

Oracle Database Appliance – Deep Dive

David Hueber, dbi services ltd.

Introduction

In 2011 Oracle has introduced a new Engineered System called Oracle Database Appliance. This platform, meant for databases consolidation and high availability, is built almost as a “Plug- And-Play” solution. This article will provide an overview of the ODA architecture and base usage (updated according latest releases) as well as the keys to go beyond the black box to get benefit of the whole power of the solution.

ODA what for?

The Oracle Database Appliance is part of the Oracle's Engineered System portfolio. If it is definitely not a "low-cost" Exadata, which is designed for high performances and high volumes, ODA provides an Easy, Reliable and Available database consolidation solution.

The ODA is built on a fully redundant hardware and integrates a new framework to ease database creation and management as well as some (but not all!) high available technologies such as RAC One Node and RAC.

ODA is also the first platform integrating a Pay as you Grow licensing model for Oracle databases.

ODA Architecture

The Oracle Database Appliance, since its version X3-2 which came out in Q3 2013, is based on three individual components: 2 servers, 1 storage shelf.



Figure 1: ODA front panel view

In its 2014 version, ODA X4-2, each server includes:

- 2 Intel Xeon E5-2697v2 12 cores
- 256 GB Memory
- 2 SAS 10k 600GB – mirrored in RAID 1 and used for boot and binaries
- 2 SAS HBA adapters
- 2 Twinnax 10Gb/s Interconnect
- 4 Ethernet 100Mb/s ports

The storage shelf includes 24 disks spread as following:

- 20 SAS 10k 900GB – used for Data and Recovery files
- 4 SSD 200GB – used for Redo Logs and Control Files

The whole hardware architecture, including cooling or power supplies, on ODA is fully redundant. Storage access for each server node is done through two SAS wire connected to two distinct controllers both served in the shelf by two expanders. In addition Oracle Automatic Storage Management (ASM) is used for data redundancy. If any component fails, the storage access is still guaranteed.

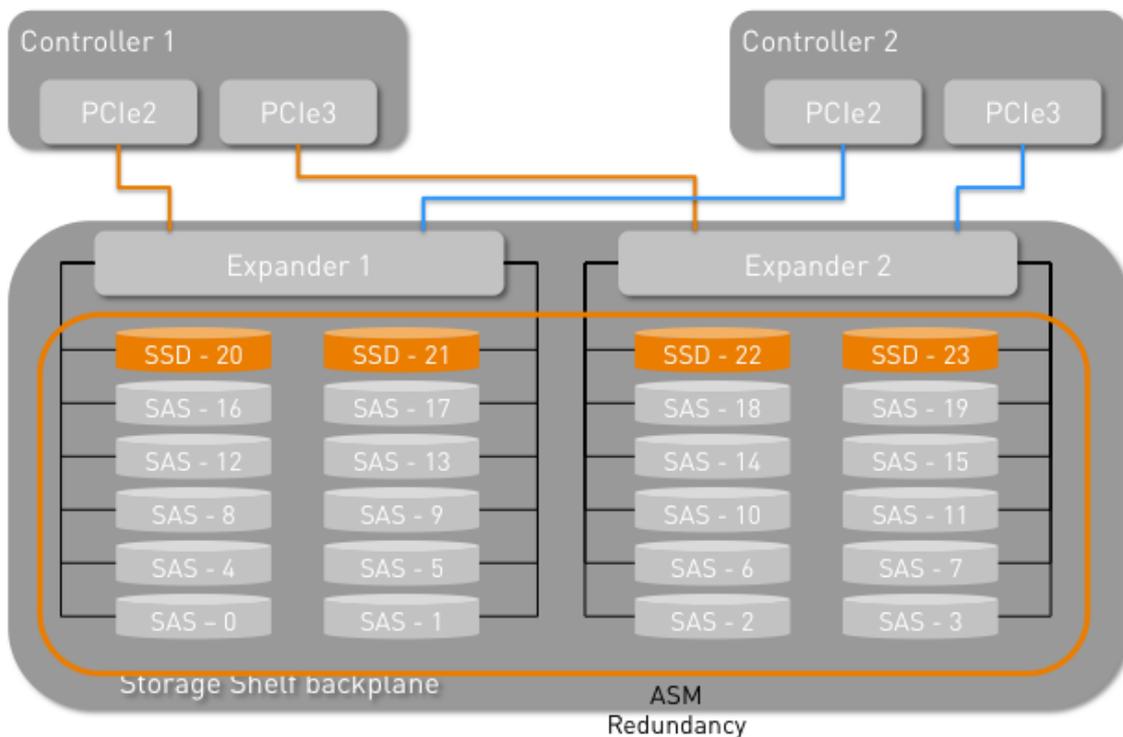


Figure 2: ODA Storage access architecture

While talking about storage capacity, two parameters have to be taken into account:

- ASM Redundancy
- Backup storage location

ODA proposes to use either the Normal or High Redundancy modes from ASM taking the raw storage capacity from 18TB to either 9TB (Normal Redundancy) or 6TB (High Redundancy) down.

The second point that directly impacts the storage capacity is if the databases' backups will be stored on the ODA itself (in the Fast Recovery Area) or on an external medium such as Tapes or NFS shares.

Depending on the chosen solutions, the ASM disk groups' sizing is adjusted.

Backup mode	+DATA (TB)	+RECO (TB)	+REDO (GB)
Normal Redundancy			
Local Backup	3,6	4,5	248
External Backup	7,2	0,98	248
High Redundancy			
Local Backup	2,4	3	248
External Backup	4,8	0,65	248

We can see that depending on the settings chosen for the Redundancy and the Backup Location, the ODA storage capacity for Oracle databases varies from 2.4 TB to 7.2 TB.

A last point to note is that these settings are configured during the ODA deployment and cannot be changed afterward without reinstalling the ODA!

ODA software stack

Basically any ODA since version 2.5 can be installed in two different modes/architectures:

- Bare Metal
- Virtualized

Bare Metal is a fully database dedicated ODA as it was designed in its version 1.0. It is an appliance based on Oracle Linux and including Grid Infrastructure and Oracle Database.

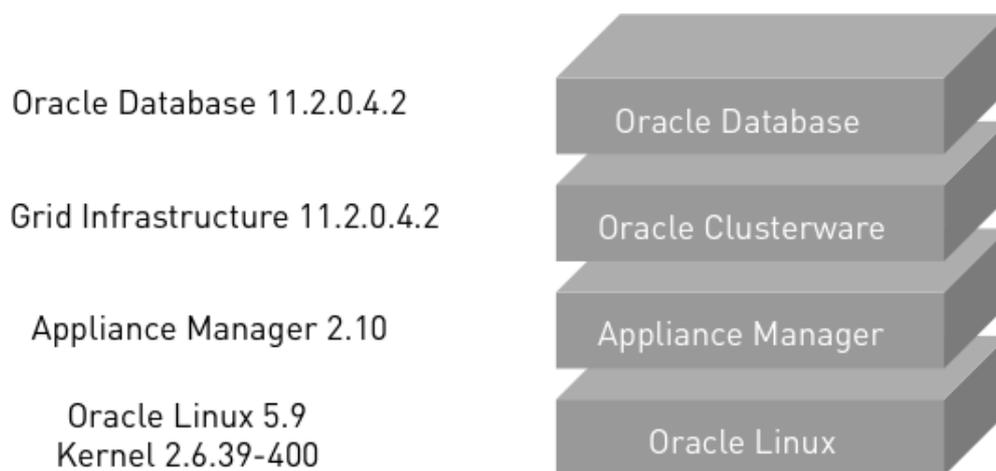


Figure 3: ODA Bare Metal software stack

The Virtualized mode, introduced in version 2.5, integrates an Oracle VM layer allowing running applications' virtual machines such as Weblogic, JD Edwards, Tomcat beside a virtual database server.

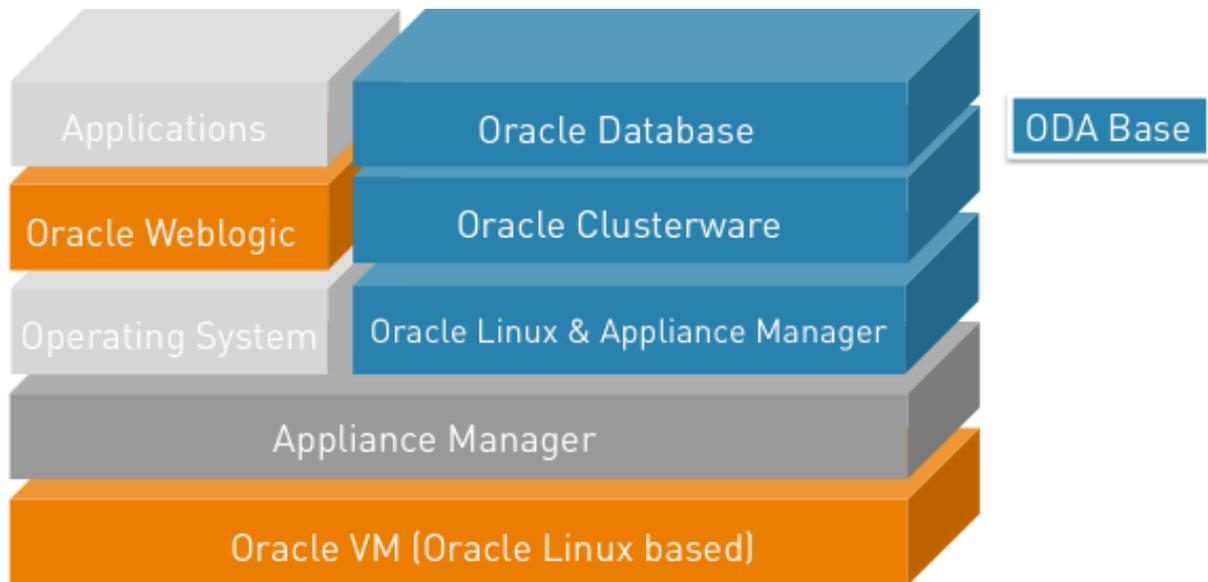


Figure 4: ODA Virtualized software stack

In this architecture, all databases run in a single and dedicated virtual environment so called ODA BASE. Additional virtual machines can be setup beside the ODA BASE to run any software component required.

The ODA BASE virtual machine is the driving director of the whole platform and is used to:

- Manage databases
- Manage binaries
- Manage virtual machines

For applications virtualization several templates are already available for download from edelivery.oracle.com/linux.

In addition Oracle provides a Weblogic appliance for ODA that provides a Java based installer, which deploys automatically an high available and virtual Weblogic infrastructure including Oracle Traffic Director (<http://www.oracle.com/technetwork/middleware/weblogic-oda/downloads/index.html>)

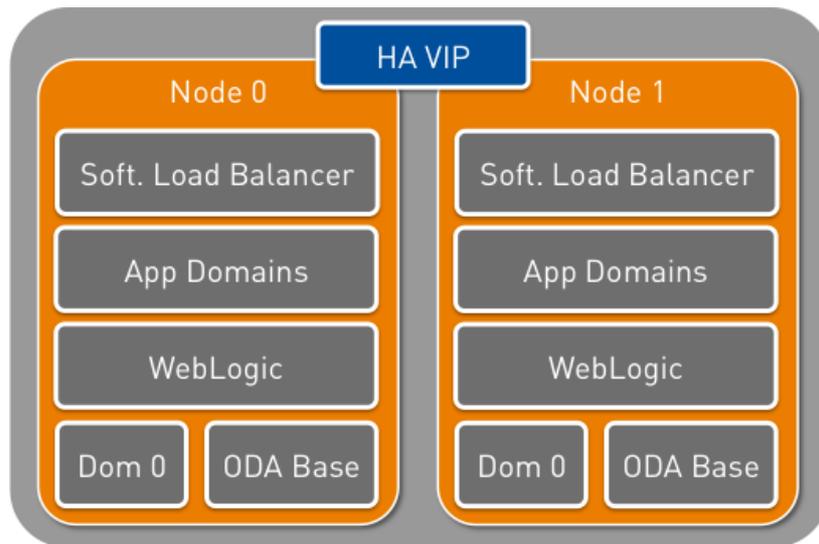


Figure 5: Weblogic appliance on ODA

Appliance Manager overview

While using Oracle Database Appliance, the biggest change for Oracle DBAs is the Appliance Manager OAKCLI.

OAKCLI is a new Command Line Interface (CLI) dedicated to ODA that allows to:

- List the hardware components
- Validate & diagnose the hardware components
- Install and upgrade software
- Apply patches
- Create & Drop Databases
- Install & uninstall Oracle Homes
- Deploy & manage virtual machines

In addition the OAKCLI commands shall always be started as root user from the first node (node 0). Otherwise the ODA triggers an error message.

```
[oracle@dbi-oda2 ~]$ oakcli create database -d DBITEST2
sh: dmidecode: command not found
ERROR: 2014-01-06 22:53:25: Insufficient privileges to create the database

[root@dbi-oda2 ~]# oakcli create database -d DBITEST2
ERROR: 2014-01-06 22:54:18: oakcli create database should be invoked from
the first node
```

Checking disk using OAKCLI

```
[root@dbi-oda1 ~]# oakcli show disk
```

NAME	PATH	TYPE	STATE	STATE_DETAILS
e0_pd_00	/dev/sda	HDD	ONLINE	Good
e0_pd_01	/dev/sdb	HDD	ONLINE	Good
e0_pd_02	/dev/sdaa	HDD	FAILED	DiskRemoved
e0_pd_03	/dev/sdab	HDD	ONLINE	Good
e0_pd_04	/dev/sdac	HDD	ONLINE	Good

Validating storage structure...and cabling

```
[root@dbi-oda2 ~]# oakcli validate -c storagetopology
```

```

It may take a while. Please wait...
INFO      : ODA Topology Verification
INFO      : Running on Node1
INFO      : Check hardware type
SUCCESS   : Type of hardware found : X3-2
INFO      : Check for Environment(Bare Metal or Virtual Machine)
SUCCESS   : Type of environment found : Bare Metal
INFO      : Check number of Controllers
SUCCESS   : Number of Internal LSI SAS controller found : 1
SUCCESS   : Number of External LSI SAS controller found : 2
INFO      : Check for Controllers correct PCIe slot address
SUCCESS   : Internal LSI SAS controller      : 50:00.0
SUCCESS   : External LSI SAS controller 0    : 30:00.0
SUCCESS   : External LSI SAS controller 1    : 40:00.0
INFO      : Check if JBOD powered on
SUCCESS   : 0JBOD : Powered-on
INFO      : Check for correct number of EBODS(2 or 4)
FAILURE   : Check for correct number of EBODS(2 or 4) : 1
ERROR     : 1 EBOD found on system, which is less than 2 EBODS with 1 JBOD
INFO      : Above details can also be found in log
file=/opt/oracle/oak/log/dbi-oda2/storagetopology/StorageTopology-2014-01-07-23:28:52_16620_29386.log

```

Managing databases using Appliance Manager

Once the ODA is configured (network, hostname, binaries deployment...), databases can be created. This operation is done using the Appliance Manager, which means that it has to be started as root user on node 0.

While creating a database several parameters can be given:

- Database name (mandatory)
- Database version
- Database NLS configuration file

```
[root@ODADBI1-base ~]# oakcli create database -db DBPROD -version
11.2.0.3.10 -params dbprodconf
```

The CREATE command starts then a wizard that requires some basic inputs such as root, oracle and sysasm users' password but also the type of database and running node:

```
Please select one of the following for Database Deployment [1 .. 3]:
1 => EE : Enterprise Edition
2 => RACONE
3 => RAC
1
Selected value is: EE
Please select one of the following for Node Number [1 .. 2]:
1 => ODADBI1-base
2 => ODADBI2-base
2
Selected value is: ODADBI2-base
```

...as well as the database class:

```
Specify the Database Class (1. Medium 2. Others) [1]:2
Please select one of the following for Database Class [1 .. 8] :
1 => Very Very Small
2 => Very Small
3 => Small
4 => Medium
5 => Large
6 => Extra Large
7 => Extra Extra Large
8 => Extra Extra Extra Large
3
Selected value is: Small
```

Indeed the database class only defines the dbca template to be used for the database creation. ODA comes with 8 templates and none can be added.

Setting	XXS	XS	S	M	L	XL	XXL	XXXL
CPU_COUNT	2	2	4	8	12	24	32	48
SGA (GB)	2	4	8	16	24	48	63	96
PGA (GB)	1	2	4	8	12	24	32	48
Processes	200	200	400	800	1200	2400	3200	4800
Redo (GB)	1	1	1	2	4	4	4	4
Nb DBs	24	24	12	6	4	2	1	1
IOPS	137	137	275	550	825	1650	3300	3300

Note that these settings, especially the number of databases and the IOPS, are mainly informational. The next steps are then to customize the environment to YOUR needs...and that is there the deep dive starts ☺

Going beyond limits!

The ODA provides definitely a quick and easy way to get databases running on a high redundant architecture that provides great performances. However as each application is different, some customization are still needed. In addition the way databases are implemented by OAKCLI can still be improved manually by a DBA.

The first topic is the NLS settings for the created database. These are defined by a configuration file used by OAKCLI during the database creation.

A default configuration is available:

```
[root@ODADBI1-base ~]# oakcli show db_config_params -detail
Available DB configuration files are:
default
DATABASE_BLOCK_SIZE      =>      8192
DATABASE_LANGUAGE        =>      AMERICAN
DATABASE_CHARACTERSET     =>      AL32UTF8
DATABASE_TERRITORY       =>      AMERICA
COMPONENT_LANGUAGES      =>      en
```

If different NLS settings shall be used then a customized configuration file must be created before running the database creation itself.

```
[root@ODADBI1-base ~]# oakcli create db_config_params -conf dbprod1
Please select one of the following for Database Block Size [1 .. 4]:
1      => 4096
2      => 8192
3      => 16384
4      => 32768
2
Selected value is: 8192

Specify the Database Language (1. AMERICAN 2. Others) [1]:1
Selected value is: AMERICAN

Specify the Database Characterset (1. AL32UTF8 2. Others) [1]:2

Please select one of the following for Database Characterset [0 .. 10]
:
0      => Others
1      => AL32UTF8
...
...
90     => UTF8
...
96     => WE8ISO8859P15
97     => WE8ISO8859P9
98     => WE8MACROMAN8S
96
Selected value is: WE8ISO8859P15

Specify the Database Territory (1. AMERICA 2. Others) [1]:1
Selected value is: AMERICA

Specify the Component Language (1. en 2. Others) [1]:1
```

```
Selected value is: en
Successfully generated the Database parameter file 'dbprodconf'
[root@ODADB11-base ~]# oakcli create database -db DBPROD -params dbprod
```

Once the database is created, some checks shall be done like the control files and redo logs settings

```
SQL> show parameter control_files
NAME                                TYPE                                VALUE
-----                                -
control_files                        string                              +REDO/dbitest/control01.ctl

SQL> select group#,members from v$log;
GROUP#  MEMBERS
-----  -
1        1
2        1
3        1
```

We can notice that if both are stored in the +REDO disk group (SSD Disks) neither the Control File nor the Redo Logs are multiplexed. If we want to be fully protected against logical corruption, a good practice is to add an additional control file and a redo member per redo log group. However to avoid any performances bottleneck, a good practice is to add them in the +REDO disk group as the first copies.

Another point is the memory consumption of the databases. Since Oracle 11g the AMM principle (Automatic Memory Management – memory_target) has been introduced. Unfortunately AMM is not compatible with the huge pages mechanism. Therefore ODA is using ASSM (Automatic Shared memory Management – sga_target & pga_aggregate_target) in combination with huge pages.

By default any ODA comes out of factory with a huge page settings of 50% of the physical memory of the ODA server.

```
[root@dbi-oda1 ~]# grep -i huge /proc/meminfo
HugePages_Total:    64000
HugePages_Free:     56200
HugePages_Rsvd:      393
HugePages_Surp:      0
Hugepagesize:       2048 kB
```

This means that if you want to use more memory for SGAs in huge pages, follow the MOS note 401749.1 to increase the huge page sizing.

Another good practice after creating databases on ODA is to take look on the tablespaces settings. Whatever the database class (XXL to XXXL) chosen, the database size won't be affected during the creation. Any newly created database will come with following tablespaces:

- SYSTEM
- SYSAUX
- TEMP
- USERS
- UNDO

This makes sense as ODA has no way to anticipate the tablespaces, which will be required for the application. Therefore these have to be created manually afterward. However the base tablespaces are all configured in autoextend with no maximum size.

Name	Nb files	AutoExt	Size GB	Maxsize GB
SYSTEM	1	YES	0.68	Unlimited
SYSAUX	1	YES	0.59	Unlimited
TEMP	1	YES	0.02	Unlimited
USERS	1	YES	0.00	Unlimited
UNDOTBS1	1	YES	0.33	Unlimited

A good practice would be to reconfigure them by setting an appropriated maximum size (i.e.: about 1G to 5G for SYSTEM and SYSAUX).

Once the databases customization is done, some high available principle can be introduced. To do so ODA provides out of the box either RAC One Node or RAC integration. However if these are non-free options for Enterprise Edition, some other solutions could be used:

- Data Guard (included on Enterprise Edition)
- Failover Cluster

While Data Guard would require an additional ODA, Failover cluster can be implemented with a single command as root user:

```
[root@ODADBI2-base bin]# ./crsctl modify res ora.dbprod.db -attr "PLACEMENT=favored"
```

Once done simply copy the database's password file on the second node and then the instance can be "switched" from one node to the other.

```
[root@ODADBI2-base bin]# ./crsctl relocate res ora.dbprod.db
CRS-2673: Attempting to stop 'ora.dbprod.db' on 'ODADBI2-base'
CRS-2677: Stop of 'ora.dbprod.db' on 'ODADBI2-base' succeeded
CRS-2672: Attempting to start 'ora.dbprod.db' on 'ODADBI1-base'
CRS-2676: Start of 'ora.dbprod.db' on 'ODADBI1-base' succeeded
```

Licensing

Finally I just want to give few words about licensing, which in Oracle environments is always a quite tough topic 😊

ODA is the first platform to introduce a Pay as you Grow licensing model. Basically ODA is licensed with the standard Oracle processor licenses with all rules applying on Intel processors (i.e.: core factor). However the huge difference with environments based on third parties' hardware (HP, IBM, DELL...) is that not all installed cores have to be licensed.

You can then, out of the 24 available, start with a minimum number of cores and increase along the time according your needs. Of course you can increase...but not decrease.

For this principle a distinction has to be done between the Bare Metal architecture and the Virtualized one where the number of minimum of core to be licensed isn't the same:

- Bare Metal
 - o Minimum of 4 cores per node
 - o Can be increase by steps of 4 additional cores per node

- Virtualized
 - o Minimum of 2 cores per node
 - o Can be increase by steps of 2 additional cores per node

Conclusion

As stated in the introduction of this article, ODA is a pretty easy, reliable and available platform for Oracle databases consolidation. It provides a straightforward and integrated way to create and manage databases. However to get all benefits from such an environment it is necessary to take a look behind the scene how it works and understand how to influence it.

There are much more interesting topics about ODA such as Data Guard implementation, virtualization, performances or TCO. If you want to get more information about it, feel free to contact us.