster 3.0 system recovery. Recovery procedures for the management server itself are required. Because the management server acts as the JumpStart server for the cluster nodes, the management server plays an important part in the recovery of a cluster node.

Restoring Management Server Files

The management server contains many different files: JumpStart software profiles for cluster nodes, copies of Solaris OE used when installing clients, AnswerBook2 documentation, Sun Management Center software support files, and so on. Most of these files are static. You can restore these files from the distribution media. However, SYSLOG files change regularly and tend to be relatively small, so they require continuous backup.

Determining When to Perform Full or Incremental Back Ups

Using a local tape drive[1], you can do a full backup of the management server when major changes of the file systems occur. For example, perform a full back up when updating the JumpStart software directory structure with a new release of the Solaris OE.

Perform incremental backups to save SYSLOG and configuration files regularly.

Recovering the Management Server

A set of DVDs that contain the software image installed on the management server in the factory enables you to rebuild the management server to the factory-installed state. If a catastrophic failure causes the loss of the management server operating environment, and you cannot recover the operating environment from backup tape, use this recovery process. Recovering to the factory-installed state by using the DVDs is significantly faster than reloading the dozens of packages installed on the management server from their distribution DVDs.

Reinstalling Cluster Nodes

A DVD image is not provided for the cluster nodes. Reinstall a cluster node by running JumpStart software against it from the management server. This technique is also useful for switching among the various database configurations available. Once the nodes are fully installed and operational, follow standard, documented procedures for backing up Solaris OE routinely on them.[2]

Backing Up and Recovering the ORACLE Database

ORACLE offers a series of backup and recovery methods, each of which provides a finer grain recovery than the one preceding it. A “cold” backup requires shutting down all database instances and making a copy of each component (data files, control files, redo log files, etc.). Cold backups can only recover the database to the point in the past when the backup was last taken. Individual tables may be backed

1. To be purchased separately.

2. See *Solaris 8 System Administration Guide.*

up to a point in time and recovered using the ORACLE utilities Export and Import features, respectively. These default backup methods can be used after the nodes are installed and configured.

Using Archive Log Mode

To provide for up-to-the-minute recovery, the DBA places the database in archive log mode, specifying a destination to which each instance copies its online redo log files when they are full. Setting up archive log mode allows the DBA to perform “hot” backups while each instance remains on-line.

To issue a hot backup, the DBA codes a script, run from either instance, that causes each tablespace, one at a time, to be placed off-line using the `alter tablespace...begin backup` command.

Processing then copies each component file of the tablespace either to tape or to another directory before bringing the tablespace back on-line using `alter tablespace...end backup`. SQL can be issued to modify data in the tablespace while it is being backed up, but doing so requires additional redo log space. Therefore, we recommend that you perform hot backups when the database is likely to undergo few data modifications.

Using ORACLE’s Recovery Manager Utility

ORACLE’s Recovery Manager utility (RMAN) performs hot backups without placing each tablespace into backup mode. An additional database must be created to store the RMAN recovery catalogue. Alternatively, you can purchase ORACLE Data Guard software to set up a database replication site, physically removed from the RAC cluster. A site running Data Guard software operates in a continuous recovery mode, constantly processing redo log files it receives from the RAC instances. You can take advantage of the Data Guard software site to perform any necessary backups, leaving the cluster fully available around the clock.

Using ORACLE’s Flashback Query

Oracle9i introduced a new feature called Flashback Query, allowing a user to perform SQL against a table or set of tables, seeing their contents as they existed at some point in the past, as specified by the user.

Flashback Query requires the database to be placed into *automatic undo management* mode, another new feature of Oracle9i. Using automatic undo management frees the DBA from having to create individual rollback segments (also known as *undo*) for

each RAC instance. Instead, the DBA creates what is known as an *undo tablespace* for each instance, and Oracle9i takes care of creating and deleting rollback segments within each tablespace as needed.

To facilitate Flashback Query, the DBA specifies an undo retention period to indicate how far back an SQL statement may need to go, thus dictating the amount of undo information retained in each undo tablespace at any given time.

Obtaining Product Documentation

The CDP 280/3 system ships with a hardcopy *Installation Guide* and *Recovery Guide*. An *Operations Guide*, *Design Guide*, and *Product Release Notes* for CDP 280/3 are available on-line by connecting from a web browser on any client to the Apache Web Server, automatically listening on port 8,080 of the management server. Copies of these documents are posted for public use at:

```
http://www.sun.com/products-n-solutions/hardware/docs/
Integrated_Platforms/index.html
```

Sun Microsystems provides AnswerBook2 documentation on the management server. You can configure the management server to act as an AnswerBook2 server, providing content through a web browser running on any client. Additional documentation is accessible on-line at:

```
http://docs.sun.com
```

ORACLE includes its *Installation Guide*, as well as the *Administrator's Reference*, on each cluster node. Complete ORACLE documentation is available on-line at:

```
http://otn.oracle.com/docs/content.html
```

Note – Free registration on the site is necessary to access documentation.

Each node contains VERITAS manual pages. In-depth, on-line documents are available at the VERITAS web site:

```
http://support.veritas.com
```

Obtaining Support

Support for the components that comprise the CDP 280/3 follows normal support channels. The data center manager must set up licensing and support contracts with Sun Microsystems, ORACLE, and VERITAS before requesting support. Address problems or questions to the applicable company that produced the component in question. Advise the support representative that you are using a SunTone Clustered Database Platform. Temporary VERITAS licenses are installed on each cluster node at the customer site. To purchase permanent license keys, you must contact VERITAS.

Additional Resources

- *Solaris Tunable Parameters Reference Manual.* Part No. 806-4015. Sun Microsystems, Inc., 2000.
- *VERITAS Volume Manager 3.1.1 Administrator's Guide*. Part No. 30-000226-011. VERITAS Software Corp., 2000-2001.
- Elling, Richard and Tim Read. *Designing Enterprise Solutions with Sun Cluster 3.0.* ISBN No. 0-13-008458-1. Sun Microsystems Press, 2002.
- *Oracle9i RAC Concepts*. Part No. A89867-02. ORACLE Corp., 1996, 2001.

About the Authors

Ted Persky is a Staff Engineer and Database Administrator, specializing in ORACLE and Sybase. Ted was Sun's key database software integrator for the VERITAS ORACLE Sun (VOS) Initiative, as well as the CDP 280/3. Also, he has experience as a UNIX Systems Administrator. Ted holds a masters degree in computer science from The George Washington University.

Richard Elling is the Chief Architect for Enterprise Engineering at Sun Microsystems in San Diego, California. Richard was a Field Systems Engineer at Sun for five years. He was the Sun Worldwide Field Systems Engineer of the Year in 1996. Prior to working at Sun, he was the Manager of Network Support for the College of Engineering at Auburn University, a design engineer for a startup microelectronics company, and an engineer for NASA, performing electronic design and experiments integration for Space Shuttle missions.