

A large, decorative, light grey curved shape on the left side of the page, tapering towards the top and bottom.

SOLARIS™ CLUSTER 3.2 SOFTWARE: MAKING ORACLE DATABASE 10G R2 AND 11G RAC EVEN MORE “UNBREAKABLE”

White Paper
September 2008

Table of Contents

Introduction1
Highly Available and Scalable Database Services2
Solaris Cluster Software3
Oracle RAC manageability features3
Solaris Cluster 3.2 software scalable device group resources4
Solaris Cluster 3.2 software scalable mount point resources4
Solaris Cluster 3.2 CRS framework resource4
Solaris Cluster 3.2 RAC proxy resources5
Oracle Clusterware storage proxy resources5
The Benefits of Using Solaris Cluster Software With Oracle RAC6
Membership and failure fencing6
The Oracle Clusterware membership framework6
Solaris Cluster software: robust membership and fencing7
Device namespace management9
Private interconnect management9
Interconnect usage by Oracle RAC application traffic11
Oracle RAC data storage options11
Extensive portfolio of application agents12
Availability and Scalability Versus Cost13
Summary14
Acknowledgements14

Chapter 1

Introduction

Solaris™ Cluster 3.2, the latest version of the Solaris Cluster software, continues Sun's history of providing robust cluster platforms for Oracle Real Application Clusters (RAC) and Oracle Parallel Server (OPS), which dates back to the introduction of the SPARCcluster® PDB 1.0 software in June 1995. With an installed base of thousands of nodes, Solaris Cluster is the highly popular platform of choice for Oracle's parallel databases. Customers include Bankers Automated Clearing Services, People's Insurance Company of China, as well as Oracle.

When Oracle released its Oracle Database 10g product, the RAC component came with its own cluster framework, Oracle Clusterware (or Cluster Ready Services — CRS). This caused a great deal of confusion in the industry, with experts wondering if Solaris Cluster software was needed now that Oracle had its own cluster framework. The purpose of this paper is to explain the benefits gained by using Solaris Cluster software with Oracle Clusterware/RAC rather than just Oracle Clusterware/RAC alone.

This article is aimed at datacenter architects and briefly outlines the reasons users might choose an Oracle RAC solution for its availability and scalability characteristics. This is followed by a discussion of the new features of Solaris Sun Cluster 3.2 pertinent to Oracle Database 10g and 11g RAC. The concluding section further describes the technical features of Solaris Cluster software that help to reduce the risk of outage through increased protection against data corruption, increased performance through simpler networking, and reduced risk of configuration error through consistent device naming. In each case, these are compared to the capabilities offered by Oracle Clusterware alone.

Notes — This paper uses the term “Solaris Cluster software” to refer in a general sense to the software and uses the product name, Solaris Cluster 3.2, to refer specifically to the new version under discussion. Unless otherwise specified, the benefits derived from using Oracle 10g R2 RAC with Solaris Cluster also apply to the equivalent Oracle 11g RAC combination.

Chapter 2

Highly Available and Scalable Database Services

Many business services have a database component in them, which typically means a market-leading Oracle database is in the software stack. Oracle Database 10g R2 RAC has a built-in clustering component, Oracle Clusterware, that allows the database to run across multiple systems. By using a cluster to run the database, greater scalability and availability can be achieved. As nodes are added to the cluster, database instances can be added — potentially allowing more supported users and greater throughput. Further, by spreading the load over several nodes, the impact of failure is lowered because only users and transactions in progress on the failed node are affected.

Database availability and scalability is achieved only through the combination of the Oracle Clusterware's cluster membership and the coordination of data access through Oracle's Cache Fusion technology. For portability reasons, Oracle Database 10g RAC Clusterware is not tightly integrated with the host platform it supports. Consequently, there is scope for a native clustering product, such as Solaris Cluster, to offer greater data protection and faster failure discovery together with a range of other operating environment-specific features. These are discussed in greater detail in the following sections.

Chapter 3

Solaris Cluster Software

Solaris Cluster software is a mature product with an extensive, robust, and well tested feature set. It also has a substantial installed base of both highly available (HA) Oracle and parallel Oracle (OPS and RAC) deployments.

The two main features of the Cluster software are its membership framework (described in detail later in this article) and its Resource Group Manager (RGM).

The RGM coordinates the starting, stopping, and monitoring of resource groups. A resource group encapsulates the components, or resources, that define an application service (e.g., an IP address, an HAStoragePlus resource to fail-over file systems, an Oracle database, and an Oracle listener for an HA Oracle service.) The start, stop, and monitoring methods for a particular resource are contained in a resource type or agent, which are shell scripts or C programs that perform the necessary tasks.

Solaris Cluster software brings the added benefit that volume managers, such as Solaris™ Volume Manager and VERITAS Volume Manager software, can be used to manage the LUNs provided by the shared storage. Without this facility, users would be forced to rely on the capabilities of the storage array alone to present their storage in usable chunks.

The latest Solaris Cluster 3.2 software includes a number of features specifically designed to enhance the manageability of Oracle RAC deployments. These new resource types allow better coordination of Oracle start-up and shutdown in a clustered environment, avoiding any failures that might otherwise be incurred.

Oracle RAC manageability features

Oracle Clusterware has two important components: the Oracle Cluster Registry (OCR) and a set of voting disks. Together with a number of daemon processes, these make up the Oracle Clusterware framework.

When installing Oracle Clusterware on systems running Solaris Cluster 3.2, database administrators can choose to create the OCR and voting disks on either shared raw hardware RAID LUNs, raw volume manager devices, or a shared QFS or network attached storage (NAS) file system. When running without Solaris Cluster software, the options are reduced to raw disk or NAS.

Once Oracle Clusterware has started, it waits for the file systems or devices holding its OCR and voting disks to come online. However, if Oracle Clusterware is not stopped at the right point during a cluster or node shutdown (i.e., before any file systems it may depend on are unmounted) then I/O time-outs will occur and one or more nodes will be panicked because their voting disks are not responding.¹

1. <http://metalink.oracle.com/> Oracle MetaLink document 265769.1

Solaris Cluster 3.2 software provides a mechanism to create dependencies between Oracle Clusterware, the RAC databases, and the underlying storage to ensure that both the start-up and shutdown operations are correctly sequenced (described in detail in the following subsections). It should be reiterated that without Solaris Cluster software, Oracle Clusterware alone would not have any option for volume management or shared file system other than NAS on the Solaris OS.

Solaris Cluster 3.2 software scalable device group resources

Scalable device group resources are used to ensure that Solaris Volume Manager disksets or VERITAS Volume Manager diskgroups are correctly configured prior to allowing any dependent Solaris Cluster software resources to start. The validation mechanisms contained in the resource type check that the diskset or diskgroup is valid, multiowner, hosted on the given nodes, and has at least one volume present. The resource also supports a parameter to define an additional list of logical devices to check.

The continuing health of these sets is then checked regularly using either `metaset` or `vxprint` commands to ensure that no problems have arisen that might cause any dependent services to fail. If a problem is found, then the resource is put into a disabled state on a per-node basis.

Solaris Cluster 3.2 software scalable mount point resources

Scalable mount-point resources are used to ensure that shared QFS and NAS mounted file systems are functioning correctly. The validation mechanisms contained in the resource type check that any NAS file system defined is exported and any shared QFS file system has an entry in the `/etc/vfstab` and `/etc/opt/SUNWsamfs/mcf` files. For shared QFS file systems, it also checks that the QFS metadata server resource exists and that the scalable mount-point resource has a dependency on it.

The health of these file systems is probed by using I/O to preallocate test files created in the mount-point directory of the file system when the resource is started. If this succeeds, the file system is deemed to be healthy; if not, the resource is put into a disabled state on that node.

Solaris Cluster 3.2 CRS framework resource

The Solaris Cluster 3.2 CRS framework resource is used to allow the Oracle Clusterware to be stopped before the file systems, NAS or shared QFS, or the diskgroups and disksets are stopped. This is achieved by setting up strong dependencies from the CRS framework resource on the appropriate scalable mount-point and device-group resources.

In addition, the resource probes for the existence of the `ocssd.bin` process on a regular basis to check whether the Oracle Clusterware framework is still healthy. The current state is reflected in the output from the `clresourcegroup` and `cluster` commands and in the Solaris Cluster Manager browser interface.

Solaris Cluster 3.2 RAC proxy resources

Solaris Cluster 3.2 RAC proxy resources are used to coordinate the start-up and shutdown processes of Oracle RAC database instances, so a single cluster could contain more than one of these entities. An instance of the resource checks the validity of the Oracle RAC and Oracle Clusterware home directories and whether the Oracle database name corresponds to an existing database. If Oracle Clusterware is found to be running, the resource proceeds to start the database instances on the appropriate nodes. Conversely, if the resource is being stopped, it will shut down these instances.

The status of the Oracle instances is monitored using the Oracle Fast Application Notification (FAN) mechanism, which is used to forward events to the Solaris Cluster monitoring system. This allows the Solaris Cluster 3.2 software to report the status of the instances in the output of commands such as “cluster status -t resource” or “clresource status” and in the Solaris Cluster Manager browser interface. As with all Solaris Cluster resources, it can be disabled to allow maintenance of the Oracle database instances.

Oracle Clusterware storage proxy resources

The features described above allow higher levels of integration between Oracle Clusterware resources and Solaris Cluster 3.2 software scalable device group and scalable mount-point storage resources. The Solaris Cluster 3.2 clsetup program simplifies the creation of both sets of framework resources. The Oracle Clusterware resources that clsetup generates include a per-node dependency on the relevant Solaris Cluster 3.2 storage resource for the particular RAC database, ensuring that its instances are not started until the storage resources, on which they rely, are available.

Chapter 4

The Benefits of Using Solaris Cluster Software with Oracle RAC

Using Solaris Cluster software in addition to Oracle Clusterware can be viewed as a stronger insurance policy against outages caused by system hangs or misconfiguration, as well as providing additional data integrity guarantees. It is analogous to the “Defense in Depth” strategy employed to secure Data Centers against unauthorized access through the use of multiple security products. So put simply put, if availability is the key driving factor, Solaris Cluster software should be used.

The reasons for this are detailed below. Each subsection describes why the particular Solaris Cluster feature is important and discusses Oracle Clusterware’s capabilities in its absence.

Membership and failure fencing

This section contrasts how the Oracle Clusterware and Solaris Cluster software handle the critical concept of cluster membership and how failure fencing is implemented.

The Oracle Clusterware membership framework

Highly available, scalable clusters are built from multiple servers configured with redundant components for networking and storage. Despite this hardware redundancy, only one copy of the Oracle database data exists. So if it experiences data integrity problems, the database will have to be restored from tape or using Oracle’s flashback feature. The other alternative would be to switch to a standby, disaster-recovery database, assuming it had not picked up the corruption yet. That type of solution, however, would add considerable expense.

Consistency of the data is maintained by the Oracle Cache Fusion technology, which, in turn, relies on the cluster membership supplied by Oracle Clusterware’s Cluster Synchronization Service (CSS). CSS uses a Solaris OS process (cssd) to provide a heartbeat mechanism to monitor remote nodes via the interconnects and voting disks.² For components to be deemed healthy, they must respond to the probes within a set time-out period. When running on the Solaris OS without Solaris Cluster software, the default values for these in the Oracle Database 10g release 10.2.0.x are 30 seconds for the network heartbeat and 200 seconds for I/O to a voting disk. Consequently, Oracle Clusterware may wait a minimum of 30 seconds before initiating a cluster membership reconfiguration and ejecting a failed node. From 10.2.0.4 onwards, many of the Clusterware processes run in the real-time (RT) scheduling class. Although this helps to ensure that these processes are scheduled in preference to other timeshare class processes, they may still have to contend with kernel-mode threads for processor time. Kernel threads also obey scheduling priorities and most are typically assigned to the system

2. <https://metalink.oracle.com/> Oracle MetaLink document 294430.1

(SYS) scheduling class. However, an RT process entering the kernel for services such as disk I/O (for voting disks) or network traffic (for CRS heartbeats) may find themselves blocked pending the completion of the same service by non RT kernel threads and other lower than interrupt priority service processing. Thus the RT class itself does not guarantee responsiveness. Consequently, real node failures could take longer to detect, resulting in an extended wait before a reconfiguration begins — during which database services may also be unresponsive.

Node eviction also uses Solaris OS user-level commands to remove a failing or failed node. There is scope for Oracle Clusterware to evict a node from a cluster and yet not have that node be forced to the boot prompt immediately. The remaining nodes, however, receive the revised cluster membership and start recovering transactions from the evicted instance. If the evicted node continues to write data during the period between being requested to abort and reaching the boot prompt, then the database may be corrupted. This, in turn, can lead to a cascade of failures across all Oracle Clusterware nodes as each node in turn comes across this unexpected inconsistency.

Oracle Clusterware relies solely on the mechanisms described above to prevent unauthorized access to shared storage by its constituent members. Readers should contrast this with the Solaris Cluster software approach described below.

Solaris Cluster software: robust membership and fencing

The kernel-based Solaris Cluster framework is extremely robust and rigorous in determining the cluster membership. Nodes wishing to join the cluster must be able to communicate with existing cluster nodes over the private cluster heartbeat networks. Once communication has been established, the node registers special registration keys on private regions of each of the shared storage devices to indicate that they are valid members of the cluster.

The failfast daemon plays a critical role in protecting the integrity of data held on the cluster's shared storage and is started during the latter part of the Solaris OS boot process (`/etc/rc2.d/S75MOUNTGFSYS` on the Solaris 8 and 9 OS or the `sc_failfast` service under the Solaris 10 OS). The daemon issues an `ioctl` to the `sd` and `ssd` drivers to set the `MHIOCENFAILFAST` flag³ on all shared storage devices. Key cluster processes, such as `rgmd`, `ucmmd`, and `udlm`, whose failure would also jeopardize cluster integrity then register with the failfast daemon. If one of these processes fails, the failfast daemon allows the process 30 seconds to dump its core for diagnostic purposes before panicking the local node. If, on the other hand, the failfast daemon detects that the local node no longer has its expected access rights to shared storage, then the daemon immediately panics the local node, thus preventing any possibility of data corruption.

3. <http://docs.sun.com/app/docs/doc/816-5177/6mbbc4g8a?a=view>

During normal operation, each cluster node regularly communicates with all of its peers over each of their mutually shared private heartbeat networks. (A Solaris Cluster system can be configured with between two and six of these networks.) These networks are also the transport for the Oracle Cache Fusion traffic (see section “Private interconnect management”).

Every node sends out low-level DLPI packets once per second to each of its peers on each of the private networks.⁴ These packets are sent in the kernel interrupt context, making them very resilient to the peaks in system load. A network, or path, between two nodes is only declared down after 10 such packets are missed, although this can be tuned to a lower value if the cluster load characteristics are bounded and well understood. As a result, node failure detection is three times faster than Oracle Clusterware and is done without the risk that its accuracy will be compromised by high levels of system load.

In the event that the cluster partitions — one subset of nodes cannot communicate with another subset of nodes (i.e., all the private interconnect paths are down) — the membership monitor goes through a series of lockstep stages to compute a new membership. The process by which Solaris Cluster determines the new membership is described in detail in the section titled Quorum Devices and Three-Site Configurations of “Architecting Availability and Disaster Recovery Solutions”.⁵ Once this has been established, the registration keys of the nodes that do not form part of the new cluster are removed by a node from the surviving portion of the cluster. Additionally, all of the disks that are shared between the two partitions are reserved for use solely by the surviving members of the cluster. This is achieved through SCSI reservation ioctls. It is at this point that the failfast driver on the nodes in the failed partition finds its access rights have been revoked and panics the nodes with a SCSI reservation conflict message.

The advantage of this approach is that the Oracle Clusterware framework does not get the revised cluster membership until the Solaris Cluster membership reconfiguration process has been completed. The only data that can be written to shared storage, by any of the nodes, is that which has previously acquired the appropriate Cache Fusion locks. Obviously, no additional locks can be acquired while the private interconnects are down. Further, by placing access controls at the storage driver rather than in user land processes, data integrity can be maintained more robustly.

Where the shared storage uses NAS, write I/Os are blocked by the NFS server removing the mount point share from the losing subset. Here, no failfast takes place because I/O is not passing through the sd or ssd drivers — it’s traversing the network stack.

Another capability that the Solaris Cluster framework has that differentiates it from Oracle Clusterware is the ability of a panicking node to inform the remaining cluster nodes of its demise without having them wait to detect it by the absence of heartbeat messages. This further allows the reconfiguration process to start earlier than it otherwise would have.

4. Designing Enterprise Solutions with Sun Cluster 3.0, Richard Elling and Tim Read, Prentice Hall, 2002, ISBN 0-13-008458-1 (page 86)

5. <http://www.sun.com/blueprints/0406/819-5783.pdf> Architecting Availability and Disaster Recovery Solutions, Tim Read, Sun BluePrints™ OnLine, April 2006

Device namespace management

The OCR and voting disk are two important components of Oracle Clusterware. The installation can be directed to use raw disk storage, a shared file system, or NAS for these components. Regardless of which of these options is used, the objects must be referenced by the same name from all cluster nodes. So if the first voting disk is a raw device named `/dev/rdisk/c5t1d0s1` on one node, it must be accessible via that name on all nodes. However, if the cluster does not consist entirely of nodes of the same type and configuration, this device may have a different name on one or more nodes. The only way to overcome this problem is to create and maintain a set of symbolic links to these devices (e.g., `/oracle/voting_disk1` linked to `/dev/rdisk/c5t1d0s1` on some machines and `/dev/rdisk/c4t1d0s1` on others). On a large cluster, this may be awkward and prone to errors.

Solaris Cluster software solves this problem by automatically creating and managing a global device namespace. The raw disk device used as an example above would have a single, consistent `/dev/did/rdisk/d10s1` name across the entire cluster. As new nodes are added to the cluster, their global namespace is automatically populated with the correct information and it remains consistent with that of the other nodes, regardless of the underlying controller numbering of the particular node.⁶

Private interconnect management

When Oracle Clusterware is configured, the `runInstaller` program asks which network interface cards (NICs) should be used for public and private networks. The private network is used by Oracle Clusterware to monitor the cluster membership (as described in section “The Oracle Clusterware membership framework”) and for Cache Fusion traffic. Ideally, the interconnect should be both highly available and trunked to allow it to use all the configured interfaces yet still survive the failure of a NIC.

Oracle Clusterware has no built-in high-availability capability for the cluster interconnect. A standard installation usually requires a single fixed interface or single IP address to use; for any additional capabilities it relies on the host operating system.⁷ If the interface or IP address is not made highly available, a failure will result in cluster nodes being aborted by the Oracle Clusterware framework.

The Solaris OS has had a built-in high-availability networking feature known as IP multipathing (IPMP) since the Solaris 8 release. Since the Solaris 1/06 update, link aggregation capabilities (IEEE 802.3ad) are also built into the Solaris OS, although a separate product known as Sun Trunking™ software has been available since 1997. These features can be used to address the Oracle Clusterware and RAC networking requirements, but each approach has some limitation when compared with the capabilities of the Solaris Cluster `clprivnet` feature.

6. <http://docs.sun.com/app/docs/doc/819-0421/6n2rmm7rk?a=view> Global namespace

7. <http://metalink.oracle.com/> Oracle MetaLink document 368464.1

If Oracle Clusterware is configured to use the IP address of an active/active IPMP, rather than a physical NIC, then high availability and outbound load spreading can be achieved, regardless of the network adapter or the number and type of Ethernet switches used with the proviso that node A always chooses the same active interface to send traffic to node B. However, the inbound packets will be received by a single interface. Further, if more than just link failure detection capabilities are needed on an interface, then IPMP test addresses and external, “ping-able” hosts must be configured. This allows IPMP to determine whether the adapter is capable of exchanging IP traffic with other nodes on the private network.

If link aggregation is used, full use can be made of all the bandwidth of the available network adapters and high availability of the aggregated link. However, all connections must be made through a single switch as there is no well-defined standard for doing 802.3ad link aggregation across more than one switch. As a result, this method introduces a single point of failure into the configuration. Using IPMP to circumvent this only results in a combination of the two shortcomings. Finally, there are currently limitations on which network drivers support link aggregation, further reducing the scope of this option. This restriction may be removed in future Solaris releases.

When the Solaris Cluster software is installed, it automatically creates a virtual network interface called `clprivnet0`. This virtual interface can be supplied directly to the Oracle Clusterware installation. The virtual interface provides transparent aggregation of all the underlying cluster private interconnect interfaces. If a path between cluster nodes fails (see section “Solaris Cluster software: robust membership and fencing”), that path is automatically removed from the trunk until such time as the path is found to be working again. If private interconnects are added or removed, this can be done online and `clprivnet0` automatically takes account of the change. Finally, the combination of `clprivnet` and the Solaris Cluster heartbeats provides an end-to-end test of the health of the communication link for potential Cache Fusion traffic. In contrast, an IPMP solution only tests the ability of a network adapter to communicate with a remote ping-able target and not necessarily the host that is the recipient of the application messages.

The `clprivnet0` feature supports all NICs that are qualified with Solaris Cluster software. This includes 100baseT, Gigabit Ethernet, 10 Gigabit Ethernet and Infiniband NICs as well as supporting jumbo frames. In addition, it does not require any special switch features — such as 802.3ad — and also supports connecting each path to a separate switch, which is actually the recommended configuration. In cases, where only two nodes are present, back-to-back Ethernet network connections are supported by Solaris Cluster.

Interconnect usage by Oracle RAC application traffic

Oracle RAC database instances use Cache Fusion to ensure data consistency for applications and on-disk data integrity. To achieve this, Cache Fusion has to communicate information about data locks held between nodes and, when required, pass data pages, too. Depending on how the database was created, the data pages may be between 2 KB and 32 KB in size. As the user load level increases, contention for important data pages grows. For the application to scale, it is important that the interconnect can handle the level of traffic by load balancing the requests across all the available NICs.

Oracle allows the configuration of multiple cluster interconnects (via the `cluster_interconnects` parameter) to support Cache Fusion traffic, but this comes at the cost of reducing availability if one database is configured to use multiple interfaces.⁸ If this option is chosen and an interconnect fails, it will result in instance evictions. Similarly, if one of the interfaces has failed, the database will not start.⁹ Furthermore, using the `cluster_interconnects` parameter means that database instances dependent on it must be taken down to make changes to its value. Alternatively, if instances inherit this setting from Oracle Clusterware, the value can be changed, post-installation, using the `oifcfg` command — but this necessitates a restart of Oracle Clusterware.

The section “Private interconnect management” describes how Solaris Cluster software creates a `clprivnet0` virtual interface for use by Oracle Clusterware and Cache Fusion traffic. This has the following load-balancing properties:

- TCP transmission is distributed on a per-connection basis. A given TCP connection is mapped to a physical path (i.e., the same path is selected for transmissions over that TCP connection). But if the physical path is detected to be faulty, the TCP connection is transparently failed over to one of the remaining healthy paths.
- UDP transmission is round-robin on a packet basis. Fragments for the same UDP datagram will go over the same link.

Therefore, using `clprivnet` simplifies Oracle configuration while ensuring high availability and maximum use of available interconnect bandwidth. Changes in the underlying make up of `clprivnet` are transparent to Oracle.

Oracle RAC data storage options

When a database administrator (DBA) deploys a standalone Oracle 10g database, the data can be placed on raw disk, on a file system (UFS, VxFS, QFS, or NFS), or under the control of the automatic storage management (ASM) facility. Despite raw disk generally offering the highest performance, many DBAs prefer using a file system to store their data because of its intuitive nature and ease of management. Database files can also be administered using standard Solaris commands such as `ls`, `cp`, `mv`, etc.

When Oracle Clusterware is used without Solaris Cluster software, the options for RAC data file storage on the Solaris OS are reduced to raw, ASM, or NFS only. While NFS is a usable option, the failure fencing available is limited to that provided by Oracle

8. http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14237/initparams025.htm Oracle® Database Reference 10g Release 2 (10.2)

9. <http://metalink.oracle.com/> Oracle MetaLink document 220970.1, Ref #: ID-4724

Clusterware's membership mechanism (see section "The Oracle Clusterware membership framework"). If ASM is used without Solaris Cluster, the DBA must manage the individual ASM initialization files (pfiles) held on each cluster node rather than one single, centralized server initialization file (spfile) unless NFS storage is available. The spfile file could be stored on a shared, raw device, but this would lack the protection of any mirroring and rely on a consistent device namespace that was managed by the administrator. (see Device namespace management)

Regardless of whether or not Solaris Cluster is used, ASM does not have any awareness of the health of the underlying storage it uses. Consequently, ASM will attempt to start even if one or more LUNs are unavailable.

Using Solaris Cluster software with Oracle Clusterware allows a DBA to use the high-performance shared QFS file system to store both Oracle Clusterware files, such as the OCR and voting disks and their RAC data, (e.g., tablespaces, online redo logs, control files, etc.). A matrix of supported options can be found in "Storage Management Requirements for Oracle Files" of Solaris Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS.¹⁰ Furthermore, the data integrity protection described in section "Solaris Cluster software: robust membership and fencing" includes data held on shared QFS file systems.

Extensive portfolio of application agents

With the increasing demand from IT departments to achieve higher levels of system utilization, it is rare to find systems running a single service. Consequently, systems often host a mixture of database, Web, and application services; and when these need to be highly available, they are clustered to ensure they meet agreed-to service levels.

Oracle Clusterware comes with a built-in framework for registering services, but its primary function is to support the Oracle Clusterware and RAC environment. Making a Web service, such as Apache or Sun WebServer software, highly available would require a user to write, test, and maintain the code themselves.

The Solaris Cluster software has its own agent development environment and comes with a simple Generic Data Service (GDS) to allow users to add support for their own services. However, this is in addition to a substantial portfolio of cluster agents that are written, tested, and supported by Sun. These include agents for Oracle (HA and RAC), SAP, NFS, Apache, and many others.¹¹

The emphasis here is that the burden of ownership, testing, and maintenance is on Sun rather than the user. Each of the agents undergoes rigorous testing, including stress tests under high system load to ensure continued correct operation. Moreover, the cost of the Solaris Cluster agents is a fraction of what it might cost a company to develop such sophisticated agents themselves.

10. Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS

11. http://1.http://www.sun.com/software/cluster/features_benefits.xml Sun Cluster agents list

Chapter 5

Availability and Scalability Versus Cost

System downtime has a cost through lost orders or reduced user productivity. Similarly, potential revenue may be lost when systems cannot be scaled quickly enough to meet a rise in customer demand. If these figures are large, prudent customers will want to take out the equivalent of an insurance policy to try and mitigate their effect. However, the cost of said policy and any “renewal fees” must be substantially lower than the cost of the event being insured against.

Although license agreements may vary from customer to customer, in general, Oracle RAC will be more expensive than the standalone (nonclustered) version of the product. However, the benefits it brings in reducing the risk of outages and increased system utilization often outweigh these costs. Solaris Cluster software also adds cost, but similar arguments can be used to justify the benefits it brings by further reducing the risk of outage and simplifying management.

Chapter 6

Summary

Datacenter architects and database administrators should recognize the many advantages of deploying the Solaris Cluster software in conjunction with Oracle Clusterware rather than using Oracle Clusterware alone. The additional costs of doing so are more than offset by the enhanced data integrity protection offered; the simplicity of namespace and interconnect handling; the choice for data file deployment; and the huge range of Sun written and supported data service agents available.

The latest Solaris Cluster 3.2 software release offers considerably greater integration for Oracle Clusterware and RAC, ensuring that the framework is brought up and shutdown in a coordinated fashion that recognizes the inherent dependencies that Oracle Clusterware has on the underlying storage components.

Acknowledgements

The authors would like to thank Eric Bezille, Honsing Cheng, Burt Clouse, Don Deal, Chris Dekanski, Nicolas Droux, Thorsten Frueauf, Prasanna Kunisetty, Martin Lorenz, Peter Memishian, Thejaswini Singarajipura, and Hartmut Streppel for their technical help in producing this white paper.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN (9786) **Web** sun.com



©2008 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Solaris, SPARCcluster, and Sun Trunking are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. Information subject to change without notice. SunWIN #495695 Lit. #SWWP14523-0 09/08