RAC FOR BEGINNERS: THE BASICS

Dan Norris, Piocon

Introduction and Terminology

This paper is designed to introduce Real Application Clusters (RAC) to beginner DBAs by providing understanding about how RAC works. After reviewing this information, you will not be an expert, nor will you have all the mechanics needed to install a RAC cluster. However, you will possess a conceptual understanding of the RAC architecture and the basic installation tasks needed to implement a cluster.

To support the discussion that follows, these definitions may be helpful:

Term	Definition
Database	the control files, data files, and online redo logs
Instance	the shared memory and background processes that operate on a database
Clusterware	software that manages cluster membership and monitors the nodes and networks in a cluster
Storage Area Network (SAN)	a storage environment where multiple servers can utilize a single storage array; a storage network is commonly implemented using fiber channel technology
Local Storage	disk space that is available to exactly one node in a cluster; this storage may be part of a SAN or may be direct-attached to the server
Shared Storage	disk space that is available to more than one node in a cluster at the same time; this storage is commonly part of a SAN
Raw Device	the character (unbuffered) special device presented by the operating system
Cluster Filesystem	a special filesystem that can be accessed by multiple cluster nodes at the same time
Oracle Automatic Storage Management (ASM)	software that will manage disks directly and provide volume management functions like striping and mirroring; this feature was new with Oracle Database 10g
Single-instance Database	the traditional "stand alone" database configuration used by most Oracle environments today
Multi-instance Database	a single database serviced by more than one instance
Oracle Services	an entity defined in RAC databases that allow you to group database workloads in order to route work to the instances best able to complete the work optimally

In the sections that follow, you will learn:

- RAC's product history
- An architectural overview of RAC
- A summary of the installation and configuration steps
- Tips for DBAs and managers regarding RAC
- RAC and packaged application implementation
- High availability alternatives to RAC

The Marketing Hype And The Real Skinny

Oracle Real Application Clusters has been steadily gaining momentum in the market with new and current customers considering RAC implementations. There is a lot of information available from Oracle and many other sources about RAC and its uses. Some purport that RAC is the answer for all database problems and that it should be used for every application in every situation. Other sources claim that RAC is somehow "broken" and that it will not function at all. Obviously, the truth is neither of these, but rather something in between.

Practical experience shows that there are certain applications that are not well-suited for RAC implementations. However, a vast majority of applications will function and perform very nicely on RAC clusters. In fact, many applications can utilize a RAC database with little or no modification.

A Brief History Of RAC

Oracle first had success with its parallel database option in RDBMS version 7.3 as Oracle Parallel Server (OPS). The OPS product had some severe limitations compared to today's RAC environment. OPS relied heavily on the cluster vendor to provide cluster management and high-speed, low-latency interconnect technology to support it. The OPS product was enhanced with database versions 8 and 8i.

With the 9i release, Oracle introduced its Cache Fusion technology and renamed the product to Real Application Clusters. Oracle also introduced its own clusterware for Windows and Linux platforms. Oracle claimed to have Cache Fusion in development for 10 years before its launch, which gives an idea of how complex the RAC software stack is. Cache Fusion had a huge impact on the product and was the primary technology that allows many more applications to use RAC databases with little or no modification to the application.

What RAC Is NOT

While some claim that RAC should be used in every application and all environments, there are cases where decision makers should not consider deploying RAC.

- RAC is not the best choice for all situations. While RAC does provide very high availability, not all situations require the highest availability. For situations where business requirements can be met by a failover cluster, RAC may be unnecessary. RAC may also require additional skills and additional licensing which require additional expenditure.
- RAC is not "production only" technology. If an organization fails to include a non-production RAC cluster in their environment, they should not use RAC for a production environment. RAC does have an impact on the application, and it sometimes affects the functionality required by the application. So, a non-production environment is necessary to ensure that upgrades, patches, and configuration changes can be tested in a similar environment before deployment.
- RAC is not a technology you should expect to learn by doing. While RAC clusters have become easier to install and manage with each release, staff must be willing to dedicate themselves to learning this new technology. For some, reading documentation, viewing online webcasts, and taking online training may be sufficient. For others, instructor-led classroom training may be the best option. Bottom line: if the RAC implementation plan does not include training in some form, the implementation will likely fail.
- RAC is not a "set it and forget it" environment. Some customers lack a full-time DBA on staff to maintain and care for their small Oracle database environment. If an organization has the right partner, RAC can be managed remotely, but it does require more administrative attention than a standalone, single-instance database. Sites that do not have their own DBA expertise in-house and lack remote support arrangements are likely to quickly find that their RAC implementation falls short of expectations.
- RAC is not a transparent change for all applications. Large enterprise applications can utilize a RAC database, but comprehensive testing should be performed by the application vendor. If the application vendor has no support for RAC, the implementation is not likely to succeed. Due to the complexity of enterprise applications and inability to change some of the internal SQL and behavior, it is difficult to successfully implement RAC for these large applications without vendor support.

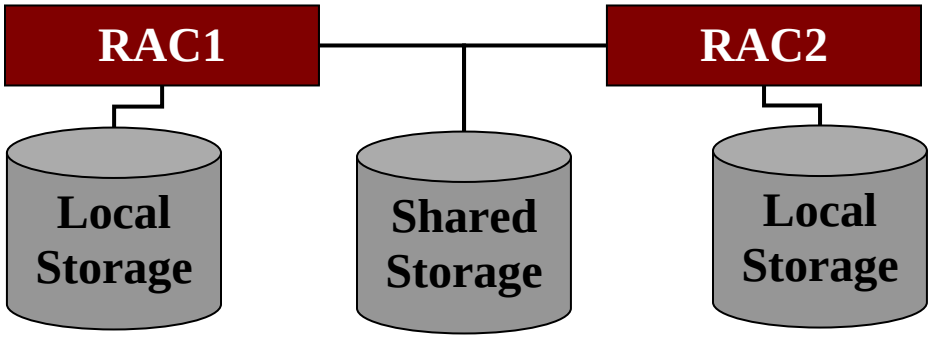
These are a few of the situations where RAC may not be the best choice. Also be on the lookout for the "instant proponent" syndrome. This is my term for an individual's condition following a seminar or other semi-technical session in which all the benefits are stressed and limitations are minimized or not mentioned.

While RAC is a strong technology that has a bright future and solid road map, it is always best to know all the facts before making a decision regarding technology.

What Is Different About RAC Vs Single-Instance

The obvious differences between RAC and a traditional single-instance database are apparent on several different levels. The sections that follow describe these differences in more detail.

System Architecture



The diagram above shows a simple RAC cluster architecture. A minimal cluster consists of multiple servers, a private network, a public network, and shared storage. Most DBAs are familiar with the single-instance configuration in which storage is not shared between nodes and all access to the database comes from the single instance.

Sharing the same storage poses several challenges. First, the operating system must be able to handle accesses from other nodes at the same time as it accesses the storage. In the early days of Oracle Parallel Server, this was addressed by using raw devices. The operating systems were accustomed to file systems that supported access from exactly one host. When only one host was involved, the host could manage all locking and synchronization from a single operating system kernel. With multiple hosts, the filesystem access must be coordinated between all hosts sharing the filesystem. This introduces some latency and, until recently, cluster-wide filesystems were very uncommon. Today, several vendors offer cluster filesystems that are supported by Oracle RAC. Oracle also offers its OCFS filesystem for Linux and Windows with reported plans to make it available for more platforms in the future.

Of course, sharing a database also creates challenges. Being able to handle the cross-host locking required to ensure database integrity is the first and most important challenge. Previously, that challenge was addressed by exchanging blocks that required serialization via disk. This was known as a block ping and was a major limiting factor for OPS. With Cache Fusion technology introduced in Oracle9i, the block ping was put on the endangered species list and was only required for a small, infrequently seen set of circumstances.

The additional hardware required for a RAC cluster usually consists of a few additional networks to support the private interconnect. The private interconnect can be made up of multiple physical network interfaces. It is common to utilize two or more gigabit Ethernet interfaces for the interconnect. In past versions, expensive and proprietary interconnect technologies were used. These proprietary interconnects were required because gigabit Ethernet created too much latency and did not have the bandwidth necessary to support some environments. With advancements in Ethernet technology and the ability to utilize multiple interconnects, gigabit Ethernet is the most common interconnect in use today. Infiniband is a relatively new technology that was recently certified for use as an Oracle private interconnect. Infiniband promises to be a significant player in the RAC environment because of its impressive technical specifications.

Prerequisites

As OPS matured into the RAC product and RAC continues to improve, the prerequisites for RAC have been significantly reduced. Additionally, most of the software requirements are now included as part of the Oracle software stack. The biggest change came with Oracle Database 10g when Oracle Clusterware was introduced. With Oracle9i, a clusterware stack was offered for Windows and Linux platforms. Oracle Clusterware, included as part of the 10g technology stack, is available on every platform where RAC is available.

Other than software, the hardware investment needed to create a RAC cluster is usually only a few networks to support the private interconnect. Shared storage is also required, but that is generally available by reconfiguring an existing storage device.

With Oracle Clusterware, each node will require a virtual IP address (VIP) in addition to the node's own IP address. The VIP will be managed by Oracle Clusterware and is used to enable faster reconnects when a node fails. The installation process will prompt you for these additional VIP addresses, so don't configure them on the node prior to performing the installation.

The installation guide and Metalink notes will help you configure host equivalency via secure shell (SSH) or other means. Host equivalency is configuring a host or individual accounts to allow them to login to a remote host without prompting for credentials.

Database Configuration

Shared and Unshared Database Files

Some database components can be used by only one instance. Online redo logs, undo tablespaces, and rollback segments (if not using automatic undo management) are the first items you'll need to configure for each instance. Each online redo log group is part of exactly one thread of redo logs. Each thread of redo logs (which will include at least two log groups) is dedicated to a single instance. In most configurations, the thread number for each instance is hard-coded as an initialization parameter (thread). Hard-coding the thread number avoids confusion and ensures reliable thread selection for each instance.

Most databases today will probably use automatic undo management. In those cases, a separate undo tablespace will need to be maintained for each instance, and the name of that tablespace will need to be specified in the server-specific unto_tablsepace parameter for each instance.

It is important to note that when an instance fails and at least one instance survives, a surviving instance must perform instance recovery on behalf of the failed instance. Instance recovery in a RAC cluster involves playing back all transactions since the last checkpoint from the online redo logs of the failed instance. Only one instance will write to any given thread of online redo logs, but any instance may read other threads in order to perform instance recovery in the event of an instance crash. With a single-instance database, instance recovery is performed when the instance is started again.

Locally-managed Tablespaces (LMT) and Automatic Segment Space Management (ASSM)

While I would recommend using locally-managed tablespaces (LMT) and automatic segment space management (ASSM) for all databases where those features are available, LMT & ASSM play particularly important roles in RAC environments. ASSM allows Oracle to manage freelists and freelist groups which can be very helpful in high-transaction-rate environments. There are many sources for detailed information and discussion on LMT and ASSM. Metalink, AskTom.oracle.com, and Oracle mailing lists all have archived articles and discussion on these topics if you would like more information.

Server Parameter File

While some sites have been slow to adopt server parameter files as part of their usual configuration, RAC environments make better use of the server parameter file (spfile) features than single-instance environments. In a spfile, there are global parameters and instance-specific parameters. For a RAC cluster, the global parameters make it much easier to ensure that all the parameters that must match on all instances stay in sync.

With a single-instance cluster, all parameters are global by default since each spfile is used by exactly one instance. With RAC, it is typical and recommended that a single spfile be used in the cluster with all instances sharing it directly. This means that the spfile in a RAC cluster must be placed on shared storage. It can use a raw device or be placed on a filesystem in order to fit all environments.

In most environments, most parameters are global parameters. Some parameters must be the same on all instances, like control_files, db_name, db_domain, and cluster_database. Other parameters like instance_number, undo_tablespace, and thread must be different on each instance.

Server parameter files are maintained using ALTER SYSTEM from one of the instances.

Database Access Considerations

For a multiple-instance database, you have choices about how to manage connectivity to your cluster. In general, you can choose to load-balance users across all instances in the cluster or segment workload across the cluster, putting a subset of the user community on each instance. There is also an option for two-node clusters to create an active-passive configuration such that all users operate on a single node and are switched to the second node in the event of a failure. With a combination of server-side configuration and client settings, you can determine which connectivity method you wish to use.

The advantages of a load-balancing configuration are that you get even workload distribution over time so you are using all servers equally. Additionally, using load balancing will mean that you have less work to do when adding or removing nodes from the cluster because your connectivity method does not depend on any particular node or a particular number of cluster nodes. Load balancing for connections is handled entire by Oracle's TNS networking software, so no additional network devices are needed. It is important to keep in mind that in Oracle9i and 10g Release 1, load balancing only occurred at connect time. The primary disadvantage when using load balancing has been that it only happens at connect time. Starting with Oracle Database 10g Release 2, Oracle has added the load balancing advisory feature that can help applications determine when relocation may be necessary due to increased workload on a specific node.

Most implementations use load balancing to distribute connections to all cluster members.

Licensing

Oracle changes their licensing scheme and pricing structure frequently, so be sure to check the current licensing offers.

Oracle offers RAC through two different database editions. First, Oracle Standard Edition (not to be confused with Oracle Standard Edition One) includes the RAC option at no additional charge. The Standard Edition restriction is that a RAC cluster cannot contain more than 4 processors. If you wish to scale beyond 4 processor cores in your cluster, you must purchase an Enterprise Edition license and the RAC option for Enterprise Edition. Both editions can be purchased with two licensing methods: named user or processor-based. Your Oracle account representative can help determine which option is best for your environment.

Installing A RAC Environment

There are two major tasks when installing a new RAC environment. The first is installing Oracle Clusterware on all cluster nodes.

Install Oracle Clusterware

Oracle Clusterware installation is facilitated by a much-improved installer that will install and configure Oracle Clusterware on each node in a single installer session. The details are also covered very clearly in the installation guide, so we won't reprint the details here. There is a separate installation guide for each platform that RAC supports.

Note that there are a few prerequisites that need to be in place before starting the installation. The Cluster Verification Utility (CVU) is available on the installation media and will help determine if you have correctly configured all the prerequisites. Chapter 2 of the installation guide shows how to use CVU. The output from CVU includes messages that are easy to understand and will describe what tests failed, if any. The installation will prompt for the VIP address and corresponding hostnames, so be sure to have that information handy.

Install Oracle ASM

Oracle Automatic Storage Management (ASM) can be installed in a separate ORACLE_HOME location. Even in single-instance configurations, it is recommended to install ASM in a separate ORACLE_HOME location instead of allowing it to share the database ORACLE_HOME. This deployment method allows for more flexibility and reduced downtime when upgrading and patching all different installations.

One of the earliest screens in the ASM installation process allows you to choose onto which nodes the software should be installed. This is one of the best indicators of a functional cluster. If your installer does not present the cluster node selection screen, you should stop, investigate the installer's logfiles and interrogate the cluster services to ensure that Oracle Clusterware is running and available.

Installing ASM is not required (at least not yet), but it is recommended. In a future release, Oracle may require ASM to be used for some or all database files. It can provide superior performance and reduce costs by eliminating the need for a volume manager.

Install Oracle Database

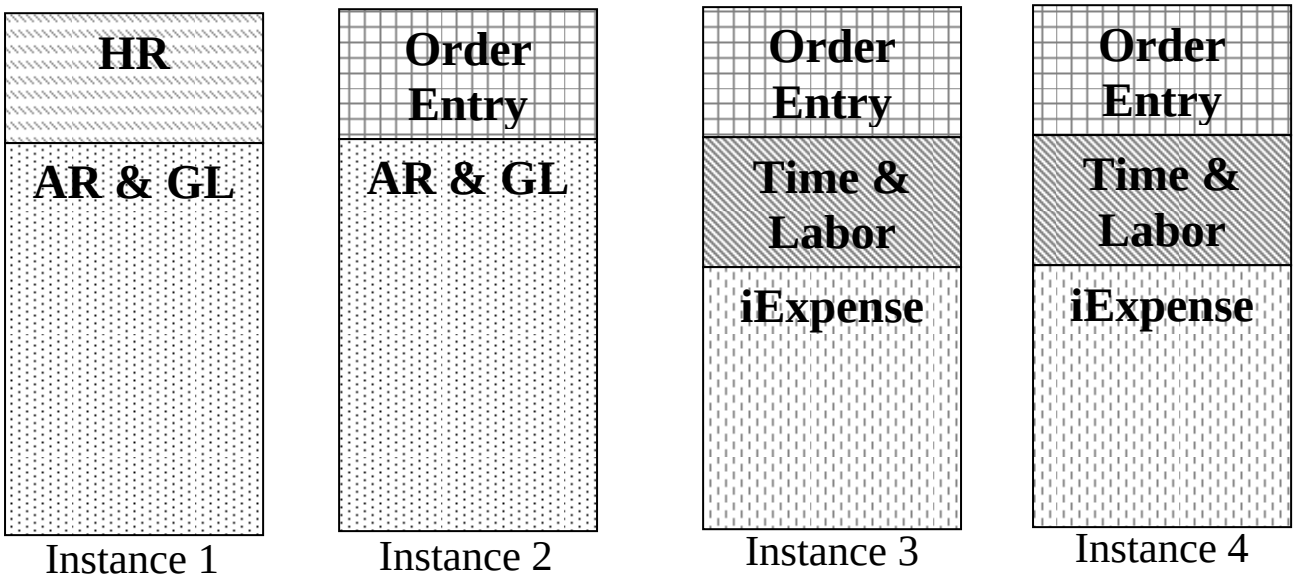
Once the Oracle Clusterware installation is complete, you can start the Oracle Database software installation. As with the ASM installation, the database installation should present a screen asking on which nodes the software should be installed. Another option presented during the installation will be whether or not you would like to create a database during the installation. I have found it to be a good practice to do a software-only installation initially and create a database after the

installation is complete. Following an initial installation, it is often necessary to install patches. Since most patches have required post-installation steps for the database, it is easier to create a database later and avoid the post-installation steps. Plus, creating a database after installing all patches confirms and tests the installation.

Oracle RAC Services & Workload Management

One area that most DBAs new to RAC struggle with is designing and utilizing Oracle services appropriately. Granted, using services is not required, but strongly recommended.

An Oracle Service is technically a service name that is registered by one or more instances in the cluster. It is common for a cluster to have multiple services. Normally, a service will be related to a particular application (if multiple applications share the database) or to a business function in the case of a larger application. Utilizing services is one way to manage workload in the cluster. For example, if you manage a cluster with 8 nodes, you may want to allow 6 nodes to serve the accounts receivable users, but only 4 nodes to service HR users.



In the diagram above, each color represents a service being offered in the cluster. As you can see, Order Entry is available on three instances while iExpense is only available on two instances. In the event of an instance failure, the services would be redistributed according to policies configured for each service. You can create, delete, relocate and manage services directly via Oracle Enterprise Manager. Services are also created as part of database creation when created with the database configuration assistant.

We should note here that starting with 10g, instance statistics are gathered on a per-service basis as well as the other levels. That means that you can report the number of logical I/O operations performed by a service as well as by an instance, user, or against a particular object. This is a very useful tool for identifying the biggest consumers on the cluster. These statistics can be viewed instance-wide using GV$ views.

New Features Overview (And Some Relevant History)

Some of the biggest changes for RAC environments have not been the database changes, but rather the cluster software changes. Oracle has slowly but surely taken ownership of the cluster software stack. In the pre-9i versions of Oracle Parallel Server (OPS), customers were required to obtain clusterware from other software vendors. Many times, customers would get this from their operating system vendor. Some platforms also required that the clusterware vendor also provide a distributed lock manager (DLM).

The main disadvantage to this approach was that Oracle was required to integrate with many different clusterware vendors in order to support all the platforms. One advantage was that the clusterware vendors were typically very good at exploiting the hardware in order to make a fast, reliable interconnect because many times, the clusterware vendor was also the hardware vendor.

Starting with Oracle9i, Oracle provided a complete clusterware solution for Linux and Windows platforms. Around that same time, Oracle also licensed the clusterware technology used by Digital's Tru64 UNIX TruCluster product and began working to port it to all the Oracle-supported RAC platforms. When Oracle Database 10g launched, it included the complete clusterware stack for all supported platforms. This simplified the installation immensely and made the product much more supportable from Oracle's point of view, but customers also benefited from the ability to obtain peer-to-peer support from other RAC users using different platforms (since all platforms were functionally the same).

Tuning A RAC Environment

Tuning a RAC environment starts with the same single-instance tuning most DBAs conduct all the time. Once single-instance tuning has stopped providing benefits, investigating RAC-specific tuning may provide additional benefits. For most environments, RAC-specific tuning includes examination and investigation of cluster interconnect activity. For experienced single-instance DBAs, this methodology will leverage their experience and minimize changes to common practices.

There are a few new things to learn in order to become familiar with RAC-specific tuning issues. For example, for those DBAs accustomed to using statspack reports to assist with tuning, a new RAC section will appear in the statspack reports created from a RAC database. Additionally, statspack information is instance-level information, so statspack snapshots will need to be gathered for each instance in the cluster separately and reports generated for each instance as well. The new RAC section in statspack reports will contain interconnect latency and utilization information. This information will help identify issues related to the RAC interconnect.

Of course, Oracle Enterprise Manager is also "RAC-aware" and can be used to help with tuning a RAC database and cluster. Current metrics related to cluster interconnect can be reviewed and reports can be generated to look at historical information as well.

You can find more specific information on RAC tuning in the server documentation for RAC.

RAC Backup and Recovery

Backup and recovery for RAC environments is complicated by the fact that each instance generates its own thread of archived redo logs. In order to perform a recovery, all redo generated by all instances will be needed. Depending on the configuration, recovery can be tricky since media recovery is performed from only one instance and that instance must have access to all the archived redo logs from all instances.

With a cluster filesystem, this is easier to manage since all archived redo logs from all instances can be written to the same location. That makes it easier to recover since all the logs are in the same directory.

The mechanics of recovery and syntax for recovery commands are very similar to single-instance recovery. RMAN is also a RAC-aware tool and can handle media recovery with multiple threads of redo.

Recommendations For Dbas New To RAC

If you're just getting started with RAC, here are some recommendations that have served me well as I continue to learn about RAC (in no particular order):

- Read and understand the concepts guide from the database documentation. Understanding concepts is key to growing your knowledge. If you have a conceptual understanding of what you're dealing with, you're more likely to know where to look in a troubleshooting or tuning situation.
- Review Metalink notes and online forums for information about key issues. Metalink is home to several how-to articles on RAC installation and configuration.
- Review OTN articles. OTN has many articles on RAC implementation, especially on the Linux platform. While many articles also include Linux setup information, the RAC installation and configuration information is generally applicable to all platforms, especially with 10g since the clusterware is the same on all platforms.
- Take advantage of the RAC Special Interest Group (RAC SIG). This group operates a website at www.oracleracsig.org and offers free live webcasts once or twice a month. See the site for the webcast schedule. The group also organizes RAC sessions and events at major conferences. The website includes a repository of RAC-related whitepapers, presentations, and an archive of all past webcasts, which is available for playback on demand.

Recommendations For Managers Considering RAC

If your organization is considering RAC, here are some considerations to keep in mind and references to useful information. All considerations must include discussion about technical advantages and disadvantages as well as an examination of alternative technologies that may meet the business requirements.

- RAC does require additional DBA expertise in order to manage it properly. Many RAC implementations are perceived as unsuccessful due to lack of training and failure to respect the complexities of the RAC environment.
- While RAC's benefits are unique and unlike many other possible solutions, a RAC environment can sometimes be more difficult to troubleshoot than a single-instance database environment. It is important to identify and measure the long-term benefits of the solution.
- Grid management is real, but a grid cannot be created overnight.
- Oracle has an entire section on their website devoted to business analysis of the RAC technology. Of course, you won't find much information about the disadvantages or challenges of a RAC environment, but there is a lot of good information on the site at http://www.oracle.com/database/rac_home.html.

RAC And Vendor-Provided Applications

The purpose of this section is to make you aware that most application vendors do specific testing on RAC to certify it with their software packages. If a software package does certify some Oracle Database versions, they most likely have a separate certification list for RAC versions. Be sure to ask software vendors if they specifically test RAC as part of their certifications. If not, you might request that they do specifically test and certify RAC as it may open them up to additional opportunities for their applications.

Currently, Oracle E-Business Suite, PeopleSoft, Siebel and SAP are some of the vendors that specifically certify RAC as a database platform.

Alternatives To RAC For High Availability

The traditional Oracle Database high availability solution has been a failover cluster. A failover cluster operates by linking two or more systems together with cluster management software to detect failures and coordinate activities. The systems in the cluster all have access to some common storage even though traditionally, only one server uses the shared storage at any given time. This allows for a configuration where a single-instance database can be transferred between cluster hosts with little effort.

The benefits of failover clusters are lower complexity since you are still managing single-instance databases and you can also save some Oracle licensing costs in many cases. On the other hand, failovers are typically on the order of a minute to several minutes depending on the database configuration and size. For RAC databases, failover from a failed instance to a working instance is usually only a few seconds.

References

- Oracle US Commercial Price List dated January 6, 2006, http://www.oracle.com/corporate/pricing/pricelists.html
- Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide 10g Release 2 (10.2), Part Number B14197-02

From the Lawyers