

Oracle Exadata Storage Server Software Architecture



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

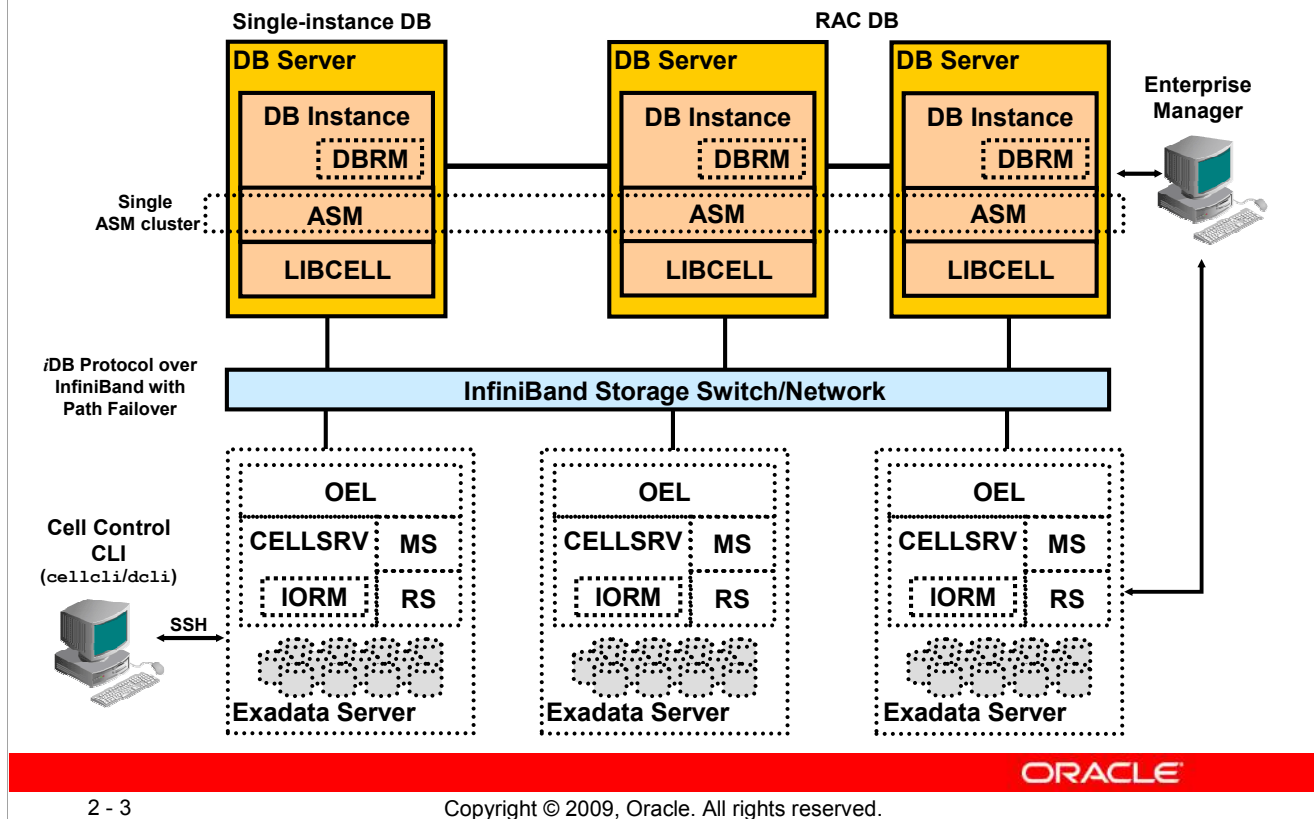
Objectives

After completing this lesson, you should be able to:

- Explain how Exadata Storage Server works
- Describe how storage security is implemented
- Define the relationships between the various storage abstractions

ORACLE®

Exadata Software Architecture Overview



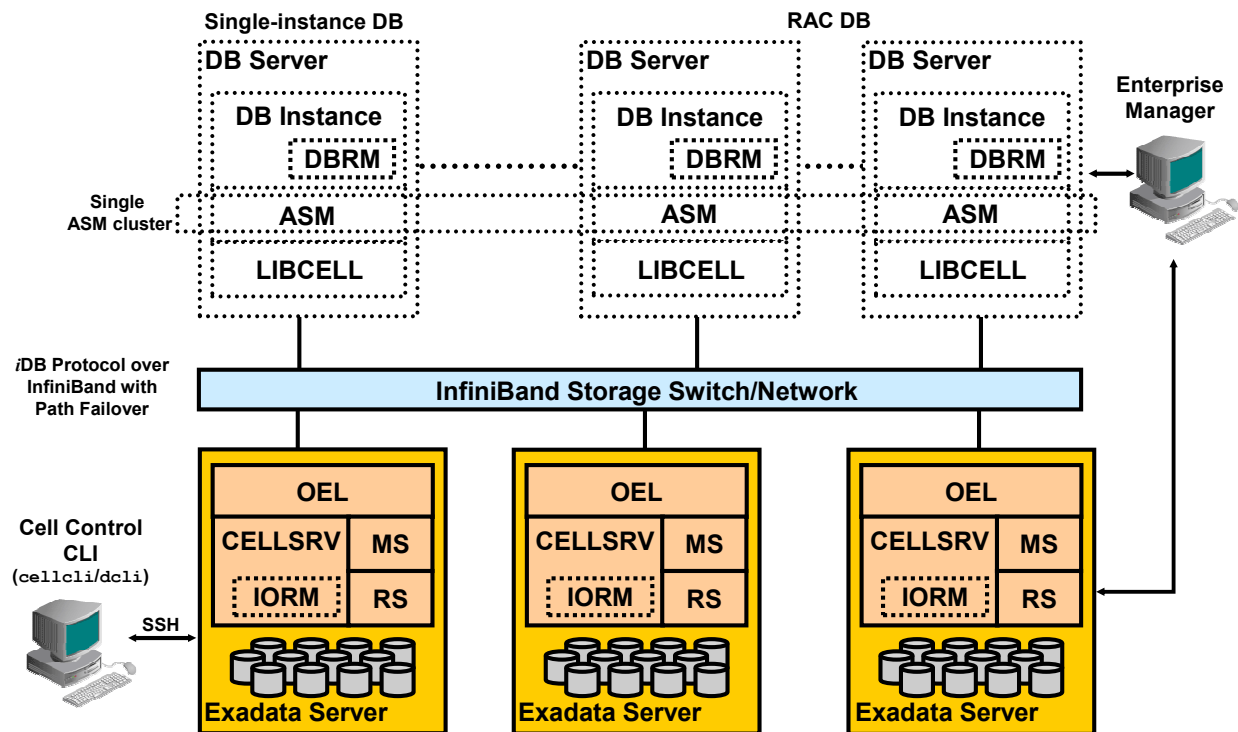
Exadata Software Architecture Overview

The architecture of the Exadata solution includes components on the database server and on the Exadata server. The overall architecture is shown in the slide. The components that reside on the database server include:

- The Intelligent Database protocol (*i*DB) is a unique Oracle data transfer protocol, built on RDS, that runs on industry standard InfiniBand networking hardware. *i*DB provides data intelligence between the database and Exadata cells and enables ASM and database instances to utilize Exadata-specific features, such as smart scan and I/O resource management. *i*DB is implemented in the database kernel and LIBCELL and transparently maps database operations to Exadata-enhanced operations. In addition, *i*DB provides InfiniBand Host Channel Adapter (HCA) bandwidth aggregation and failover, without relying on bonding drivers and expensive network switches. Single-instance or Oracle RAC databases access storage cells using *i*DB over an InfiniBand network.
- Automatic Storage Management (ASM) is required and provides a file system and volume manager optimized for the Oracle Database.
- Database resource management (DBRM), in combination with I/O resource management (IORM), ensures that I/O resources are allocated based on defined priorities.

Note: The slide illustrates the recommended configuration where a single ASM cluster is used to consolidate storage for all of your databases. Alternatively, you can connect multiple separate ASM environments with separate disk groups to Exadata.

Exadata Software Architecture Overview



2 - 4

Copyright © 2009, Oracle. All rights reserved.

ORACLE

Exadata Software Architecture Overview (continued)

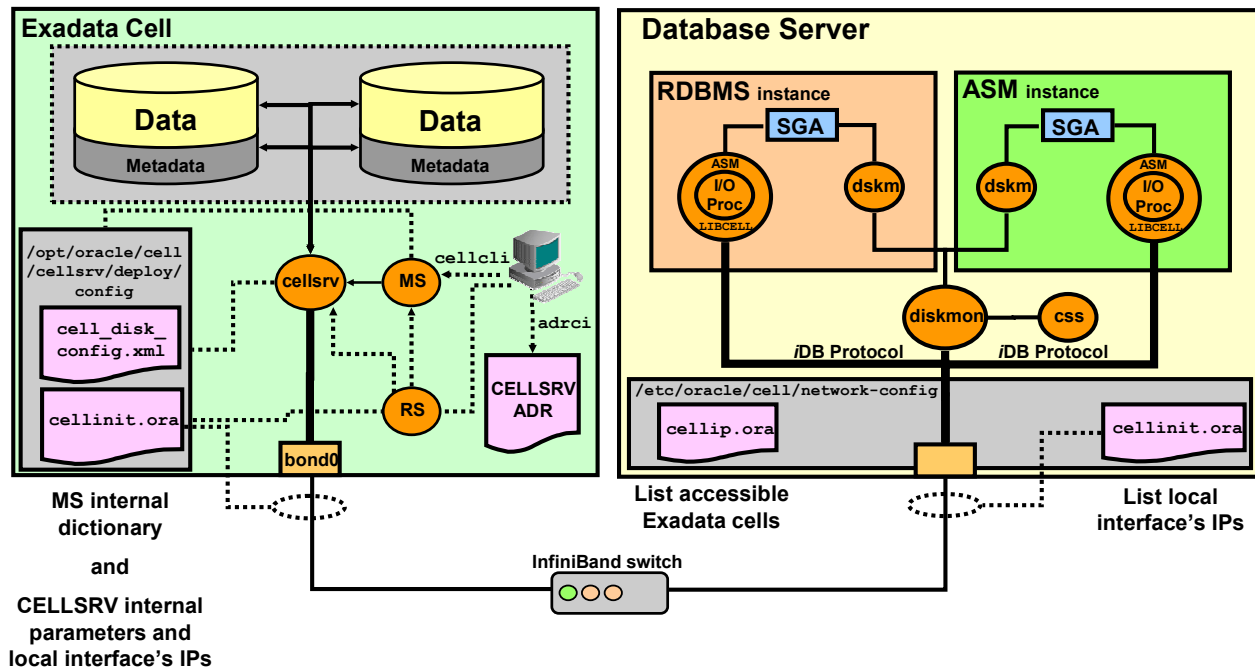
The software components that reside on the Exadata server include:

- Oracle Enterprise Linux (OEL) provides the Exadata server operating environment.
- Cell Server (CELLSRV) is the primary component of the Exadata software running in the cell and provides the majority of Exadata storage services. CELLSRV is a multithreaded server. CELLSRV clients use LIBCELL to communicate with it, and LIBCELL converts I/O requests into messages that are sent to CELLSRV using the iDB protocol. CELLSRV serves simple block requests, such as database buffer cache reads, and smart scan requests, such as table scans with projections and filters. CELLSRV also implements I/O resource management (IORM), which works in conjunction with the database resource manager (DBRM), to meter out I/O bandwidth to the various databases and consumer groups issuing I/Os. Finally, CELLSRV collects numerous statistics relating to its operations.
- Management Server (MS) provides standalone Exadata cell management and configuration. It works in cooperation with the Exadata cell command-line interface (CellCLI) and provides stand-alone Exadata cell management and configuration. Each cell in the Exadata storage grid is individually managed with CellCLI. However, you can run the same CellCLI commands remotely on multiple cells with the dcli utility. In addition, MS is responsible for sending alerts and collects some statistics in addition to those collected by CELLSRV.
- Restart Server (RS) is used to start up/shut down the CELLSRV and MS services and monitors these services to check whether they need to be restarted.

Exadata Software Architecture Overview (continued)

- Oracle Enterprise Manager Exadata Cell Plug-in enables you to monitor Exadata cell configuration and performance in a GUI display. In addition to reports, it enables you to set metric thresholds for alerts and monitor metric values to determine the health of a storage cell.

Exadata Software Architecture Details



2 - 6

Copyright © 2009, Oracle. All rights reserved.

ORACLE®

Exadata Software Architecture Details

Database-host side Exadata software:

- **LIBCELL Library:** Provides UNIX-like I/O primitives and is linked with ASM, RDBMS, and ASM utilities. It uses the *i*DB Protocol to communicate with the Exadata cell software.
- **DISKMON (Network/Cell Monitor):** Checks the network interface state and cell liveness. It uses a nodewide master process and one slave process (`diskm`) for each RDBMS or ASM instance. The master performs monitoring and propagates state information to the slaves. Slaves use the SGA to communicate with RDBMS or ASM processes. **DISKMON** performs DBRM plan propagation and also performs I/O fencing. Cluster Synchronization Services (CSS) still decides what to fence. Master **DISKMON** starts with the clusterware processes.

Cell-side Exadata software:

- CELLSRV is a multithreaded server which provides the majority of Exadata storage services. It provides smart storage capabilities, serves data blocks when offloading is not possible, and implements I/O resource management to meter out I/O bandwidth.
- Management Server (MS) is an OC4J application that provides storage cell management and configuration functions such as cell administration, and metrics and alerts generation. It also communicates with CELLSRV and the operating system.

Exadata Software Architecture Details (continued)

- Restart Server (RS): Monitors CELLSRV and MS Server and restarts them, if necessary.
- CellCLI: Executes user cell administration commands. The user must use SSH to connect to the cell and use CellCLI. It communicates with MS using Web Services.
- ADRCI: CELLSRV uses the Automatic Diagnostic Repository (ADR) to log software errors. An Exadata administrator may use the ADR viewer (ADRCI) to view and package ADR incidents.

InfiniBand fabric: InfiniBand provides a high-speed, high-bandwidth and low-latency network fabric to support Exadata. InfiniBand is the only network fabric supported for communication between Exadata cells and database servers.

Exadata requires the open source RDS/Open Fabrics Enterprise Distribution (OFED) be installed in the operating system used on the database server. The version of the OFED/RDS packages should be the same as that on the Exadata Storage Servers.

Oracle Unbreakable Linux customers can obtain the OFED/RDS packages through the Unbreakable Linux Network (ULN) for OEL5.3 and RHEL5U3.

Note: Red Hat does not bundle or support OFED/RDS in their current distributions. For Red Hat users who do not have Oracle Unbreakable Linux support, Oracle will provide the OFED/RDS packages. In addition, Oracle will make reasonable efforts to support RDS issues but this support will be limited only to RDS issues. Please refer to the My Oracle Support bulletin 745616.1 for more information.

Note: The only Oracle Database server platform that is supported in conjunction with Oracle Exadata Storage Server Software 11g Release 2 is Linux x86_64.

Note: Oracle Exadata Storage Server Software 11g Release 2 can only work with version 11.2.0.1 and higher of the Oracle Database and ASM.

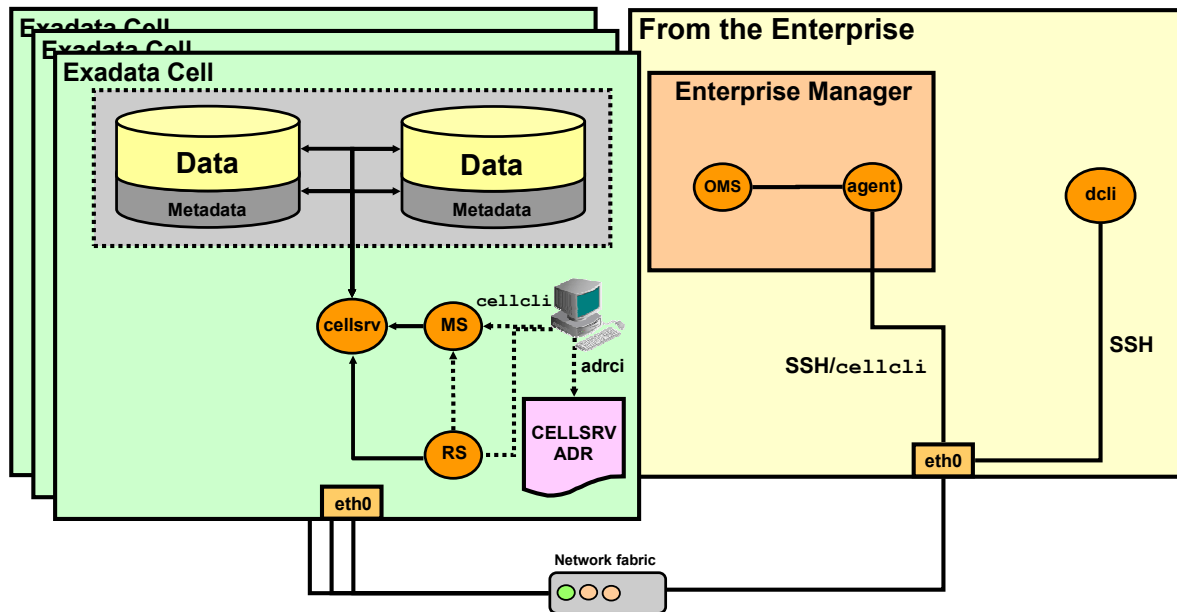
Exadata Smart Flash Cache Architecture

This is a placeholder.

Question for subject matter experts: Is there something worthwhile to say about the architecture of Exadata Smart Flash Cache or is it such a "black box" that it's not worth breaking down any further?

ORACLE

Exadata Monitoring Architecture



Exadata Monitoring Architecture

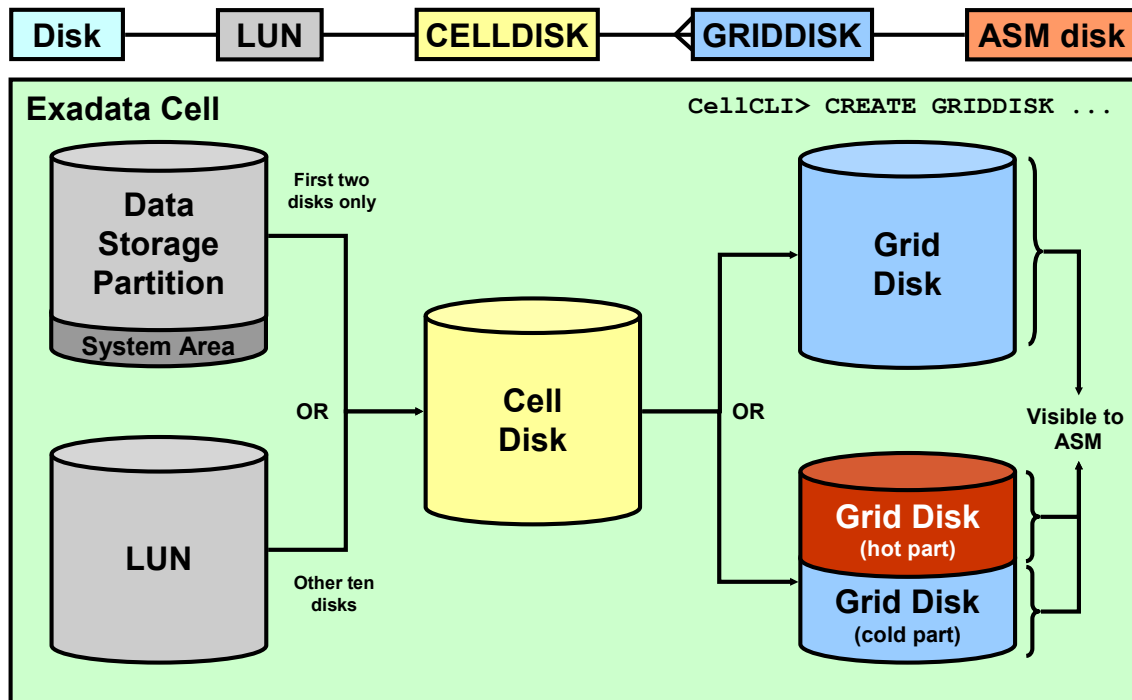
For monitoring, there is an Enterprise Manager plug-in that you use in conjunction with Grid Control. Using this plug-in you can monitor all the Exadata Storage Servers in your enterprise.

The Enterprise Manager plug-in for Exadata does not require an agent on each Exadata cell. Instead, the Enterprise Manager agent on the Grid Control server uses SSH to connect to each cell and execute `cellcli` commands. Using this architecture, monitoring information from numerous Exadata cells can be consolidated on to a single Enterprise Manager screen.

The `dcli` utility facilitates centralized management across an Exadata realm by automating the execution of `cellcli` and other cell-level operating system commands on a set of cells and returning the output to the centralized management location running the `dcli` utility. The `dcli` utility runs commands on multiple cells in parallel threads. The cells are referenced by their domain name or IP address. Local files can be copied to cells and command scripts may be executed on cells using this tool. Finally, you can use the `dcli` utility to set up SSH user-equivalence to a cell or group of cells.

Note: `dcli` is a Python script that is available on any Exadata server. You can copy it to your designated central management console and execute it from there. The `dcli` utility requires Python version 2.3 or later.

Disk Storage Entities and Relationships



Disk Storage Entities and Relationships

Each Exadata cell contains 12 physical disks. On each of the first two disks, Exadata reserves a system area that spans four disk partitions with a total size of 29GB. The system area contains the OS image, swap space, Exadata software binaries, metric and alert repository and various other configuration and metadata files. The two system areas are a mirror copy of each other that is maintained via software mirroring.

To the cell OS, each physical disk is represented as a unit of storage called a Logical Unit (LUN). A LUN is the lowest level of storage abstraction visible to the Cell Server software. You cannot create or remove a LUN, it exists because of the presence of the corresponding physical disk.

A cell disk is a higher level abstraction for the data storage area on a physical disk. A cell disk is based on a LUN and contains additional attributes. For the two disks that contain the system areas, the Exadata software recognizes the way that the LUN is partitioned and maps the cell disk to the disk partition reserved for data storage. For the other 10 disks, the Exadata software maps the cell disk directly to the LUN. After a cell disk is created, it can be further subdivided into grid disks, which are directly exposed to the database and Oracle ASM instances.

Disk Storage Entities and Relationships (continued)

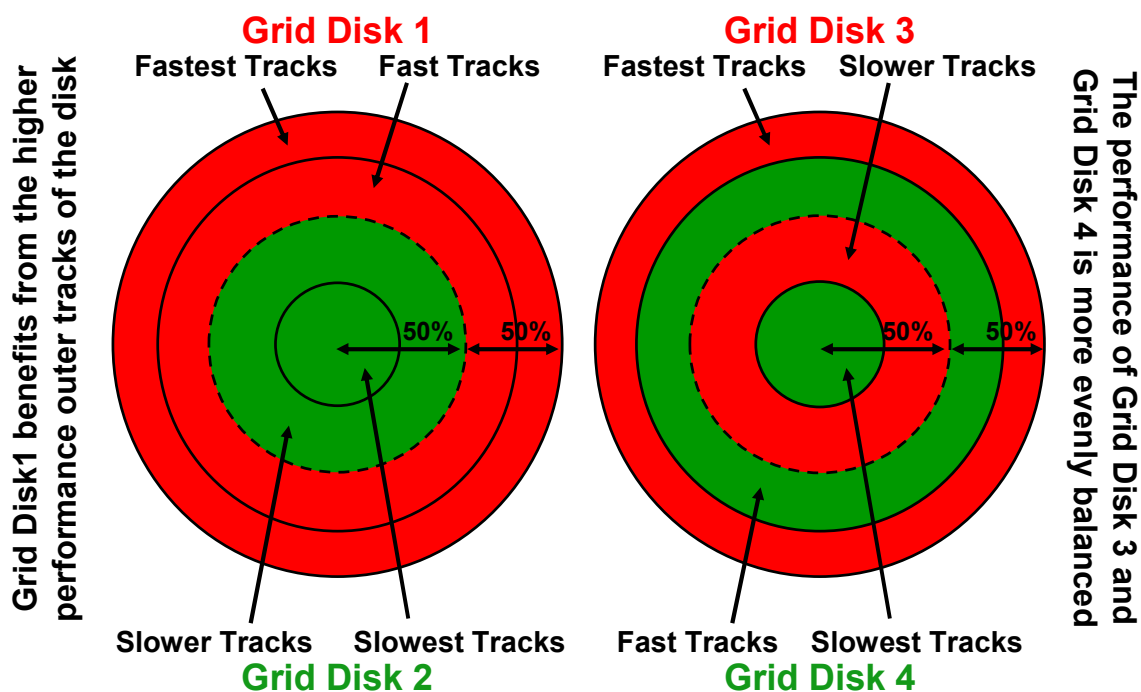
Placing multiple grid disks on a cell disk allows the administrator to segregate the storage into pools with different performance characteristics. For example, a cell disk could be partitioned so that one grid disk resides on the highest performing portion of the disk (the outermost tracks on the physical disk), whereas a second grid disk might be configured on the lower performing portion of the disk (the inner tracks). The first grid disk might then be used in an ASM disk group that houses highly active (hot) data while the second grid disk might be used to store less active (cold) data files.

Placing multiple grid disks on a cell disk also allows the administrator to segregate the storage into pools that can be assigned to different databases.

In cases where the entire cell capacity is required for a single database or where it is difficult to clearly define hot and cold data sets, an Exadata administrator will usually define a single grid disk containing all the space on each cell disk.

Note: The diagram on the slide shows the cases where one or two grid disks are created from the space on a cell disk. However, you can create more than two grid disks from a cell disk.

Interleaved Grid Disks



ORACLE

Interleaved Grid Disks

By default, space for grid disks is allocated from the outer tracks down to the inner tracks of a physical disk. However, space for grid disks can be allocated in an interleaved manner. Grid disks that use this type of space allocation are referred to as interleaved grid disks. This method attempts to equalize the performance of the grid disks residing on the same physical disk.

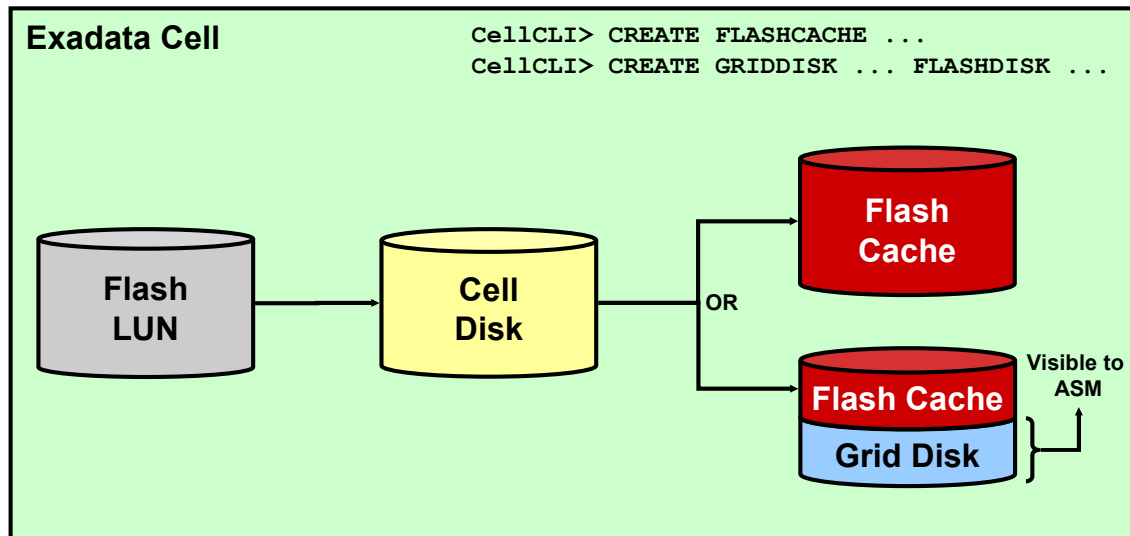
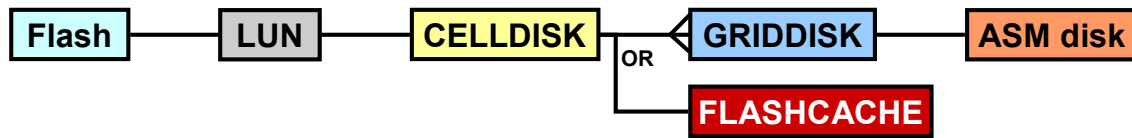
The slide contrasts default grid disk allocation with interleaved grid disks. On the left, two grid disks have been created on a physical disk using default space allocation. In this case Grid Disk 1 occupies all the fastest (outer) tracks while Grid disk 2 occupies all the slower (inner) tracks.

On the right you see an example of interleaved grid disks. With interleaving enabled, a disk is divided into two equal parts, the outer half (upper portion) and the inner half (lower portion). When a new grid disk is created, half of the grid disk space is allocated on the upper portion, and the other half of the grid disk space is allocated on the lower portion.

Interleaved grid disks are best used in situations where you want to create separate ASM disk groups that share cell disks without a performance bias.

Note that interleaving is enabled by setting the `INTERLEAVING` attribute for the cell disk.

Flash Storage Entities and Relationships



ORACLE

2 - 13

Copyright © 2009, Oracle. All rights reserved.

Flash Storage Entities and Relationships

Each Sun Oracle Exadata Storage Server contains 384 GB of high performance flash memory distributed across 4 PCI flash memory cards. Each card has 4 flash devices for a total of 16 on each cell. Each flash device has a capacity of 24 GB.

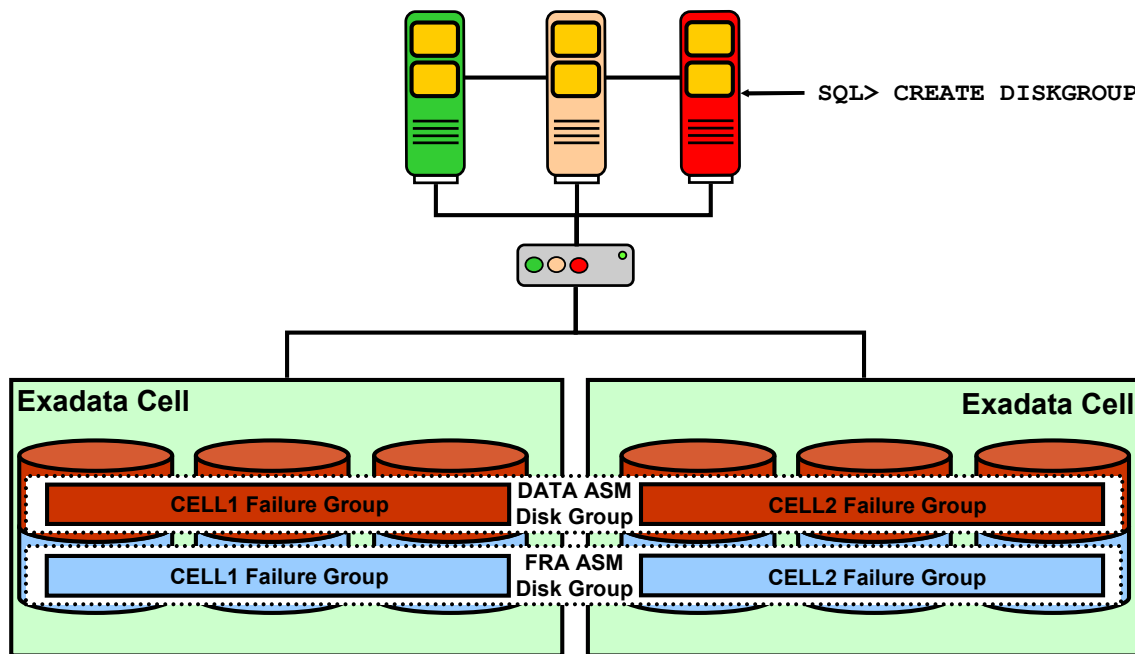
Essentially, each flash device is much like a physical disk in the Exadata storage hierarchy. Each flash device is visible to the Cell Server software as a LUN. You can create a cell disk based on a flash-based LUN. You can then create numerous grid disks on each flash-based cell disk. In addition, space on a flash-based cell disk can be allocated to a special area that supports Exadata Smart Flash Cache.

By default, the initial cell configuration process creates flash-based cell disks on all the flash devices, and then allocates all the available flash space to Exadata Smart Flash Cache. To create space for flash-based grid disks you would need to drop the default flash cache. Then you can create a flash cache and flash-based grid disks with your chosen sizes.

Unlike physical disk devices, the order in which you allocate your flash space is not important from a performance perspective. Likewise, interleaving is not applicable for flash-based cell disks.

Note: The diagram on the slide shows the case where a flash-based cell disk is allocated entirely to flash cache, and the case where a flash-based cell disk is used for flash cache and one grid disk. However, you can allocate up to one flash cache area along with zero or more flash-based grid disks from a flash-based cell disk.

Disk Group Configuration



Disk Group Configuration

After the grid disks are configured, ASM disk groups can be defined across your Exadata configuration.

The slide illustrates an example where two ASM disk groups are defined. The DATA disk group is defined across all the *hot* grid disks, and the FRA disk group is defined across the *cold* grid disks. When data is loaded into each disk group, ASM will evenly distribute the data and I/O across the grid disks in each disk group.

To protect against the failure of an entire Exadata cell, ASM failure groups are automatically defined on a per cell basis. This is to ensure that mirrored ASM extents are placed on different Exadata cells. This is also illustrated in the slide. By default when failure groups are automatically created, their names correspond to the cell name. So, different disk groups can have the same failure group names. When using Exadata, it is strongly recommended to use at least NORMAL ASM redundancy for all of your disk groups in conjunction with ASM failure groups spread across at least two Exadata cells. Following this recommendation provides good protection against disk and cell failure.

Using HIGH ASM redundancy in conjunction with ASM failure groups spread across at least three Exadata cells provides the best available level of data protection. Such a configuration can tolerate the simultaneous failure of two complete cells without compromising data availability.

Quiz

What are the three main Exadata cell processes?

1. OMS
2. MS
3. GMON
4. CELLSRV
5. RS

ORACLE®

2 - 15

Copyright © 2009, Oracle. All rights reserved.

Answers: 2, 4, 5

Quiz

If you use `NORMAL` ASM redundancy for all of your disk groups in conjunction with ASM failure groups spread across two Exadata cells, under which of the following scenarios will you maintain data availability?

1. A single disk failure in a single cell
2. Simultaneous failure of multiple disks in a single cell
3. Simultaneous failure of a single disk in both cells
4. Complete failure of a single cell

ORACLE

2 - 16

Copyright © 2009, Oracle. All rights reserved.

Answers: 1 ,2, 4

The prescribed configuration may provide protection against failure scenario 3 if, and only if, there are no data extents mirrored to both of the failed disks. To guarantee data availability in cases where simultaneous failures affect two cells you must use `HIGH` ASM redundancy in conjunction with failure groups spread across at least 3 Exadata cells.

Summary

In this lesson, you should have learned:

- Explain how Exadata Storage Server works
- Describe how storage security is implemented
- Define the relationships between the various storage abstractions

ORACLE

Demonstration Overview

The demonstrations associated with this lesson cover the following topics:

- A
- B
- C

ORACLE[®]