



Why Not Oracle Standard Edition?

A Dbvisit White Paper

By Anton Els

Executive Summary.....	3
Target Audience.....	3
Introduction.....	3
Key Differences Between Database Editions.....	5
License Cost Comparison.....	6
Example A: Oracle Licensing using Single Instance Database Environment.....	7
Example B: Oracle Licensing using RAC and DR Database Environment.....	8
Standard Edition and High Availability.....	10
Standard Edition and Disaster Recovery.....	11
Test Case: Oracle RAC With Standby Database using Standard Edition.....	13
Is Two and Four Sockets Really a Limitation?.....	16
Reasons to Consider using Standard Edition.....	17
Conclusion.....	17
Appendix A: Oracle Processor Core Factor Table Extract.....	19
Appendix B: Flashback Query.....	21
References.....	23

Executive Summary

Why not look at Standard Edition? Sometimes we tend to just forget about it and we focus only on the top of the range products with all the latest and greatest features. Yet, on completion of a new solution implementation you realize that you did not use any of these features.

The objective of this paper is to assist you in:

- Gaining an understanding of some of the core features available in Oracle Database Standard Edition and how these can be used to provide you with a robust environment at a lower cost.
- Understanding how Oracle Real Application Clusters (RAC) can be implemented with Standard Edition, and what the possibilities and restrictions are.
- Learn key points for implementing a Highly Available solution using Oracle Database Standard Edition

Target Audience

Whether you are a director, manager or just the junior database administrator in the team; understanding what the capabilities of Oracle Database Standard Edition are, is valuable and can assist you in making the correct decisions when implementing, designing or budgeting for new solutions.

Introduction

Oracle Database Standard Edition is sometimes the neglected smaller brother of Enterprise Edition. Most will focus on the new features available in the Enterprise Edition stack, but what about Standard Edition? People seem to forget that it is still the same core database. It can provide you with highly available solutions as well as help you establish a robust Disaster Recovery (DR) environment. It is even possible to use Oracle Real Application Clusters (RAC) with Standard Edition, and when combined with a standby database, you can implement a powerful solution that will suit most environments. To top this all off, you gain the added advantage of lower licensing cost. Whether you are a small business or enterprise, Standard Edition may well suit your needs and as the cost saving can be substantial it is worth consideration.

This presentation will take you on a journey into some of the options available with Oracle Standard Edition - options which can provide you with a highly available solution with disaster recovery in place.

Oracle Database Editions

Before we focus on some of the key Standard Edition features, it is good to get an understanding of the key differences between the different Oracle Database Editions. Oracle provides five database Editions. These are:

- Oracle Database Express Edition (XE)
- Oracle Database Personal Edition
- Oracle Database Standard Edition One (SE1)
- Oracle Database Standard Edition (SE)
- Oracle Database Enterprise Edition (EE)

If we look at these Editions we tend to classify them by company size. We will state that the Standard Edition One is for small companies, with small systems and then move up with Standard Edition being for the small to medium sized companies or departments of large companies. Enterprise Edition is seen as the big brother, for all the enterprise companies and large systems out there. In some sense this is true. But there is no size limit on the Standard Edition One database, so why classify it as for small companies. But the reason for this 'classification' is not just the size, the feature limitations and most of all the licensing costs play a roll in our perception that Standard Edition is only for the smaller companies or small databases.

What are these feature limitations and is it really a limitation. To understand this better it is recommended that you do a detailed review of the database Editions available to make sure the features your application or business require is available.

In this paper I will not focus on Oracle Database Express Edition or Personal Edition, but I will provide a summary of these two:

- **Express Edition**

The Express Edition is seen as the entry-level database edition, but that does not mean it is not worth looking at. It is based on the same core technology used for the Enterprise Edition and the database we all came to love (or hate sometimes). It is Free to download and use, quick and easy to install and maintain. There is no limitation on the hardware it can be installed on, but it is limited to only use one CPU, 1GB Memory and 4GB of user data. The size limitation of the database is key, but taking into account the technology at your fingertips, this is truly a powerful little database that can be used for small applications not needing a lot of storage. It is ideal for web based applications.

- **Personal Edition**

The Personal Edition is probably one of the least used editions. This is a powerful edition that includes all the features and options that are available in the Enterprise Edition, except the Real Application Clusters (RAC) option as well as the management packs are not included. The first key restriction on this edition is it is only available for Single-User license, such as development or deployment environments that require similar features as what is available in Enterprise Edition. The second limitation is that it is only available on Windows and Linux platforms.

- **Standard Edition One**

Standard Edition One is the same as Standard Edition. It is important to note that this is still the same core database technology as what is available and used in Enterprise Edition, but with specific features and options removed. There are two key restrictions on Standard Edition compared to Standard Edition:

- Limited to two CPU sockets (maximum capacity per server)
- Oracle Real Application Clusters (RAC) option is not included

- **Standard Edition**

Standard Edition can be licensed and used on servers with a maximum capacity of 4 CPU sockets. One of the most important factors of Standard Edition is that it comes bundled with the Oracle Real Application Cluster (RAC) option since 10g. This option provides business to implement higher available solutions using Oracle's proven RAC technology at a lower cost. The restrictions on using Oracle RAC with standard edition are that you can only have a maximum of 4 CPU sockets in the total cluster configuration and that you have to make use of Oracle Clusterware and ASM Storage for the database.

• Enterprise Edition

Enterprise Edition is Oracle’s flagship database edition without CPU limitations and includes features that provide performance, availability, scalability and security. It should be noted that some of the optional features require additional licenses (at additional costs). This includes but is not limited to:

- Oracle RAC
- Partitioning
- Performance Tuning Pack
- Diagnostics Pack

Key Differences Between Database Editions

Some of the key differences between the three primary database editions provided by Oracle are highlighted in the table below. Note that this is only an extract of some key differences of which some will be discussed in more detail in this document.

Database Features ¹	Standard Edition One	Standard Edition	Enterprise Edition
Maximum CPU’s	2 CPU Sockets	4 CPU Sockets	No Limit
Oracle Real Application Clusters (RAC)	X	✓ (Included with SE up to Max. Total of 4 Sockets in Cluster)	✓ (Extra License Cost Option)
Oracle Data Guard	X (3rd party options available)	X (3rd party options available)	✓ (Active Data Guard requires additional license)
Flashback (Table, Database, Transaction)	X	X	✓
Parallel Options (Parallel Query, Stats Gathering, Index Building and Parallel Data Pump)	X	X	✓
Automatic Workload Repository	X (Statspack and 3rd party options are available)	X (Statspack and 3rd party options are available)	✓ (Extra License Cost Option)
Recovery Manager (RMAN)	✓	✓	✓
Some key options not available in SE1/SE: <ul style="list-style-type: none"> - Parallel Backups - Fast incremental backups with Block Change Tracking - Block-level media recovery 			

Reviewing the table above, it is clear that the big brother of them all is Enterprise Edition with all options available, although with some being available at an additional cost. But it is important to know that Standard

Edition, even though it does not have all the features available compared to Enterprise Edition, is still capable of providing a highly available solution with disaster recovery. One of the core reasons for this is that Oracle RAC is included with Standard Edition at no additional cost. This is truly a game changer as smaller to medium sized companies can now afford to look at using this amazing technology to help them achieve their business requirements. The second component to highlight is Standby Databases. Even though Standard Edition does not include Data Guard, it is still capable of providing standby database functionality. The next section will focus on the different licensing options available and the possible cost savings when using Standard Edition.

License Cost Comparison

Now that we have a bit of an understanding on the differences between the versions it is always a good to get an understanding of licensing cost. To illustrate this two examples will be used. One example will show you the licensing details for a Single instance (single server) environment with two quad core Intel Xeon CPU sockets. The second scenario will do a comparison of using Oracle RAC implementation using two nodes and to add a little more complexity to it, we will add a single instance standby environment. But before doing this comparison, let's review the Oracle Licensing structure.

Oracle Database Licensing makes use of two metrics:

- **Named User Plus**

When looking at Named User Plus (NUP), the key is you can identify and count the users that will be making use of the database. One of the key factors to take into account is that these include both human and non-human devices. All of these counted users should be licensed. Example, if you have 20 users that can use the database either via their own PC or just a shared device, all 20 needs to be licensed. If you have non-human operated devices, they will require a license as well. However it should be noted that a licensed user (NUP) could access multiple instances with this license as long as the "minimums" are met for each server.

- **Processor**

If you cannot identify and count your users, such as Internet facing environments, then you need to look at the Processor License metric. All processors in the server where Oracle is running should be licensed. The Processor License can logically be split into two sections:

- **Processor License Standard Edition**

With Standard Edition, the licensing is socket based. Standard Edition does not take into account the processor cores, but only the physical CPU socket. Example, you can have a Standard Edition One license (which is up to a maximum of 2 CPU sockets) using two CPU sockets, but each socket containing a CPU with 4 Cores. With Standard Edition, you are limited to a maximum of 4 CPU sockets in total, meaning if it is a cluster, you can only have 4 CPU sockets in use in the cluster. Note: if multi chip modules are used, each chip counts as a socket.

- **Processor License Enterprise Edition**

With Enterprise Edition the processor license is a little bit more complex. The first thing you need to be aware of is that it is taking CPU Cores into account. Oracle is using a "Processor Core Factor Table" – See Appendix A, in which they assign a value to each specific CPU. This value should then be multiplied by the total amount of Cores on the CPU socket to identify the required Processor License.

Example, for an Intel E5-2609 Xeon CPU with 4 Cores, a Factor of 0.5 is used. This means that you will require $4 \times 0.5 = 2$ Oracle Processor Licenses.

- **Minimums**

It is important to take note of the minimums. When purchasing licensing you need to take into account the minimum amount of licenses you will need to acquire. These minimums are especially important when buying Named User Plus licensing.

- **Standard Edition and Standard Edition One**

The minimum for Standard Edition and Standard Edition One is 5 Named user Plus (NUP) licenses for the whole company.


- **Enterprise Edition**

For Enterprise Edition, the minimum is 25 Named User Plus licenses per installed Processor where installed processor is equal to the value of processors being calculated taking into account the core factor. For example, if you have a 4 Core Processor with a 0.5 factor, you require 2 (4×0.5) Oracle Processor licenses. Taking the minimum of 25 Named User Plus into account, you will require 50 (2×25) Named User Plus licenses.

The next section will now provide you with two License examples to illustrate the lower cost of Standard Edition.

Example A: Oracle Licensing using Single Instance Database Environment

In the first example, we will be taking a Single Server with 2 CPU Sockets using the Intel Xeon CPU with 4 Cores. For licensing purpose memory and storage is not taking into account. It is also assumed the environment will be web facing and Processor licensing will be required as there is no way to specify or identify the exact users. Note that for the Intel Xeon E5-4603 Quad Core CPU the processor factor is 0.5 when using Enterprise Edition licensing. Pricing below is in US\$ and only include the first year support costs.



Processor: 2 x Intel Xeon
E5-4603 (4 Cores/CPU)

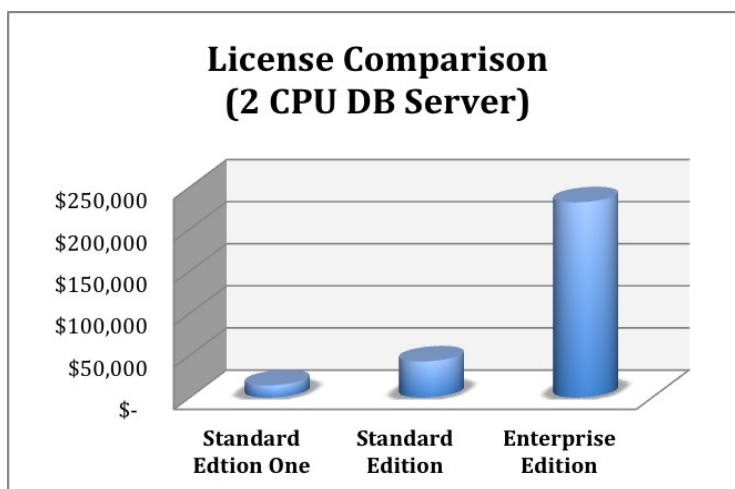
Memory: 64G

Storage: Internal 1TB

Standard Edition One	Standard Edition	Enterprise Edition
2 Processor License - (2 Sockets)	2 Processor License - (2 Sockets)	4 Processor License - $((2 \times 4) \times 0.5) = 4$

\$ 5,800 x 2 = \$ 11,600	\$ 17,500 x 2 = \$ 35,000	\$ 47,500 x 4 = \$ 190,000
First Year Support: \$ 1,276 x 2 = \$ 2,552	First Year Support: \$ 3,850 x 2 = \$ 7,700	First Year Support: \$ 10,450 x 4 = \$ 41,800
Total Cost \$ 14,152	Total Cost \$ 42,700	Total Cost \$ 231,800

Note: Pricing used was extracted from <https://shop.oracle.com> as at 22 Feb 2013. Note that pricing does not include any discounts.



Example A Summary

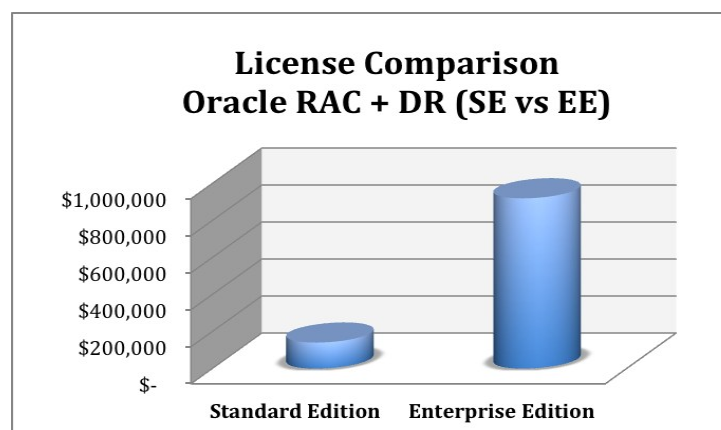
As you can see there is a substantial difference in upfront cost as well as ongoing support between the different editions. In this example Standard Edition One is 6.1% of the Enterprise Edition licensing cost. It is important to state that the pricing above does not include any special discounts and that pricing is purely taken as it is stated on the Oracle website.

Example B: Oracle Licensing using RAC and DR Database Environment

In this test case we will be looking at two scenarios. The 1st is using Oracle Enterprise Edition for the cluster and DR configuration using Oracle RAC and Data Guard. The 2nd option will be implementing the same solution, but with Standard Edition using Oracle RAC and 3rd Party product for the DR (Standby Configuration). The environment consists out of a Primary site running a Two Node Oracle RAC cluster. Each node containing 2 x Intel Xeon E5-4603 processors. The Standby site consists out of a Single instance environment running on one server using 2 x Intel Xeon E5-4603 Processors. Note that pricing is in US\$ and only includes the first year of support. The 3rd Party product used to implement the DR solution using Standard Edition is Dbvisit Standby.



Standard Edition	Enterprise Edition
6 Processor License - (2 Sockets per server - 3 Servers)	12 Processor License - (24x0.5)=12 (8 Core/server - 3 Servers)
Oracle Database Licensing: \$ 17,500 x 6 = \$ 105,000	Oracle Database Licensing: \$ 47,500 x 12 = \$ 570,000
First Year Support: \$ 3,850 x 6 = \$ 23,100	First Year Support: \$ 10,450 x 12 = \$ 125,400
Oracle Real Application Cluster Licensing: None	Oracle Real Application Cluster Licensing: \$ 23,000 x 8 = \$ 184,000
3 rd Party Product to provide/maintain Standby Database:	Oracle RAC First Year Support: \$ 5,060 x 8 = \$ 40,480
Dbvisit Standby: 6 Processor Multi Instance = \$ 11,893	Standby Database using Data Guard (included with EE)
Total Cost \$ 139,993	Total Cost \$ 919,880



Example B Summary

Similar to Example A, in this example we can clearly see a big difference in the cost of using Standard Edition vs. Enterprise Edition. Key to this cost difference is the fact that with Standard Edition Oracle RAC is free of charge. When implementing Disaster Recovery capability, Oracle Enterprise Edition does have Data Guard at your disposal and this is truly a powerful feature especially if you combine it with “Active Data Guard”, but again this option will be at an additional cost. To provide Disaster Recovery capability such as a Standby Database, Standard Edition does require 3rd party products or custom scripts to be implemented. But even with these additional included, the cost is still substantially lower than the Enterprise Edition Solution.

Standard Edition and High Availability

When thinking about high availability, there are a few options that come to mind. Some of the most common ones are listed below. But how do they fit in with Standard Edition?

- **Oracle Real Application Clusters (RAC)**

Oracle RAC is available for use with Standard Edition and it is 100% free since release 10g, with the only limitation being that the cluster may only have a maximum of 4 Sockets in total. For me this is one of the key reasons to look at Standard Edition. There are a number of companies out there requiring a more highly available solution and using Enterprise Edition with the Oracle RAC options due to high cost, is just not an option. But now they have this amazing technology at their fingertips at no additional cost. If your application or business requirements does not need any of the Enterprise Edition only options such as Partitioning, Parallel Query or Flashback Database, using Standard Edition together with Oracle RAC is truly an option worth considering.

With Oracle RAC in a Standard Edition environment, there are some strict requirements as well as included options:

- Oracle Advanced Security SSL/TLS is included
- Maximum of 4 CPU sockets allowed in the Cluster
- Automatic Storage Management (ASM) must be used for all Oracle database file types.
- 3rd Party Clusterware is not supported, you have to use Oracle Clusterware
- No other Clusterware software can be installed on the same system running the Oracle RAC configuration, this list include OCFS, OCFS2 and any 3rd party cluster file systems.

- **Standby Databases**

Another critical component in achieving high availability is Standby Databases, and YES this option can be implemented on Standard Edition, even when using Oracle Standard Edition with Oracle RAC. This is done using 3rd party products or even custom scripts - if you have the time and knowledge to write them.

- **Backup and Recovery (Recovery Manager - RMAN)**

Oracle Recovery Manager (RMAN) can perform backup and recovery tasks in all the database editions, even in Oracle Express; but there are limitations when working with Standard Edition and Standard Edition One, of which two of the key ones not available are parallel backup options and block-level recovery. However the core functionality of RMAN is fully available in all editions and makes this one of the most important components of implementing a highly available solution.

- **Flashback Technology (Flashback Database, Table, Transaction and Query)**

If we look at the flashback options, unfortunately most of the flashback options such as Flashback Database, Table and Transaction are only available in Enterprise Edition. But one of them, Flashback Query is available in Standard Edition, and even though it is not as powerful as the other Flashback options, it is still worth knowing it is there (it might come in quite handy one day). See "Appendix B: Flashback Query" for a quick summary of Flashback Query.

Before we get into an example, it is good to have an understanding of what Oracle Standard Edition's capability with regards to Disaster Recovery is.

Standard Edition and Disaster Recovery

Three disasters that always get mentioned when talking about Disaster Recovery (DR) are:

- Natural Disasters
- Infrastructure (Hardware) Failure
- Human Error

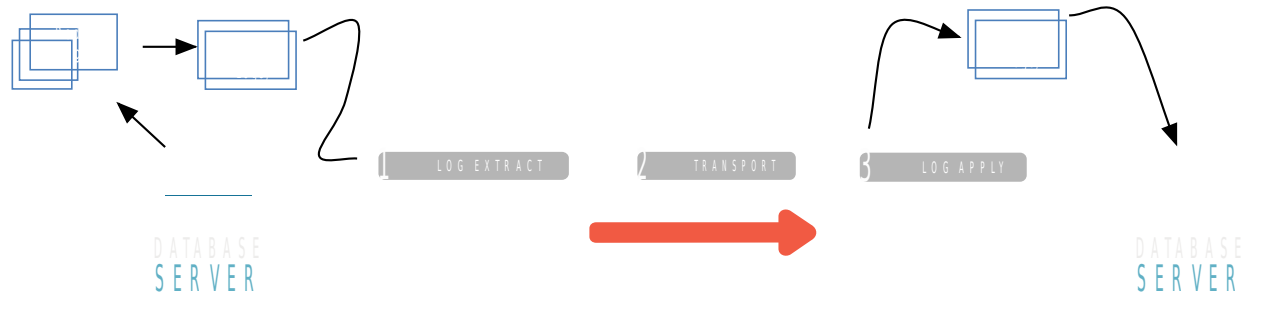
To counter-act these disasters, we implement processes, policies and procedures to provide comfort to the organization so that if disaster struck, infrastructure, applications and databases could continue to function with limited downtime or can be recovered within agreed time frames.

In most cases this could include having a second datacenter in a different geographical location. Data is replicated using one of many methods between your primary datacenter to your “backup” secondary site - your DR site. All of this; the set up and configuration of the new infrastructure, as well as establishing the processes, policies and procedures are critical to ensure business continuity and this can be seen as the core of Disaster Recovery.

What can Standard Edition provide? Does it even cater for Disaster Recovery? The quick and easy answer to this question is YES. Standard Edition and Standard Edition One does provide options for Disaster Recovery of which one key option stands out; Standby Databases. Most don't realize that even though Data Guard is only available to Enterprise Edition, Standard Edition can still provide you with Standby Database functionality. The main issue with this is that you have to build some of the components not available to achieve this. Example processes that will need to be implemented manually include (not limited to):

- The creation of the standby database (clone of the primary database)
- The creation of standby control file
- The automatic shipping of redo information from the primary database to the standby database
- The starting of recovery sessions to apply this redo to the standby database.
- Management of the archive logs after they were applied
- Performing role-switch operations where the primary and standby can switch roles
- Activating (failing over to) the standby database in case of disaster

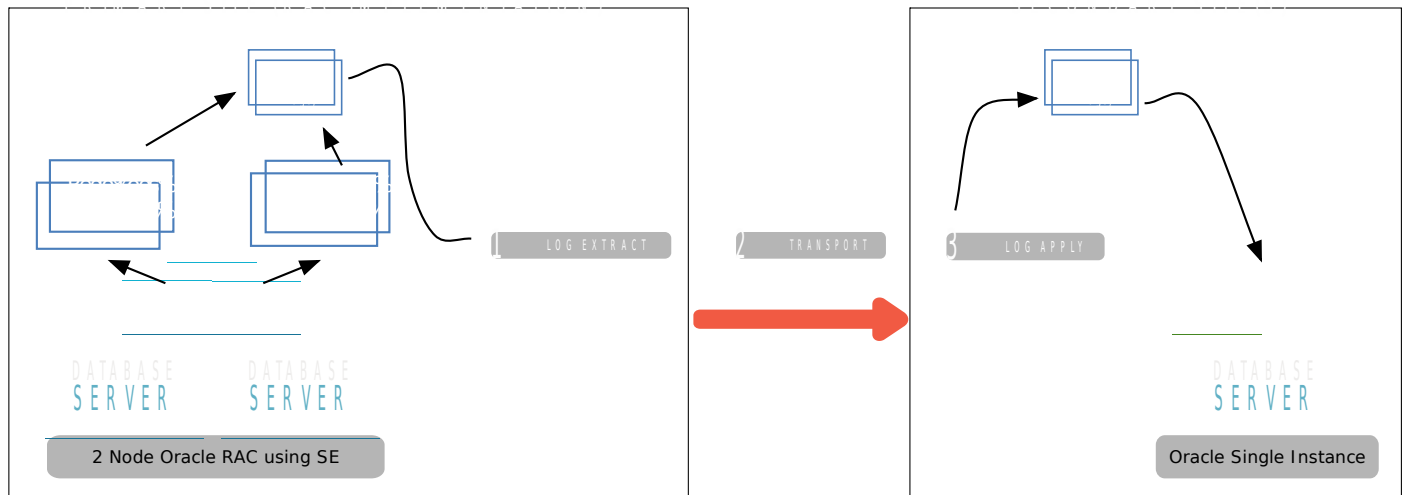
Below is an image that provides you with a high level overview of a standby database.



In summary the 3 Key steps required for implementing and maintaining a standby database are:

- **Archive Log Extraction**
Archive Logs are key to database recovery and are also used to move redo information from your primary database to the standby database. The first step is to extract the archive logs from the primary environment as they could possibly be located in an ASM disk group.
- **The Redo Transport**
The second step is the transferring of the Archive Log between the primary and the standby server. Using Oracle Net for this is not available. Archive logs can be transferred using other network copy operations.
- **Redo (Archive) Apply**
The last step in the process is applying the archived log to the standby database. This process brings the standby database up to date with the standby. The key in this process is not to lose any archive logs as you will then end up with an unrecoverable gap between the primary and standby database.

Even if you are running an Oracle Standard Edition environment with Oracle RAC on the primary side, having a standby database is possible. The principles stay the same. The archive logs from the Oracle RAC database is shipped to the standby server, which will then apply the archive logs (redo) from both threads to the standby database. Below is a high-level demonstration of this:



The next section will provide you with an overview of how you can achieve a high availability environment using a Standby Database to provide you with a solution that not only provides higher availability, but also Disaster Recovery capability.

Test Case: Oracle RAC With Standby Database using Standard Edition

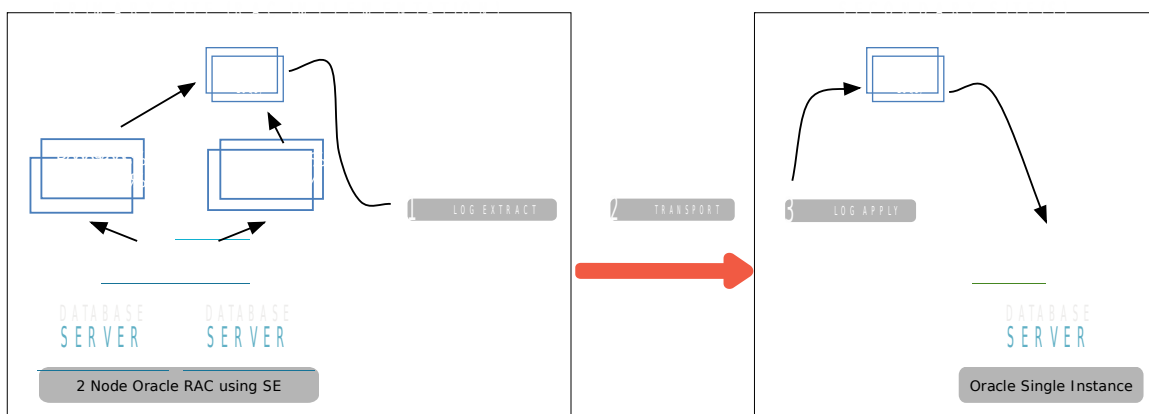
Oracle RAC is in a league of its own, a powerful option that provides you with highly available, scalable solution at an affordable cost as you only pay for the database license, not the RAC license. Implementing a standby database for an Oracle RAC environment is possible and as mentioned before can be done in two ways: writing your own custom scripts, or by using 3rd Party products. In this Test Case I will provide a high level overview of how you can implement and maintain a standby database using 3rd Party product called Dbvisit Standby.

To implement a standby database in an Oracle RAC environment using Dbvisit Standby is not that complex and I will provide you with the steps you need to know to get started. The table below provides a high level summary of the test case environment.

Primary Environment (Oracle RAC Cluster)	Standby Environment (Single Node)
2 x Oracle Linux 5 64bit (update 7) Nodes Oracle Clusterware version 11.2.0.3 configured with ASM storage, using two disk groups +DATA and +FRA Oracle Database Standard Edition (11.2.0.3.0) Oracle Home: /u01/app/oracle/product/11.2.0/db_1 Primary Server Names: dbvrlin307.dbvisit.co.nz and dbvrlin308.dbvisit.co.nz	Oracle Linux 5 64bit (update 7) Oracle Grid Infrastructure (11.2.0.3.0) with ASM storage, using two disk groups +DATA and +FRA Oracle Database Standard Edition (11.2.0.3.0) Oracle Home: /u01/app/oracle/product/11.2.0/db_1 Standby Server Name: dbvlin309.dbvisit.co.nz Standby Database Name: DEV

Database Name: DEV Datafile Locations: +DATA Fast Recovery Area: +FRA	Datafile Locations: +DATA Fast Recovery Area: +FRA
---	---

The diagram below provides a high level overview of the implementation:



The 3 step process as shown in the above diagram performing the extract, transportation and apply of the redo information is some of the key steps that should be performed by the 3rd party products to maintain a standby database in a standard edition configuration.

Step 1. Perform Prerequisite Steps

Before installing Dbvisit Standby the following should be reviewed

- Add entries to your `/etc/oratab` file (Unix environments) to allow you to quickly set your environment using the `“. oraenv”` script. An example entry in the `/etc/oratab` file is in format: `$ORACLE_SID: $ORACLE_HOME:<N|Y>`
- Ensure Primary Database is in Archivelog Mode
 - o Review with “archive log list” command
 - o Enable with “alter database archivelog;” while in mounted state. (Done from one instance in cluster)
- Enable Force Logging on the Primary
 - o Run command “alter database force logging;”
- Create the directory `/usr/local/dbvisit` on all nodes and ensure the oracle software owner own this directory

Step 2. Install Dbvisit Standby

This is a simple process and the online documentation will explain all the steps required in detail. The high level steps of the installation can be summarized as follow:

- Download the latest Dbvisit Standby Software into a temporary folder, example `/home/oracle/dbv_6042`
- Run the following to start the installer (as the Oracle software owner):
 - o `cd /home/oracle/dbv_6042`

- o `unzip *.zip; tar xf *.tar; cd dbvisit; chmod 755 dbvisit_install; ./dbvisit_install`
- o This will start the installer from where you can respond to the questions such as installation directory and software owner.
- o Perform the above tasks on all Primary RAC nodes as well as on the Standby database server. It is recommended to install the software on all nodes in `/usr/local/dbvisit` as the Oracle database software owner which in this case will be the UNIX oracle account. The size of the installation is less than 40MB.
- You now have two options, you can either do the configuration using the GUI which would have been started on all nodes following the above installation (https://your_server_name:8443) or you can use the command line (run `/usr/local/dbvisit/standby/dbvisit_setup`). This initial configuration needs to be run only on both the primary nodes, but not on the standby server.
- During this configuration process you will be creating Dbvisit Standby Configuration files also referred to as the DDC file. There will be one per primary node. Example for the DEV1 instance running on `dbvrlin307` you will have a `dbv_DEV1.env` configuration (DDC) file located in `/usr/local/dbvisit/standby` and for the DEV2 instance running on the `dbvrlin308` node will have a `dbv_DEV2.env` configuration (DDC) file in `/usr/local/dbvisit/standby`. These configuration files are specific to that node. All the detail required to create this file is filled in while running the initial configuration.
- Once the installation is complete and you have created the DDC configuration files you are ready to create the database.

Step 3. Create Standby Database

This process is also known as the CSD process. Creating the standby database is an easy process and can be done by either running the “SETUP -> Create Standby Database” option or by running `/usr/local/dbvisit/standby/dbvisit_setup` from the command line and selecting option 7. Create Standby Database.

This process will only take a few minutes and you will be asked to provide information about your standby server configuration. Once done, you can start the standby database creation process. Dbvisit Standby will make use of RMAN to backup the primary database and restore it on the standby server. The standby database creation process may take some time depending on the size of the primary database.

Once this process is done, you will have a fully functional standby database on the standby server.

Step 4. Sending and Applying logs

Once the installation is complete you can start sending logs from the primary nodes to the standby as well as apply the received archived redo logs on the standby. Dbvisit Standby is schedule based, meaning you will need to create a schedule to run these “sending” and “application/apply” of logs. The default command to run is the same for sending and applying logs: `/usr/local/dbvisit/standby/dbvisit <DDC Name>`. Example in this environment you would run the following:

Primary Node 1: dbvrlin307

```
cd /usr/local/dbvisit/standby  
./dbvisit DEV1
```

Primary Node 2: dbvrlin308

```
cd /usr/local/dbvisit/standby  
./dbvisit DEV2
```

On the standby (single instance) environment: dbvrlin309

```
cd /usr/local/dbvisit/standby  
./dbvisit DEV1
```

Step 5. Other options

There are a number of options available to perform Failover, Graceful Switchover or even run internal Dbvisit Reports. These options are also available from the command line as well as the GUI interface. See Appendix A for an example of the GUI Interface. The key commands to be aware of on the command line include:

- o Send or Apply logs: `./dbvisit <DDC>`
- o Perform Switchover: `./dbv_oraStartStop switchover <DDC> <random_Unique_key>`
- o Perform Activation: `./dbv_oraStartStop activate <DDC>`
- o Run a log gap report: `./dbvisit -i <DDC>`

It is not the focus of this white paper to show you the details of Dbvisit Standby, but purely to highlight that there are options available to help you create and maintain standby database configurations when using Oracle Standard Edition. For more details on these products please review their online documentation.

Is Two and Four Sockets Really a Limitation?

One of the restrictions when looking at Standard Edition One and Standard Edition is the amount of CPU sockets allowed to be used in the configuration. For Standard Edition One the limit is up to maximum of two CPU sockets where with Standard Edition this is increased to a maximum of four CPU sockets, even if Oracle RAC option is used (Maximum four Sockets in cluster). Now for most this might seem like a showstopper, but it might not be.

In the last five years, the CPU modules have improved drastically and more and more cores are now found on these models adding more and more processing power to one CPU module. Just take the Intel E7-8870 CPU as an example. It has 10 cores and 20 threads. On top of that, this is a 2.40Ghz CPU with 30MB cache. That is a lot of processing power on one CPU module. Now image four of these in one system or a two node Oracle RAC cluster consisting out of two nodes each running two of these modules. You have all this processing power to your availability and yet you are only paying for the four sockets.

With the pace and speed of the development going into these CPU modules you might find that every year new modules are released, faster and with more cores on them. Maybe you do not have to upgrade your licensing but rather upgrade your old hardware. By just doing that, you might be able to meet your processing requirements and still stay within your limited budget.

Reasons to Consider using Standard Edition

Standard Edition is still the same core database we know and is rated at the top of its class. Oracle Database technology is at the forefront of new innovation providing exceptional performance, high availability, scalability, backup and recovery and Standard Edition is part of this proven technology. For me there are a few key reasons to consider using Standard Edition:

- Standard Edition One and Standard Edition is proven database technology
- Affordable cost of ownership
- Standard Edition One pricing is truly attractive and Standard Edition almost a third of the Enterprise Edition price tag
- Oracle Real Application Clusters (RAC) is FREE with Standard Edition
- Disaster Recovery with Standard Edition is still possible with Standby Databases
- The Automatic Workload Repository, performance diagnostics and tuning options are not available for Standard Edition, but what about *Statspack*? It is powerful and still there for you to use as well as many other 3rd Party products that can help you with performance tuning when using Standard Edition.

Conclusion

There are a number of database editions available. Which one do you choose? It should not be purely on cost. If that is the case everyone will select Oracle Standard Edition One. Before you select the database edition to use in your environment, you need to review the following:

- Oracle licensing structure and costs associated with each available option
- Review each Database Edition Capability. Make sure you understand what is included and excluded
- Then most importantly: review your Business and Application Requirements. Make sure you know what the options and requirements are. You might be surprised to find Standard Edition to be the perfect fit.

This paper provided you with an overview of the different editions available, as well as some examples of cost comparison to give you an idea on the cost benefit of using Oracle Standard Edition. But most importantly, Oracle Standard Edition might be the smaller brother of Enterprise Edition, but with Oracle RAC and Standby databases being available, you can still achieve a highly available environment with DR capability at an affordable cost.

Appendix A: Oracle Processor Core Factor Table Extract

Below is an extract of the latest Processor Core Factor table available at the time of writing this white paper. A full copy of this document can be found here: <http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>

Oracle Processor Core Factor Table

Effective Date: March 16, 2009

Vendor and Processor	Core Processor Licensing Factor
Sun and Fujitsu UltraSPARC T1 processor (1.0 or 1.2 GHz) Only named servers including: Sun Fire T1000 Server, SPARC Enterprise T1000 Server*, with 6 or 8-core 1.0 GHz UltraSPARC T1 processor Sun Fire T2000 Server, SPARC Enterprise T2000 Server*, with 4, 6, or 8-core 1.0 GHz, or 8 core 1.2 GHz UltraSPARC T1 processor Sun Netra T2000, 1.0 or 1.2 GHz UltraSPARC T1 processor	0.25
SPARC T3 processor	0.25
Sun and Fujitsu UltraSPARC T1 1.4 GHz Only named servers including: Sun Fire T2000 Server and SPARC Enterprise T2000 Server*, with 8-core, 1.4 GHz UltraSPARC T1 processor	0.5
Sun T6300, 1.4 GHz UltraSPARC T1 processor	0.5
AMD Opteron Models 13XX, 23XX, 24XX, 32XX, 41XX, 42XX, 43XX, 61XX, 62XX, 63XX, 83XX, 84XX or earlier Multicore chips	0.5
Intel Xeon Series 56XX, Series 65XX, Series 75XX, Series E7-28XX, Series E7-48XX, Series E7-88XX, Series E5-24XX, Series E5-26XX, Series E5-46XX, Series E5-16XX, Series E3-12XX or earlier Multicore chips	0.5
Intel Itanium Series 93XX or earlier Multicore chips <i>(For servers purchased prior to Dec 1st, 2010)</i>	0.5
Intel or AMD Desktop, Laptop/Notebook, or Netbook Multicore chips	0.5
Sun UltraSPARC T2+	0.5
SPARC64 VII+	0.5
SPARC64 X	0.5
SPARC T4 processor	0.5
Sun and Fujitsu SPARC64 VI, VII	0.75
Sun UltraSPARC IV, IV+, or earlier Multicore chips	0.75
Sun UltraSPARC T2	0.75

Appendix B: Flashback Query

Have you ever run a delete statement just to kick yourself two minutes later for doing that as you still need the data? I know I have and this is something I am sure happens to most during their DBA career. But it does not have to be the end of it. Oracle introduced a truly powerful option since Oracle 9i called Flashback Query. With Flashback Query you can “go-back in time” and get the data back.

Flashback Query is making use of Undo information to make this possible and the following is required:

- Have to use Automatic Undo
- Sufficient space in the Undo Tablespace
- Adjust the UNDO_RETENTION (in seconds) as required

In summary Flashback Query allows you to view data as it was at a particular point in time in the past, using **TIMESTAMP** or **SCN** to point to a point back in time can do this in multiple ways:

- Using the **DBMS_FLASHBACK** package functions:
 - o `get_system_change_number ();`
 - o `enable_at_system_change_number();`
 - o `enable_at_time();`
 - o `disable();`
- From 10g the “**AS OF**” clause can be used with **TIMESTAMP** and **SCN**:
 - o `select * from <table> AS OF TIMESTAMP <timestamp>;`
 - o `select * from <table> AS OF SCN <scn>;`

Example A Using Flashback Query

In this basic example you can see the power of Flashback Query. First a table “test” is created in the schema “aels”. Two records are inserted into this table. The current SCN number is extracted and then a row is deleted with a commit being executed after that. Flashback Query using “**AS OF SCN**” option is then used to get a view of the data as it was at that time (SCN number).

```
SQL> grant execute on dbms_flashback to aels;
Grant succeeded.

SQL> connect aels/aels
Connected.

SQL> create table test (id number);
Table created.

SQL> insert into test values (1);
1 row created.

SQL> insert into test values (2);
1 row created.

SQL> commit;
Commit complete.

SQL> select dbms_flashback.get_system_change_number from dual;

GET_SYSTEM_CHANGE_NUMBER
-----
```

4300790

```
SQL> delete from test where id=1;
1 row deleted.
```

```
SQL> commit;
```

Commit complete.

```
SQL> select * from test;
```

```
      ID
-----
      2
```

```
SQL> select * from test as of scn 4300790;
```

```
      ID
-----
      1
      2
```

```
SQL> select * from test;
```

```
      ID
-----
      2
```

References

<http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>

<http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>

<https://shop.oracle.com>

<http://www.oracle.com/us/corporate/pricing/sig-070616.pdf>

Oracle® Database Licensing Information 11g Release 2 (11.2), Part Number E10594-26

Oracle® Database Advanced Application Developer's Guide 11g Release 2 (11.2), Part Number E25518-06

Dbvisit Software Limited (www.dbvisit.com)