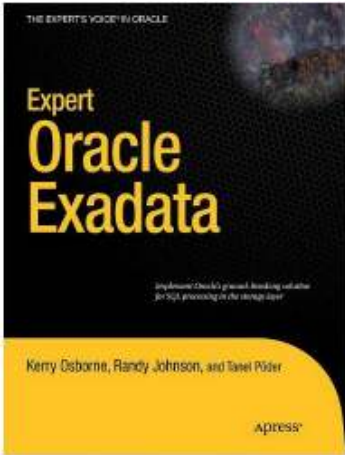


# Enkitech

## *Exadata Storage Layout*



## Randy Johnson

- Principal Consultant, Enkitech LP.
- 20 or so years in the IT industry
- Began working with Oracle RDBMS in 1992 at the launch of Oracle 7
- Main areas of interest are HA, Backup & Recovery
- Working exclusively with Exadata for over a year now.
  - Numerous Exadata implementations
  - Numerous POC's & workshops

# Agenda

1. What is Exadata
2. What Is ASM
3. Storage Cell Disks
4. Creating and Allocating Storage at the Cell
5. Demo – Creating & Presenting Grid Disks to ASM
6. Q&A

# What Is Exadata?

## **Database Servers**

2 - 8 Servers, per rack

- 24 - 96 CPU Cores
- 96 - 768 Gigabytes Memory

## **Storage Cells**

3 - 14 Storage Cells per rack

72 - 336 CPU Cores

72 - 336 Gigabytes Memory

1.2 - 5.3 Terabytes Flash Cache

72 - 336 Terabytes Raw Disk Storage

## **Infiniband Switches**

2 Per rack + 1 spine switch for expansion

40Gb/s - 5 gigabytes per database server / storage cell

**Expandable up to 8 racks - more with additional spine switches**

# A Data Center In A Can

## Database Servers

- Sun Fire 5670, 1U
- 2, Hex-Core CPU's, 2.93GHz
- 96G Memory
- OEL5, 64 bit

## Infiniband Network

- Redundant 40Gb Switches
- Unified Internal Network
- Used by both DB and Storage
- Can connect externally



## Storage/Cell Servers

- SunFire X4270 2U
- 2, Hex-core CPU's, 2.93GHz
- 24G Memory
- 384GB Flash Cache
- OEL5, 64 bit
- 12 Drives per Storage Server
  - 2TB SAS - High Cap
  - 600GB SAS - High Perf.
- 5.4GB / Sec Transfer Rate

# Exadata Configurations

## **Full Rack**

- 8 Database Servers
- 14 Storage Cells

## **Half Rack**

- 4 Database Servers
- 7 Storage Cells

## **Quarter Rack**

- 2 Database Servers
- 3 Storage Cells

# How do the Components Work?

## Exadata Database Servers



## Exadata Storage Servers



### Compute Intensive Processing

- Complex Joins
- Aggregation
- Functions
- Data Conversion
- I/O Over Infiniband Using iDB Protocol

### I/O Intensive Processing

- Index / Table Scans
- HCC Uncompress
- Functions
- Primitive Joins
- Predicate Filtering
- Column Projections
- Storage Index Processing

# What is ASM?

## Database Volume Management

- Dynamic provisioning
- Disk redundancy \*\*
- IO Spread across all disks
- Clustered Storage (using RAC)

## Database File System

- File and directory names
- File management

# S.A.M.E.

ASM is the Implementation of S.A.M.E.

The goal of S.A.M.E. is to optimize storage I/O utilization

- Bandwidth
- Sequential and random access
- Workload types

# Implementing Database Storage Without ASM

## Design DB layout strategy

- How many file systems, data files
- Where data files reside

## Inventory of files:

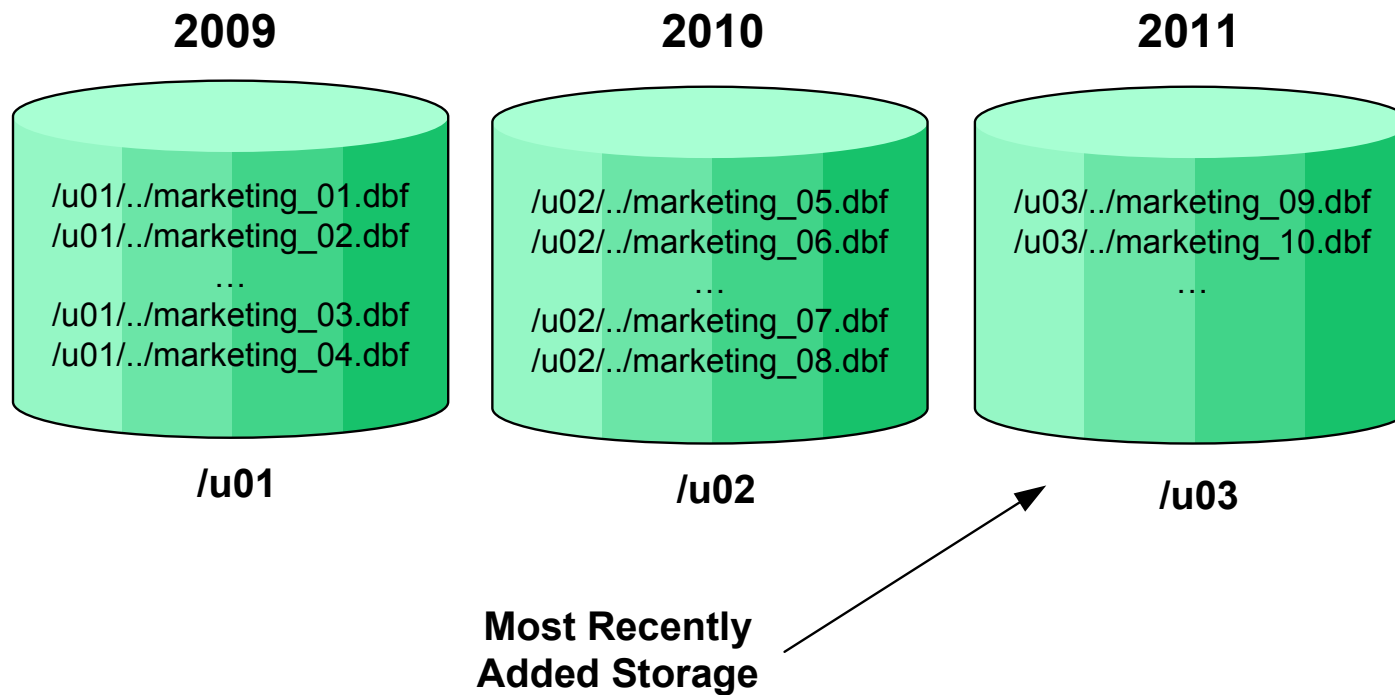
- 2 control files, 2 log files, multiple datafiles per tablespace, backup files, temp files, etc...
- 100's or 1000's of files to create, name, manage and not make mistakes!
- Multiplied by n number of Databases

What about tuning, expanding/contracting your DB ?



# Provisioning with File Systems

Current data is what end users are usually interested in. So as space is added I/O tends to follow the new disks, creating hot spots.



# Virtually Eliminates I/O Tuning

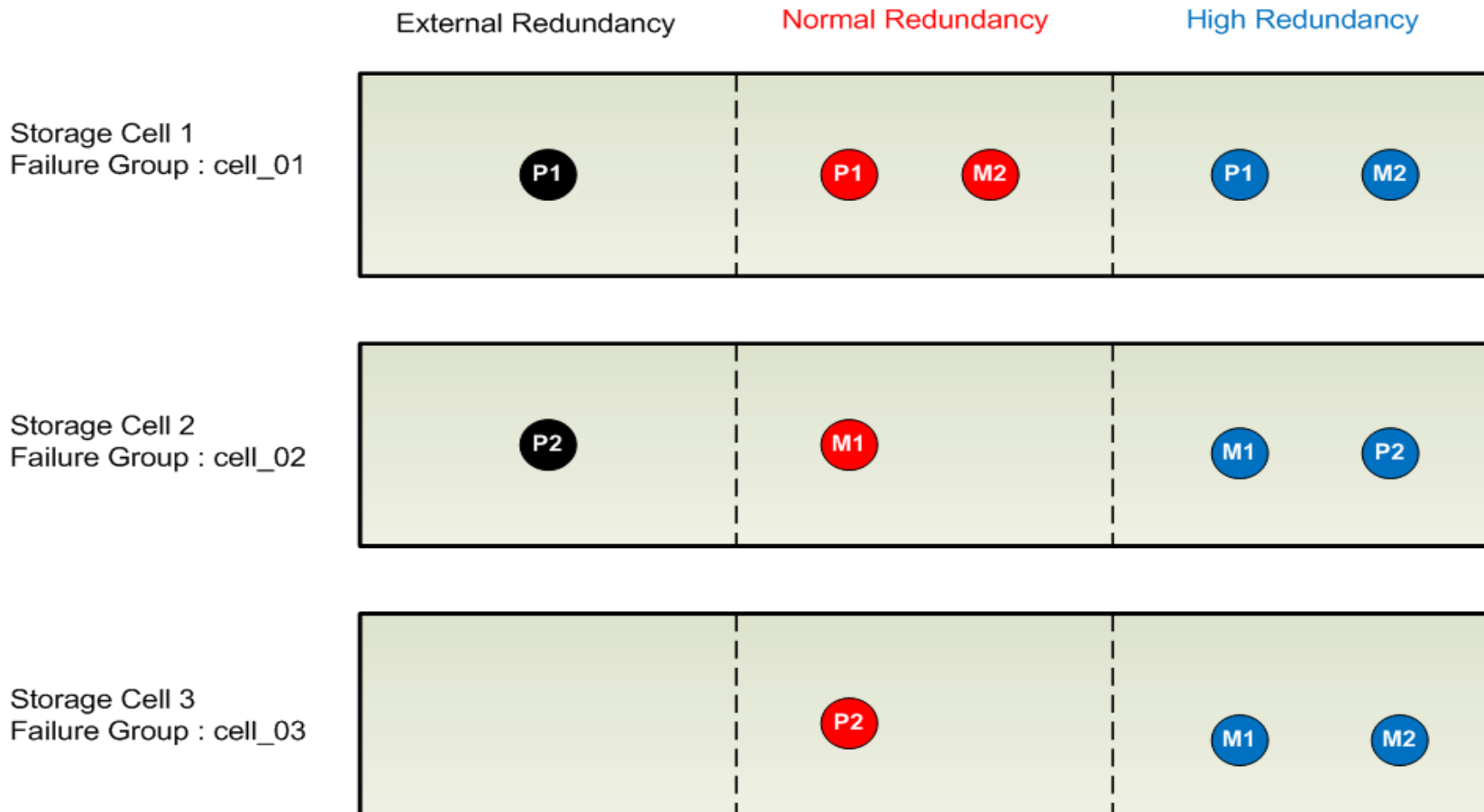
- Data automatically striped and mirrored
- I/O automatically spread across all disks
- I/O is balanced because data is balanced, eliminating hot spots
  - ✓ When disks are added, data is rebalanced
  - ✓ When disks are removed, data is rebalanced

# Data Redundancy

- ASM provides redundancy using failure groups
- Failure groups are user defined units of failure
  - Singular disk
  - Group of disks (SAN, or Exadata Storage Cell)
- Failure groups ensure mirror copies of data are not stored on the same disks (or groups of disks)

# Exadata Storage Server Config

## How Exadata Storage Server Redundancy Works

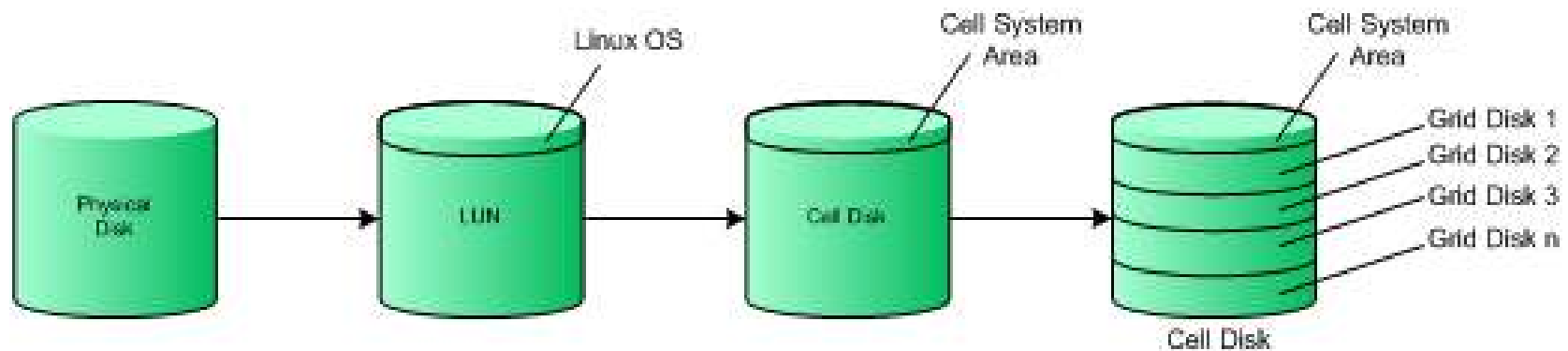


# Exadata Storage Configuration

## Disk Configuration Overview

The CellCLI program is used on the Storage Servers to configure disks for presentation to ASM

- Physical - Physical Disk
- LUN - OS Block Device
- Cell Disk - Used by Cell Server to manage disks
- Grid Disk - Presented to the DB as ASM disks



# Exadata Storage Server Config

## ASM View of Grid Disks

```
SYS:+ASM1> select path, total_mb, failgroup from v$asm_disk
              order by failgroup, path;
```

### Non-Exadata

PATH	TOTAL_MB	FAILGROUP
-----	-----	-----
/dev/rdisk/...	11444	DATA01
/dev/rdisk/...	11444	DATA02
...		

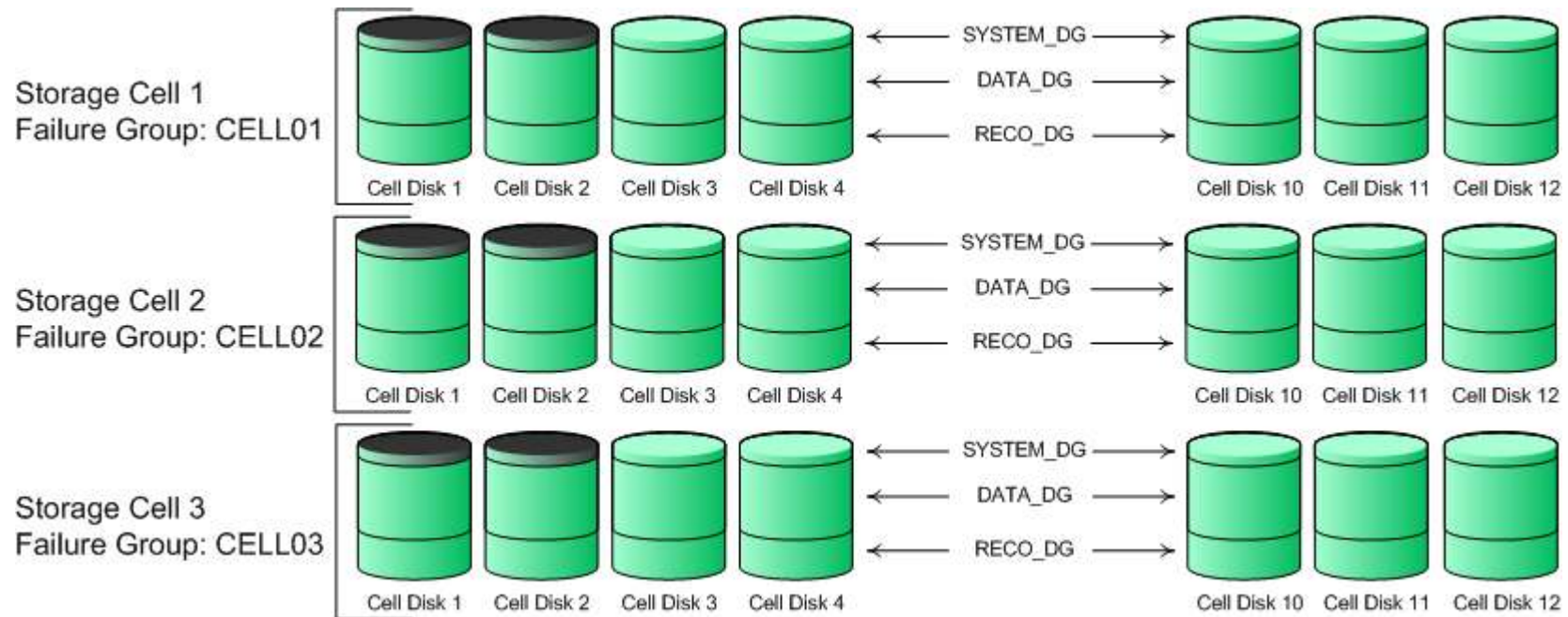
### Exadata

PATH	TOTAL_MB	FAILGROUP
-----	-----	-----
o/192.168.12.3/DATA_CD_00_cell101	1313600	CELL01
o/192.168.12.3/DATA_CD_01_cell101	1313600	CELL01
o/192.168.12.3/DATA_CD_02_cell101	1313600	CELL01
...		

# Exadata Storage Server Configuration

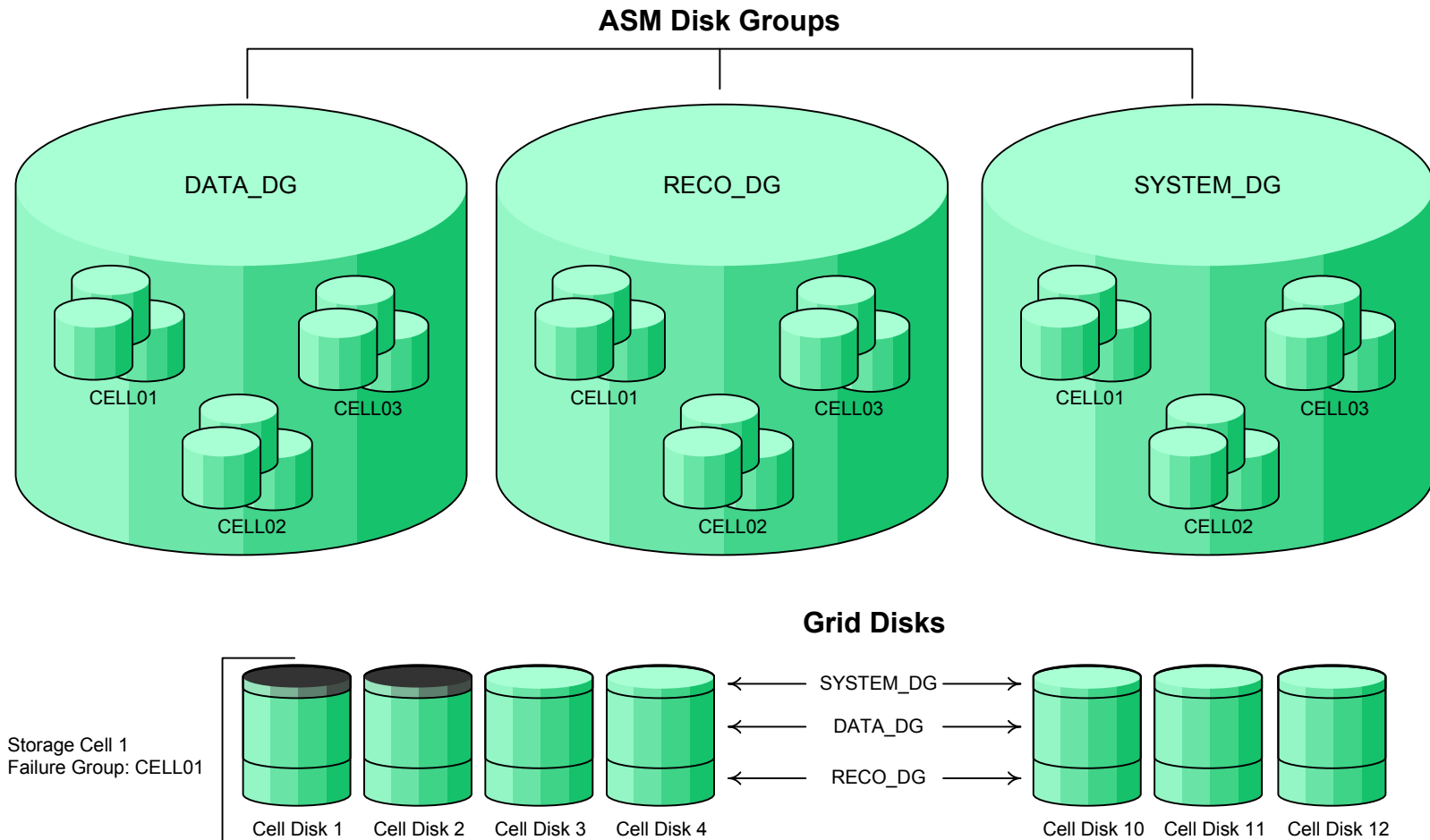
## Data Redundancy - Failure Groups / Mirroring

- Grid Disks are combined into ASM Disk Groups across Cells
- Failure Groups ensure that mirrored data is physically separated



# Exadata Storage Server Configuration

## Typical Exadata Storage Configuration



# Grid Disk Names

## Grid Disk Names

TEST\_DG\_CD\_00\_cell03

└──────────┬──────────┘

Diskgroup      Cell Disk

## Exadata

PATH	TOTAL_MB	FAILGROUP
o/192.168.12.3/DATA_CD_00_cell101	1313600	CELL01
o/192.168.12.3/DATA_CD_01_cell101	1313600	CELL01
o/192.168.12.3/DATA_CD_02_cell101	1313600	CELL01
...		

# Creating Grid Disks

## There are two ways to create grid disks

1. Create a single grid disk by **fully qualifying** the name.

```
CellCLI> create griddisk TEST_DG_CD_00_cell103 -  
          celldisk='CD_00_cell103', size=100M
```

```
GridDisk TEST_DG_CD_00_cell103 successfully created
```

```
CellCLI> list griddisk attributes name, celldisk, size -  
          where name='TEST_DG_CD_00_cell103'  
TEST_DG_CD_00_cell103      CD_00_cell103      96M
```

# Creating Grid Disks

## There are two ways to create grid disks

### 2. Create a collection of grid disks

- using the **'all'** parameter
- specify a name **'prefix'**
- Grid disks are named as follows:

*{prefix}\_{celldisk\_name}*

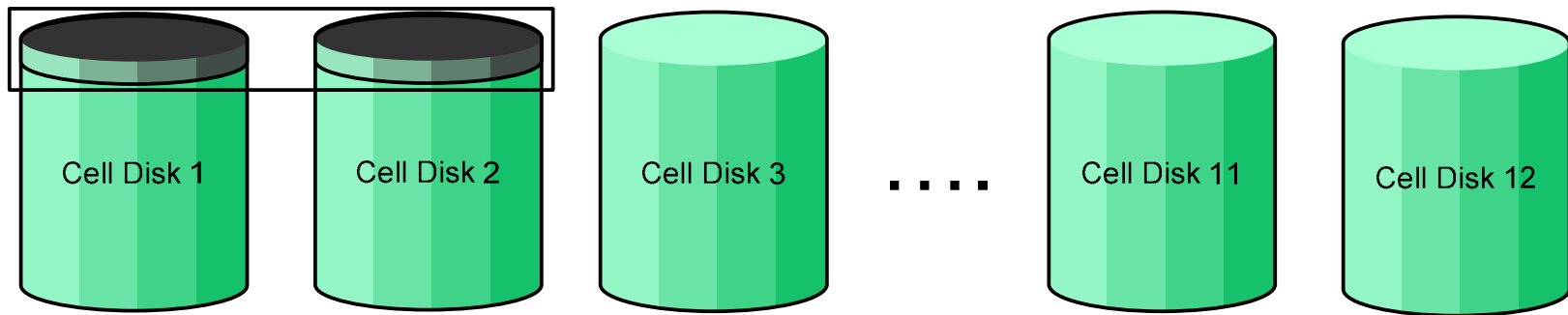
```
CellCLI> create griddisk all harddisk prefix='TEST', size=100M
GridDisk TEST_CD_00_cell103 successfully created
GridDisk TEST_CD_01_cell103 successfully created
...
GridDisk TEST_CD_10_cell103 successfully created
GridDisk TEST_CD_11_cell103 successfully created
```

```
CellCLI> list griddisk attributes name, cellDisk, diskType, size -
      where name like 'TEST_.*'
TEST_CD_00_cell103      FD_00_cell103      HardDisk      96M
...
TEST_CD_10_cell103      FD_10_cell103      HardDisk      96M
TEST_CD_11_cell103      FD_11_cell103      HardDisk      96M
```

# Grid Disk Sizing

## Dealing with the 'short disks'

OS Partitions



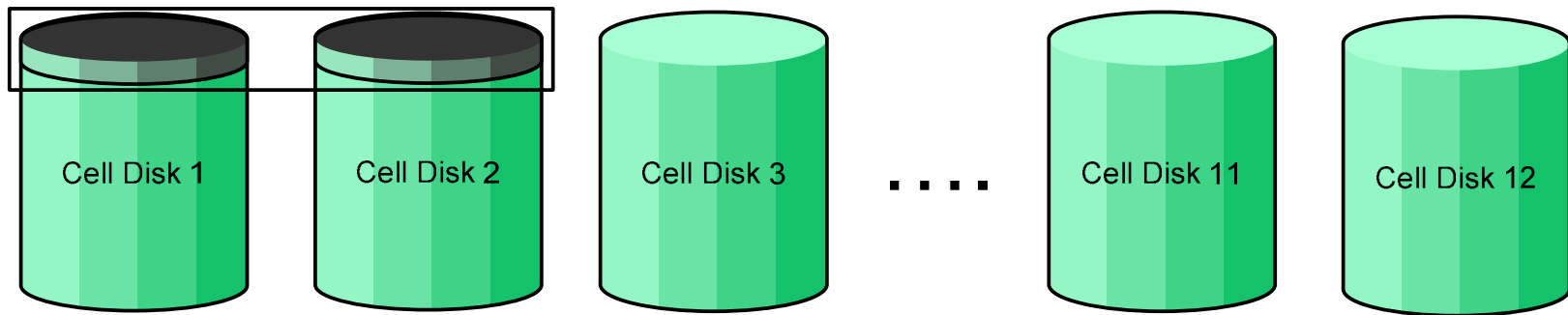
```
[enkcel103:root] /root  
> fdisk -lu /dev/sda
```

```
...  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sda1    *           63       240974     120456   fd  Linux raid autodetect  
/dev/sda2                240975     257039       8032+   83  Linux  
/dev/sda3                257040   3843486989 1921614975 83  Linux  
/dev/sda4          3843486990 3904293014   30403012+  f   W95 Ext'd (LBA)  
/dev/sda5          3843487053 3864451814   10482381   fd  Linux raid autodetect  
...  
/dev/sda10        3897995598 3899425319     714861   fd  Linux raid autodetect  
/dev/sda11        3899425383 3904293014     2433816   fd  Linux raid autodetect
```

# Grid Disk Sizing

## Dealing with the 'short disks'

OS Partitions



```
CellCLI> list celldisk attributes name, devicePartition, size -  
          where diskType = 'HardDisk'
```

```
CD_00_cell103    /dev/sda3      1832.59375G  
CD_01_cell103    /dev/sdb3      1832.59375G  
CD_02_cell103    /dev/sdc       1861.703125G  
CD_03_cell103    /dev/sdd       1861.703125G  
...  
CD_07_cell103    /dev/sdh       1861.703125G  
CD_08_cell103    /dev/sdi       1861.703125G  
CD_09_cell103    /dev/sdj       1861.703125G  
CD_10_cell103    /dev/sdk       1861.703125G  
CD_11_cell103    /dev/sdl       1861.703125G
```

# Grid Disk Sizing

- The 'size' parameter determines the size of each grid disk.
- If no size is specified then **all remaining cell disk space is used** for the grid disk.

```
CellCLI> create griddisk all prefix='RECO_DG'  
GridDisk RECO_DG_CD_00_cell103 successfully created  
GridDisk RECO_DG_CD_01_cell103 successfully created  
...  
GridDisk RECO_DG_CD_10_cell103 successfully created  
GridDisk RECO_DG_CD_11_cell103 successfully created
```

```
CellCLI> list griddisk attributes name, cellDisk, diskType, size where name  
like 'RECO.*'
```

<b>RECO_CD_00_cell101</b>	<b>CD_00_cell101</b>	<b>HardDisk</b>	<b>91.265625G</b>
<b>RECO_CD_01_cell101</b>	<b>CD_01_cell101</b>	<b>HardDisk</b>	<b>91.265625G</b>
RECO_CD_02_cell101	CD_02_cell101	HardDisk	120.375G
RECO_CD_03_cell101	CD_03_cell101	HardDisk	120.375G
RECO_CD_04_cell101	CD_04_cell101	HardDisk	120.375G
...			

# Storage Cell Provisioning

- Storage Cells may be allocated to specific database servers, creating multiple storage grids
- Multiple RAC Clusters
  - Production Cluster
  - Test Cluster
  - Dev Cluster
- Caution – This reduces IO bandwidth
  - Storage Cells need not be carved up to support multiple database servers.

# The Cellinit.ora File

/opt/oracle/cell11.2.2.2.0\_LINUX.X64\_101206.2/cellsrv/deploy/config/cellinit.ora

```
#CELL Initialization Parameters
```

```
version=0.0
```

```
DEPLOYED=TRUE
```

```
SSL_PORT=23943
```

```
JMS_PORT=9127
```

```
ipaddress1=192.168.12.9/24
```

```
HTTP_PORT=8888
```

```
BMC_SNMP_PORT=162
```

```
RMI_PORT=23791
```

←-- Storage Grid address of  
the storage cell.  
(on the Infiniband switch)

# The Cellip.ora File

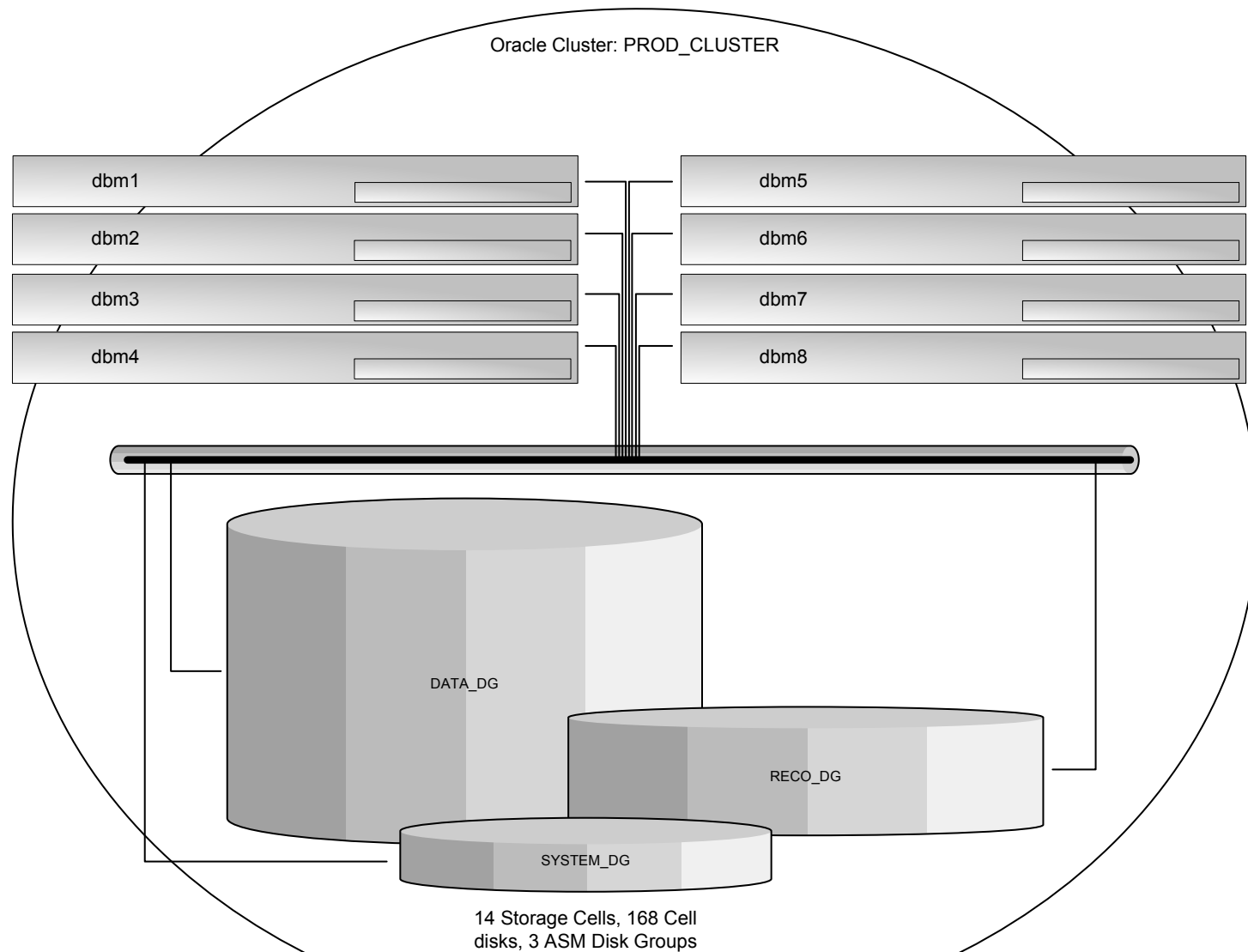
/etc/oracle/cell/network-config/cellip.ora

<b>Production Database Servers, 1-6</b>	<b>Production Storage Cells, 1-11</b>	
cell="192.168.12.9"	dm01cel01	192.168.12.9
cell="192.168.12.10"	dm01cel02	192.168.12.10
cell="192.168.12.11"	dm01cel03	192.168.12.11
cell="192.168.12.12"	dm01cel04	192.168.12.12
cell="192.168.12.13"	dm01cel05	192.168.12.13
cell="192.168.12.14"	dm01cel06	192.168.12.14
cell="192.168.12.15"	dm01cel07	192.168.12.15
cell="192.168.12.16"	dm01cel08	192.168.12.16
cell="192.168.12.17"	dm01cel09	192.168.12.17
cell="192.168.12.18"	dm01cel10	192.168.12.18
cell="192.168.12.19"	dm01cel11	192.168.12.19

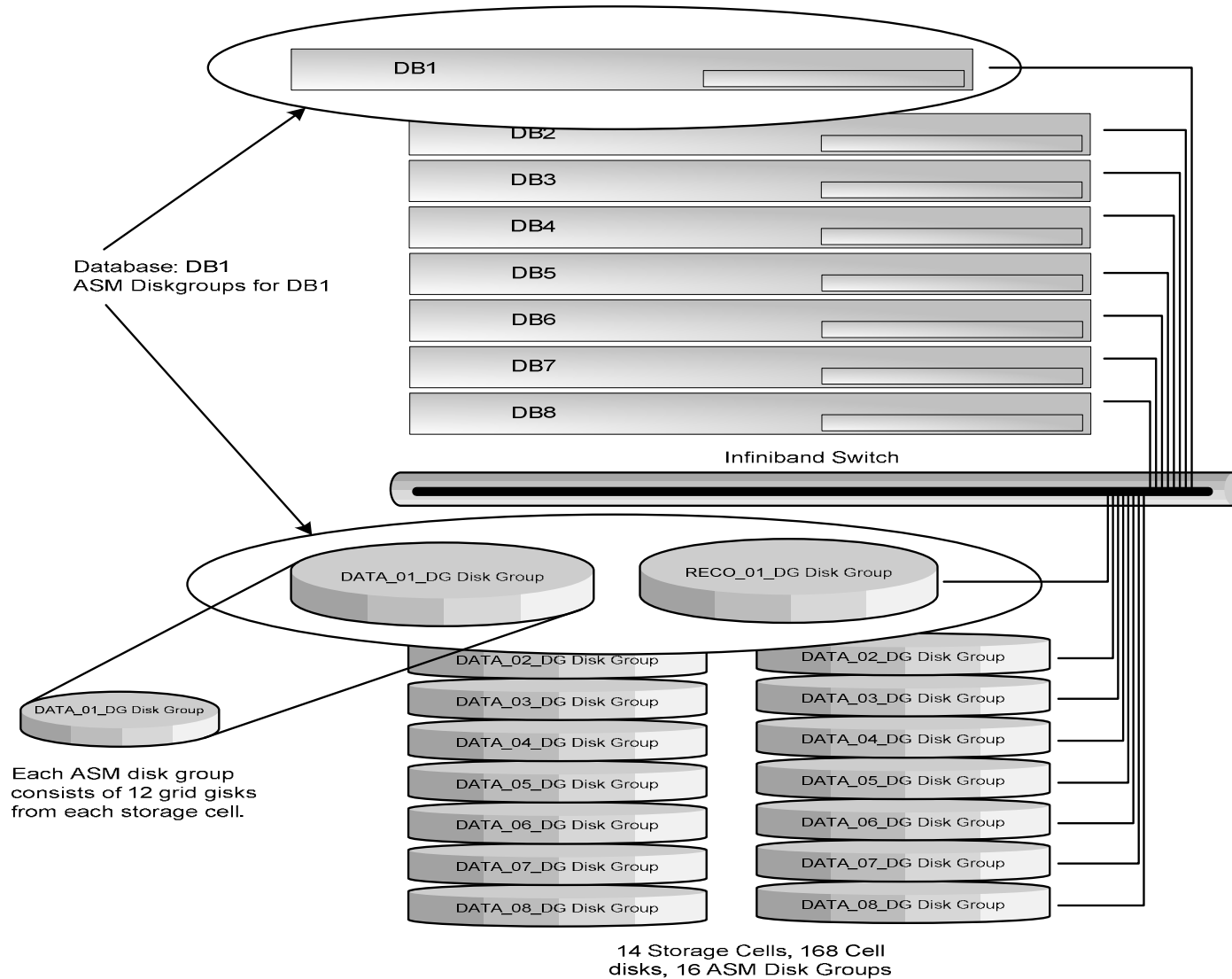
  

<b>Test Database Servers, 7-8</b>	<b>Test Storage Cells, 12-14</b>	
cell="192.168.12.20"	dm01cel12	192.168.12.20
cell="192.168.12.21"	dm01cel13	192.168.12.21
cell="192.168.12.22"	dm01cel14	192.168.12.22

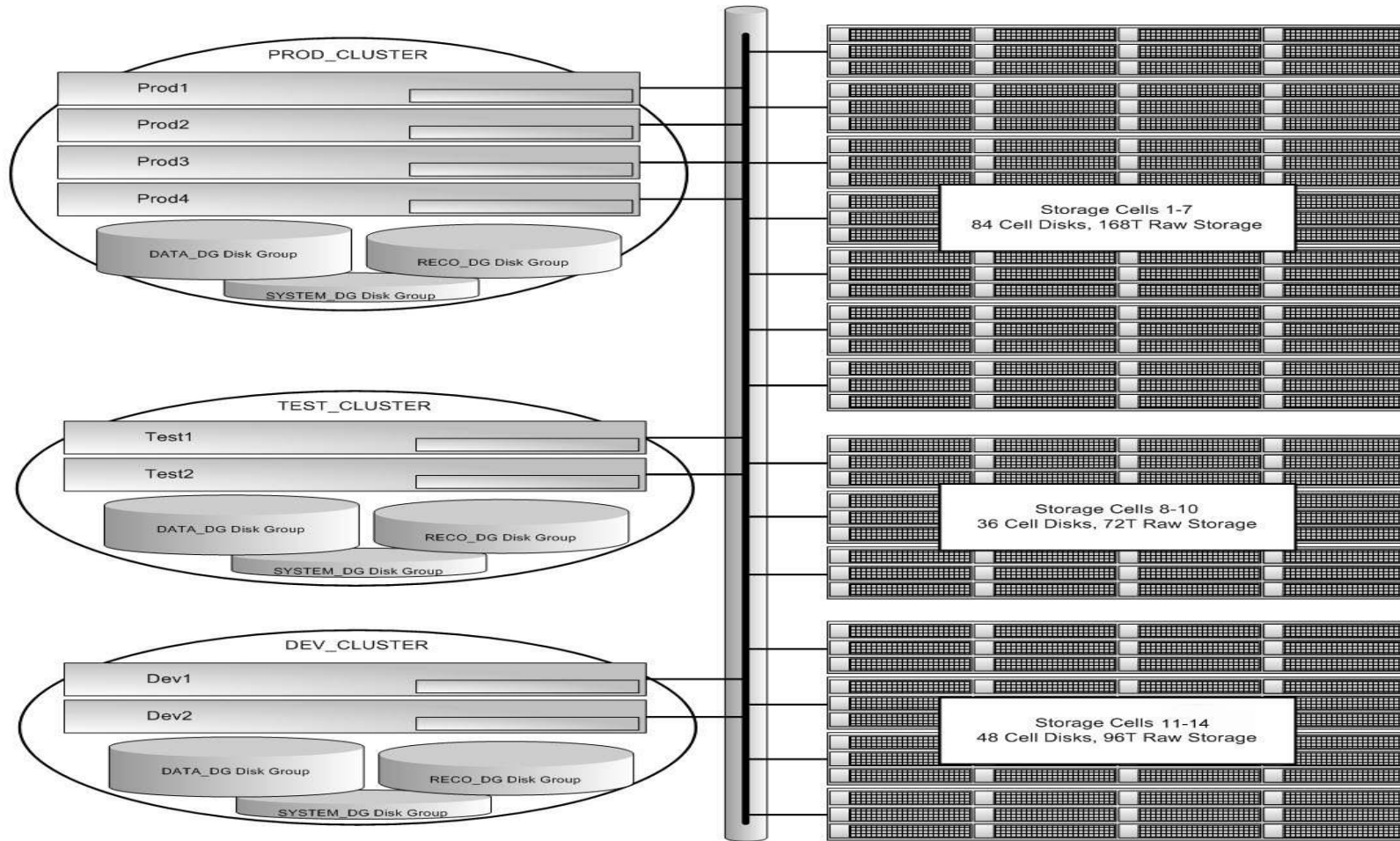
# Exadata Full Rack: Default Configuration



# Exadata Full Rack: 8 non-RAC Databases

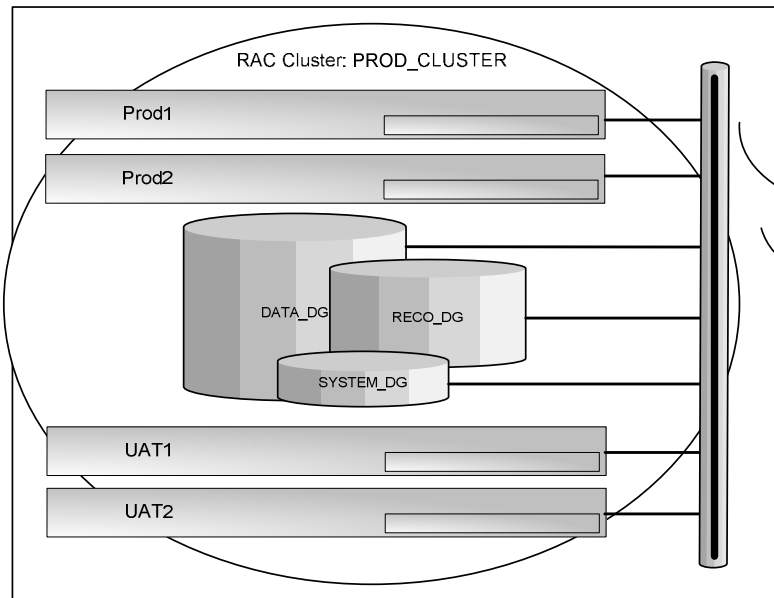


# Exadata Full Rack: 3 RAC Clusters



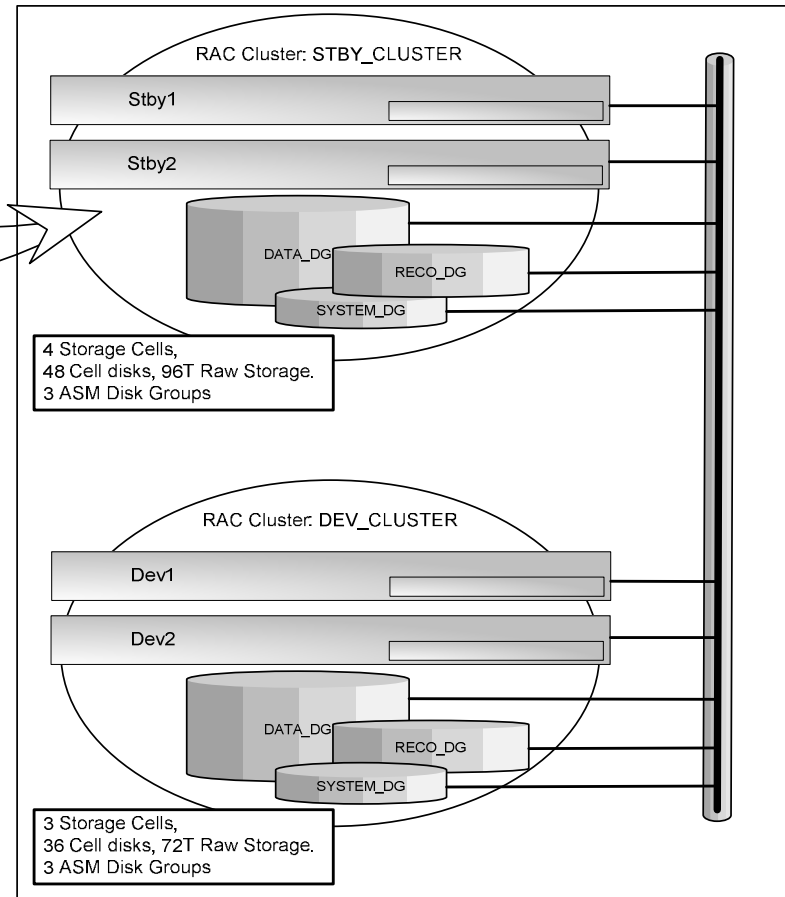
# Real World Example: 2 Exadata Half Racks

## Exadata Rack #1



Half Rack with 1 RAC Cluster  
7 Storage Cells,  
84 Cell disks, 168T Raw  
Storage. 3 ASM Disk Groups  
Shared by Prod and UAT  
databases.

## Exadata Rack #2



4 Storage Cells,  
48 Cell disks, 96T Raw Storage.  
3 ASM Disk Groups

3 Storage Cells,  
36 Cell disks, 72T Raw Storage.  
3 ASM Disk Groups

Half Rack with 2 RAC Clusters  
4, 3 Storage Cells respectively,  
Disk Groups not shared between  
Stby & Dev

# *Demo*

Q&A

*[randy.johnson@enkitec.com](mailto:randy.johnson@enkitec.com)*