1

Oracle Exadata Exascale Overview

1.1 What is Oracle Exadata Exascale?

Oracle Exadata Exascale further empowers Exadata to meet the most demanding corporate and cloud computing requirements by decoupling Oracle Database and Oracle Grid Infrastructure clusters from the underlying Exadata storage servers. With Exascale, you can centrally manage a large fleet of Exadata storage servers connected by the Exadata RDMA Network Fabric. This architecture allows multiple Oracle Grid Infrastructure clusters and databases to securely access a large pool of shared storage resources, providing:

- Secure storage sharing with strict data isolation, ensuring users and databases can access only the data for which they have the proper privileges
- Flexible, dynamic storage provisioning for many users and databases
- Increased storage utilization and efficiency, maximizing your return on investment
- Improved utilization of storage processing resources to boost overall performance

Furthermore, Exascale introduces advanced snapshot and cloning capabilities that are tightly integrated with Oracle Database. For example, Oracle Database provides native snapshot and cloning functionality for pluggable databases (PDBs) through the CREATE PLUGGABLE DATABASE and ALTER PLUGGABLE DATABASE SQL commands. When Oracle Database utilizes Exascale storage, the pluggable database snapshot and snapshot copy (cloning) functions automatically use native Exascale snapshots and clones, which are space-efficient file copies based directly on the underlying Oracle Database files, thereby eliminating the need for a test master database to support snapshots and clones on Exadata.

In addition to unparalleled support for Oracle Database, Exascale provides block storage services, which deliver sophisticated capabilities to create and manage arbitrary-sized raw block volumes based on Exascale storage.

While end users can create and use Exascale block volumes for numerous applications, Exadata also leverages Exascale block volumes internally to store Exadata database server virtual machine images. Placing virtual machine images in Exascale removes the dependency on local storage inside the Exadata compute nodes, which enables the creation of more virtual machines and provides the infrastructure to support seamless migration of virtual machines between different Exadata compute nodes.

Despite the fact that Exascale transforms Exadata storage, Exascale also preserves the proven strengths and benefits of Exadata:

- Scalability - including efficient support for hundreds of Exadata storage servers in an Exascale cluster
- High availability - based on a clustered architecture with built-in redundancy and dynamic fail-over of software services
- High performance - utilizing Exadata RDMA memory and Exadata Smart Flash Cache

- Reliability - using proven Exadata storage server technologies
- Security - employing advanced security protocols and automatic encryption

Exascale runs on 2-socket Oracle Exadata system hardware with RoCE Network Fabric (X8M-2 or later). For full-featured native Oracle Database file storage in Exascale, you must use Oracle Database 23ai release 23.5.0 or later. You can also employ Exascale block volumes to support databases using older Oracle Database software releases back to Oracle Database 19c.

1.2 Exascale Components and Concepts

This section describes key Exascale components and concepts:

- Exascale Services
- Exascale Resources and Attributes
- Storage Media Types
- Pool Disks
- Storage Pools
- Vaults
- Files
- File Storage Attributes
- Templates
- Extended Attributes
- Clone Files
- Snapshot Files
- Datasets
- Exascale Users
- Exascale User Credentials
- User Privileges
- Access Control Lists
- Vault and File Access Control
- Trust Store
- Resource Management
- Exascale Block Store
- Features

1.2.1 Exascale Services

Exascale system components are implemented using a set of software services in a clustered architecture. The core Exascale storage services predominantly run on the Exadata storage servers.

Exascale contains the following storage services:

- **Cluster Services**

Exascale cluster services, also known as Exascale global services (EGS), provide the core foundation for the Exascale system. EGS primarily manages the storage allocated to Exascale storage pools. It also manages storage cluster membership, provides security and identity services for storage servers and Exascale clients, and monitors the other Exascale services. Exascale cluster services use the Raft consensus algorithm.

For high availability, every Exascale cluster contains five EGS instances.

For Exascale clusters with five or more Exadata storage servers, one EGS instance runs on each of five storage servers.

For Exascale configurations with fewer than five storage servers, one EGS instance runs on each Exadata storage server, and the remaining EGS instances run on the Exadata compute nodes to make up the required total of five. In a bare-metal configuration, EGS compute node instances run on the server operating system. In a configuration with compute nodes running on virtual machines (VMs), EGS compute node instances run in the hypervisor.

- **Control Services**

  Exascale control services, also known as Exascale RESTful Services (ERS), provide a management endpoint for Exascale management operations. All Exascale management operations come through ERS. But, no file I/O operations come through ERS.

  ERS service instances are deployed using front-end and back-end server processes. The front-end ERS processes provide a highly available client endpoint with load-balancing capabilities. The back-end ERS processes work with other software services to process requests and reply back to the client.

  Multiple ERS instances provide high availability and share the Exascale management workload. Typically, five ERS instances are distributed across the Exadata storage servers. However, for configurations with fewer than five storage servers, one ERS instance usually runs on each Exadata storage server.

  The Exascale Command Line (ESCLI) utility provides a simple command-line interface to perform Exascale monitoring and management functions. ESCLI commands are translated into ERS calls and run through ERS. ESCLI works in conjunction with the Exadata cell command-line interface (CellCLI) and does not replace it.

- **Exascale Vault Manager Services**

  Exascale vault manager, also known as Exascale data services (EDS), is the collective name for the software services that manage metadata for Exascale files and vaults:

  – **System Vault Manager**

    The system vault manager service (SYSEDS) serves and manages the metadata for Exascale vaults. This metadata includes vault-level access control lists (ACLs) and attributes.

    SYSEDS is a lightweight process, so one instance can usually service the load for an entire Exascale cluster. However, to ensure high availability, five SYSEDS instances are typically distributed across the Exadata storage servers. For configurations with fewer than five storage servers, one SYSEDS instance usually runs on each Exadata storage server.

  – **User Vault Manager**

    The user vault manager service (USREDS) serves and manages the metadata for files inside the Exascale vaults. This metadata includes file-level access control lists (ACLs) and attributes, along with metadata that defines clones and snapshots. All file control operations, such as open and close, are serviced by the user vault manager service.

Multiple USREDS instances provide high availability and share the user workload. Typically, five USREDS instances are distributed across the Exadata storage servers. However, for configurations with fewer than five storage servers, one USREDS instance usually runs on each Exadata storage server.

- **Block Store Manager**

  The block store manager service (BSM) serves and manages the metadata for Exascale block storage. All block store management operations are serviced by BSM. These block store management operations include creating a volume, creating a volume snapshot, and so on. It also coordinates the block store worker processes.

  To provide high availability, five BSM instances are typically distributed across the Exadata storage servers. However, for configurations with fewer than five storage servers, one BSM instance usually runs on each Exadata storage server.

- **Block Store Worker**

  The block store worker service (BSW) primarily services requests from block store clients and performs the resulting storage server I/O. It plays a role in clone and snapshot creation operations and is also responsible for performing volume backup and restore operations.

  To provide high availability and share the user workload, five BSW instances are typically distributed across the Exadata storage servers. However, for configurations with fewer than five storage servers, one BSW instance usually runs on each Exadata storage server.

- **Instant Failure Detection**

  The instant failure detection (IFD) service ensures high availability by instantly detecting failures across the entire Exascale cluster, thereby enabling quick recovery of critical Exascale services to minimize the impact of any service failure. IFD automatically runs as a dedicated lightweight service on every storage server associated with the Exascale cluster, along with any Exadata compute node running an Exascale global services (EGS) instance or the block store worker service (BSW).

  The instant failure detection (IFD) service is a dedicated lightweight service that quickly detects failures across the Exascale cluster. IFD automatically runs on every storage server associated with the Exascale cluster, along with any Exadata compute node running an Exascale global services (EGS) instance or the block store worker service (BSW).

- **Exadata Cell Services**

  Exascale works in conjunction with, and relies on, the core Exadata cell services. Specifically, Exascale requires running instances of Cell Server (CELLSRV), Management Server (MS), and Restart Server (RS) on every storage server. Also, Exascale requires running instances of Management Server (MS) and Restart Server (RS) on every compute server that is associated with the Exascale cluster.

In addition to the Exascale storage services, the following client-side services provide specific support for various Exascale functions on the Exadata compute nodes:

- **Exascale Node Proxy**

  The Exascale node proxy (ESNP) service maintains information about the current state of the Exascale cluster, which it provides to local Oracle Grid Infrastructure and Oracle Database processes.

  ESNP is a background process that runs on each Exadata compute node. In a bare-metal configuration, ESNP runs on the server operating system. In a configuration with compute nodes running on virtual machines (VMs), ESNP runs inside the guest VMs.

- **Exascale Direct Volume**

Exascale Direct Volume (EDV) is the default and recommended volume attachment mechanism within the Exadata RDMA Network Fabric.

The EDV service exposes Exascale volumes as EDV devices on Exadata compute nodes and services the I/O on each EDV device. EDV-managed storage can be used as raw block devices or to support various file systems, including Oracle Advanced Cluster File System (ACFS).

The EDV service is required on all Exadata compute nodes where you want to use EDV devices. In a bare-metal configuration, the EDV service runs on the server operating system. In a configuration with compute nodes running on virtual machines (VMs), the EDV service runs in the KVM host to provide the hypervisor access to VM image files based on EDV-managed storage. The EDV service also runs in each guest VM to enable direct access to EDV-managed storage from inside the VM.

Related Topics

- Administer the Exascale Cluster

1.2.2 Exascale Resources and Attributes

The metadata describing Exascale system objects is organized into a collection of resources. Exascale resources define the primary objects that Exascale users interact with, such as files, vaults, storage pools, volumes, and others.

Each resource contains a specific set of attributes. For example, every file has numerous attributes, including the file name, file size, and the name of the vault containing the file. Many attributes are automatically populated by the system, such as the file creation time. Some attributes, such as the size of a file, may be set during resource creation, while others, like the access control list (ACL), can be modified after creation.

At any point, users can utilize the Exascale Command Line (ESCLI) utility to view information about any Exascale resource and its attributes. See also Describing Resources and Attributes.

1.2.3 Storage Media Types

Exascale storage is categorized based on its underlying media type. The supported media types are:

- HC: Identifies high capacity storage, using hard disk drives (HDDs) on high-performance Exadata storage servers.
- EF: Identifies extreme flash storage, using low-latency, high-throughput flash devices.

1.2.4 Pool Disks

Exascale uses pool disks and storage pools to organize the physical storage provided by Exadata storage servers.

A pool disk is an Exascale-specific Exadata grid disk that reserves storage space for Exascale from an Exadata storage device.

Like any other Exadata grid disk, a pool disk is created by allocating space from an Exadata cell disk.

Based on its underlying physical characteristics, each pool disk is implicitly associated with a storage media type.

Related Topics

- Administer Exascale Pool Disks

1.2.5 Storage Pools

Exascale uses pool disks and storage pools to organize the physical storage provided by Exadata storage servers.

An Exascale storage pool is a collection of pool disks that provides persistent physical storage for Exascale vaults and files.

Each storage pool is a collection of pool disks with a common media type (for example, HC). You cannot define a storage pool with pool disks having a mixture of media types (for example, HC and EF).

A working Exascale system requires at least one storage pool. However, there is no need to define multiple storage pools for data separation because Exascale vaults provide strong data isolation.

A storage pool can contain pool disks that reside on different generations of Exadata storage server hardware, which allows for easy migration to new generations of Exadata hardware.

You can dynamically reconfigure a storage pool by changing the size of the pool disks or by adding or removing pool disks (or Exadata storage servers). However, you should pay attention to these recommendations:

- For each storage pool, use pool disks that are spread across all of the available storage devices in each storage server.
- On each storage server, maintain consistent sizing for all of the pool disks belonging to a storage pool. However, within a storage pool, pool disks on different storage servers can have different sizes.

Internally, a storage pool contains separate areas known as storage pool rings. Furthermore, the physical storage is arranged into smaller groups of pool disks and their associated storage servers, known as disk partner groups and cell partner groups.

The disk and cell partner groups maximize availability by limiting the potential effects of multiple simultaneous failures. This is because the probability of multiple simultaneous failures within a small group is much lower than the probability of multiple simultaneous failures anywhere across a large system. Also, simultaneous failures affecting different groups can be handled independently within each group.

Furthermore, Exascale uses separate storage pool rings for data and recovery files. The organization of the storage pool rings guarantees that data file extents and related recovery files automatically use different disk partner groups. So, in the unlikely event that multiple failures affect a data file, its corresponding recovery files remain available.

You cannot specify or change the internal organization of the storage pool rings or the cell and disk partner groups. However, Exascale provides information about these constructs, which is helpful for monitoring the system and understanding the impact of any storage failure scenario.

Related Topics

- Administer Exascale Storage Pools

1.2.6 Vaults

An Exascale vault is a logical storage container that uses the physical resources provided by Exascale storage pools.

By default, a vault can use all the underlying storage pool resources. However, an Exascale administrator can limit the amount of space, I/O resources (I/Os per second, or IOPS), and cache resources associated with each vault.

To an end-user and Oracle Database, a vault appears like a top-level directory that contains files. Exascale uses the convention of beginning vault names with the at sign (@) character. For example, @my-vault. So, a fully qualified Exascale file path always starts with the at sign (@) and vault name, followed by the rest of the file path. For example, @my-vault/my-example/my-file-name.

Exascale vaults facilitate strict data separation, ensuring that data is isolated to specific users and separated from other data and users. A vault, and its contents, are invisible to users without the appropriate privileges. Without the correct entitlements, users of one vault cannot see another vault, even though data from both vaults may be striped across the same underlying storage pools.

Related Topics

- Administer Exascale Vaults

1.2.7 Files

Exascale is optimized to directly store and manage files associated with Oracle Database and Oracle Grid Infrastructure.

When Oracle Database or Oracle Grid Infrastructure creates a file inside a vault, Exascale automatically recognizes the file type and applies the appropriate file storage attributes using a template associated with that file type. Among other things, the template defines the file's media type, which in turn determines the storage pool where the file is stored.

You can also use the Exascale Command Line (ESCLI) mkfile command to create a file. By this means, you can explicitly set the file storage attributes, or you can influence the file storage attributes by specifying the file type or applying a specific template. You can then use the ESCLI putfile command to copy the contents of a file into Exascale.

To optimize storage efficiency, physical space is consumed only when data is written to the file. For example, when you use the ESCLI mkfile command to create a file, the specified file size is reserved in the vault, but the file does not use any physical storage at that point. Physical storage is only allocated when data is written to the file.

Technically, you can store any type of files inside an Exascale vault. However, Exascale is optimized to manage Oracle Database files, which are typically much larger than regular files. To optimally store and manage regular files, you can define a block volume on Exascale, attach the volume to a server, and then use the volume attachment to support a file system. See Exascale Block Store.

Related Topics

- Administer Exascale Files

1.2.8 File Storage Attributes

During file creation, every file in Exascale is associated with file storage attributes, which govern how the file is stored and managed. The attributes are:

- mediaType: Specifies the physical media type used to store the file. Exascale uses this attribute to place the file in a storage pool that utilizes the specified media type. Possible values are:

  - HC: Identifies high capacity storage, using hard disk drives (HDDs) on high-performance Exadata storage servers.
  - EF: Identifies extreme flash storage, using low-latency, high-throughput flash devices.
- redundancy: Specifies the number of data copies that are maintained. Currently, the only permitted value is:
  - high: Indicates three mirrored copies of the file data.
- contentType: Specifies the type of content in the file. Exascale internally uses this attribute to place file extents on physically separate devices in a manner that maximizes availability in the event of a failure. Possible values are:
  - DATA: Principally associated with user data.
  - RECO: Primarily for data used in backup and recovery operations.

File storage attributes are assigned to all files when they are created. Typically, the file storage attributes are implicitly assigned using templates, which assign attribute settings based on the file type. File storage attributes can be assigned explicitly when a file is created by using the ESCLI mkfile command.

You cannot change the file storage attributes after the file is created.

1.2.9 Templates

During file creation, every file in Exascale is associated with attributes that govern how the file is stored and managed.

An Exascale template is a named collection of file storage attribute settings. For example:

```
name: DATAFILE
mediaType: HC
redundancy: high
contentType: DATA
```

With templates, you can set file storage attributes automatically and consistently. When creating a file, Exascale automatically uses the template that matches the file type by default. For example, when Oracle Database creates a data file, the DATAFILE template is automatically used by default.

Note that templates only govern the assignment of file storage attributes when creating a file. You cannot change the file storage attributes after the file is created. Any change to a template applies only to files created after the change.

Within Exascale, templates are defined at two levels:

- **Cluster Templates** are defined at the cluster level. A cluster template is used if no vault-level template exists to override it.
- **Vault Templates** are defined at the vault level. A vault template defines vault-specific attributes that override the corresponding cluster template. For example, your cluster-level DATAFILE template may specify the use of high capacity (HC) storage media, but you may override that setting for a vault by having a vault-level DATAFILE template that specifies the use of extreme flash (EF) storage.

Additionally, **User Templates** are templates with user-specified names that do not correspond to a specific file type. User templates effectively override file-type templates. However, to use a user template, you must explicitly specify it during file creation. User templates can be defined

at the cluster level or at the vault level. A vault-level user template will override a cluster-level user template having the same name.

Related Topics

- Administer Cluster Templates
- Administer Vault Templates

1.2.10 Extended Attributes

An extended attribute is non-standard metadata that is associated with an Exascale vault or file. An extended attribute consists of an extended attribute name and an associated value. The extended attribute name is a user-defined string value. The extended attribute value can be a string value or a binary value read from a file.

In addition to user-defined extended attributes, extended attributes reserved for internal use by Exascale services and tools have names starting with a dollar sign ($). You cannot create, modify, or delete these extended attributes.

Multiple extended attributes are allowed for each Exascale vault or file. For each file or vault, the total space available for all extended attributes is 16 kilobytes. This allowance includes the extended attribute names and values, along with a small allowance for internal metadata.

Related Topics

- Administer Extended Attributes

1.2.11 Clone Files

An Exascale clone is a thinly-provisioned point-in-time copy of a file that is readable and writable. The source file for a clone can be a regular file, a snapshot, or another clone. Exascale uses redirect-on-write techniques to create and maintain clones very quickly and space-efficiently.

The clone creation process can work on an individual file or a groups of files. All clones created in the same operation are point-in-time consistent, and all files in a clone operation must be in the same vault.

A clone file inherits the ACL and file storage attribute settings from the original file. After creation, you can modify the clone ACL and any modifiable attributes independently from the original file. You can also modify, or even delete, the original file without affecting the clone.

Typically, you will automatically use Exascale clones through the Oracle Database pluggable database (PDB) snapshot and cloning capabilities. For example, the Oracle Database CREATE PLUGGABLE DATABASE ... FROM ... command internally uses Exascale clones when the FROM database is on Exascale.

You can also use the ESCLI clonefile command to manually create clones, and you can use lssnapshots to view the association between clones and their sources.

Exascale also contains separate cloning capabilities for Exascale block volumes. See Volume Clones.

Related Topics

- Administer Exascale Files
- Cloning Files

1.2.12 Snapshot Files

An Exascale snapshot is a thinly-provisioned read-only point-in-time copy of a file. The source file for a snapshot can be a regular file, a clone, or another snapshot. Exascale uses redirect-on-write techniques to create and maintain snapshots very quickly and space-efficiently.

The snapshot creation process can work on an individual file or a groups of files. All snapshots created in the same operation are point-in-time consistent, and all files in a snapshot operation must be in the same vault.

A snapshot file inherits the ACL and file storage attribute settings from the original file. After creation, you can modify the snapshot ACL and any modifiable attributes independently from the original file. You can also modify, or even delete, the original file without affecting the snapshot.

You can use the ESCLI snapshotfile command to create snapshots, and you can use lssnapshots to view the association between snapshots and their sources.

While an Exascale snapshot is read-only, you can effectively manufacture a writable snapshot by creating a snapshot and then making a clone of the snapshot. In this case, the snapshot provides a durable point-in-time copy of the file, and the clone enables writing to a thinly-provisioned copy of the snapshot. At any point, you can effectively roll back to the snapshot creation time by removing and re-creating the clone.

Exascale also contains separate snapshotting capabilities for Exascale block volumes. See Volume Snapshots.

Related Topics

- Administer Exascale Files
- Snapshotting Files

1.2.13 Datasets

An Exascale dataset is a logical grouping of files within a vault. The primary purpose of a dataset is to enable tracking and management of storage utilization for files associated with Oracle Database and Oracle Grid Infrastructure.

Exascale automatically creates and maintains system-defined datasets, which are conceptually organized in a hierarchical tree. Each dataset directly contains some number of files and indirectly contains child datasets that correspond with other entities in the hierarchy. The hierarchy of system-defined datasets contains the following levels:

1. Exascale vault
2. Oracle Grid Infrastructure (GI) cluster
3. Oracle multitenant container database (CDB)
4. Oracle pluggable database (PDB)

Starting at the bottom, the files belonging to each PDB are contained in a separate dataset. The parent CDB dataset contains the files belonging to the CDB and all of the associated PDB datasets. All of the CDBs in a GI cluster as grouped in a GI-level dataset, which also contains GI-specific files such as the Oracle Cluster Registry (OCR) and voting files. At the top of the hierarchy, the vault-level dataset contains all of the GI-level datasets that consume storage in the vault. The vault-level dataset is also the default container for any files that don't belong to Oracle Database and Oracle Grid Infrastructure.

Each system-defined dataset is identified by a composite ID, which contains unique identifiers for the associated entities in the hierarchy. For example, each vault-level dataset is simply identified by the vault name (for example, @my-vault). But, a PDB dataset ID has the following format:

@Vault-name:GI-cluster-ID:CDB-ID:PDB-ID

Each system-defined dataset also has a name. The dataset name is also a composite of the relevant entities, but it contains human-readable names for the GI cluster, CDB and PDB components (instead of the system-generated unique identifiers). For example, a PDB dataset name has the following format:

@Vault-name/GI-cluster-name:CDB-name.PDB-name

Related Topics

- Administer Exascale Datasets

1.2.14 Exascale Users

Exascale has a system of user accounts enabling different users to perform actions and access data according to their assigned privileges. Though you can create a single Exascale user with privileges to do everything, a typical configuration contains cluster administration users and storage users:

- Cluster administration users are typically provisioned with privileges to administer the Exascale cluster. Cluster administration users typically administer the physical storage objects; namely storage servers, storage pools and pool disks. They also administer Exascale software services, vaults, cluster templates and user accounts. See Oracle Exadata Exascale System Administration.

  By default, each Exascale cluster contains one superuser account. The user identifier (ID) for the superuser account is admin. The admin user can implicitly perform any system operation and effectively holds all system privileges.

  While you can use the admin user to perform cluster administration tasks, Oracle recommends that you create your own cluster administration users with specific privileges. For example, rather than having one cluster administrator that does everything, you may choose to have dedicated user accounts for security administration, storage administration, and so on.
- Storage users are typically provisioned with privileges to use storage within Exascale vaults. Storage users often administer the vaults they use, and sometimes even create new vaults. Storage users typically manage their own files and the access control lists (ACLs) that govern file access. They also administer vault-level templates, extended file attributes, and their own user credentials. See Oracle Exadata Exascale User-Specific Administration.

Additionally, Exascale contains one node administration account for every node (storage server or compute node) that runs Exascale software services. Each node administration account inherits its user ID from the server hostname and each account contains the privileges required to run the Exascale software services on the node. Do not directly use or modify these accounts.

Related Topics

- Administer Exascale Users

1.2.15 Exascale User Credentials

Exascale user authentication uses public and private key pairs, with user credentials stored in a digital key store.

Each Exascale user is associated with a public key. To prove their identity and connect to Exascale, a user must supply the matching private key.

Exascale user credentials are contained in a digital key store, also known as a wallet. To use Exascale, a user's wallet must contain their private key. It must also contain a copy of the Exascale trust store. As a matter of convenience, a wallet can also store the default endpoint for Exascale control services.

To facilitate flexible key management, each Exascale user can be associated with up to three public and private key pairs. However, each wallet should contain only one Exascale user name and one private key.

For maximum security, an Exascale user should create their own public and private key pairs and manage their own wallet. This is recommended to ensure the integrity of the private keys, since the user should never share a private key, not even with the Exascale administrator.

Related Topics

- Administer Exascale User Credentials

1.2.16 User Privileges

User privileges control the actions performed by Exascale users.

Each Exascale user is subject to a set of user privileges, which govern the actions that the user is allowed to perform.

User privileges are assigned to users by using the ESCLI mkuser or chuser commands.

There are four types of Exascale user privileges, and any user may hold privileges across multiple privilege types. The following list describes the privilege types and the available user privileges:

- **Cluster Level Storage Privileges** primarily govern the administration actions that the receiving user is allowed to perform on storage resources in the Exascale cluster. Typically, cluster level storage privileges are only assigned to users that administer the Exascale cluster. A user may hold zero or one of the following cluster level storage privileges:
  - cl_monitor: Enables the receiving user to monitor the Exascale cluster by performing list operations using ESCLI and CellCLI.
  - cl_operator: Enables the receiving user to:
    - Monitor the Exascale cluster by performing list operations using ESCLI and CellCLI.
    - Manage pool disks (create, drop, online, offline).
    - Manage software services (list, startup, shutdown, restart, delete).
    - Manage the trust store.
  - cl_admin: A set of system administrator privileges that includes all the cl_monitor and cl_operator privileges, along with all of the privileges from the other privilege types; namely:

    - All the cluster level user privileges: user_create, system_restore, and on_behalf_of.
    - All of the vault top-level privileges specified in vlt_manage.
    - All the service privileges: cellsrv, egs, ers, syseds, usreds, bsm, and bsw.

    This privilege also enables the receiving user to:

    - Grant any privilege to any user.
    - Reset a key for any user.
    - Create and delete storage pools.
    - View extent map information.

- **Cluster Level User Privileges** govern the administration actions that the receiving user is allowed to perform on the Exascale cluster. Typically, cluster level user privileges are only assigned to users that administer the Exascale cluster. A user may hold zero or more of the following cluster level privileges:
    - user_create: Enables the receiving user to create new users in the cluster.
    - system_restore: Enables the receiving user to restore an Exascale backup.
    - on_behalf_of: A special privilege that enables the receiving user to send a request to Exascale control services (ERS) on behalf of another user.

      For example, consider a user that sends a request to ERS, which involves an action that must be performed by another Exascale service. In this case, ERS uses this privilege to forward the action to the other Exascale service on behalf of the original end user.

      Typically, this privilege is only assigned to the internal administration accounts that reside on each Exascale node.
- **Vault Top-Level Privileges** govern the actions that the receiving user is allowed to perform on all vaults and files. Typically, vault top-level privileges are assigned to users that use and manage files in Exascale vaults. A user may hold zero or one of the following vault top-level privileges:
    - vlt_inspect: Enables the receiving user to create new vaults. The receiving user also gets complete control over files created in those vaults. This privilege is assigned to new users by default.
    - vlt_read: Includes the vlt_inspect privileges and also enables the receiving user to list all existing vaults, display attributes for any vault, create files in any vault, list files in any vault, and display attributes for any file.
    - vlt_use: Includes the vlt_read privileges and also enables the receiving user to open any file for reading.
    - vlt_manage: Includes the vlt_use privileges and also enables the receiving user to open any file for read and write, alter any vault or file, and drop vaults and files.

    Vault top-level privileges work in addition to access control lists (ACLs). To perform an action on a vault or file, a user requires the appropriate vault top-level privilege or the appropriate ACL privilege. See Vault and File Access Control.
- **Service Privileges** govern the Exascale software services that the receiving user is allowed to run. Typically, service privileges are only assigned to the internal node-specific administration accounts that reside on each Exascale node. A user may hold zero or more of the following service privileges:
    - cellsrv: Enables the receiving user to run the core Exadata cell services.

- egs: Enables the receiving user to run Exascale cluster services (also known as Exascale Global Services).
- ers: Enables the receiving user to run Exascale control services (also known as Exascale RESTful Services).
- syseds: Enables the receiving user to run the system vault manager service.
- usreds: Enables the receiving user to run the user vault manager service.
- bsm: Enables the receiving user to run the block storage manager service.
- bsw: Enables the receiving user to run the block storage worker service.
- ms: Facilitates the transfer of telemetry information between Exascale RESTful Services (ERS) and the Exadata Management Server (MS).

Additionally, no_privilege is a special privilege that removes all privileges from the receiving user. When it is assigned to a user, no_privilege cannot be combined with any other privilege.

Related Topics

- Modify User Privileges

1.2.17 Access Control Lists

Access control lists (ACLs) govern the operations that users can perform on Exascale vaults and files.

Each Exascale vault or file has an ACL. A vault ACL enables users to perform actions on the vault and on the files that it contains. A file ACL only controls the file that it is associated with.

The following table lists the ACL privileges and the actions that they enable users to perform:

ACL Privilege	In a vault ACL, the ACL privilege enables the user to:	In a file ACL, the ACL privilege enables the user to:
inspect	• Create a file in the vault. • View attributes of the vault, but not the vault contents.	• View attributes of the file, but not the file contents.
read	• View attributes of all files in the vault, but not their contents. • Perform all inspect actions.	• Read the file contents. • Perform all inspect actions.
use	• Read the contents of all files in the vault. • Perform all inspect and read actions.	• Read and write the file contents. • Alter attributes of the file. • Perform all inspect and read actions.
manage	• Read and write the contents of any file in the vault. • Alter the attributes and ACL for the vault and any file in the vault. • Drop the vault and any file in the vault. • Perform all inspect, read and use actions.	• Alter the file ACL. • Drop the file. • Perform all inspect, read and use actions.

Note that the same ACL privilege enables different actions in a vault ACL or a file ACL. For example, in a file ACL the read privilege enables the user to read the contents of the file. However, to read file contents using a vault ACL requires the use privilege.

Every ACL is a list of user IDs and privilege pairs. Depending on the user creation method, the user ID may be a system-generated value or a user-specified value. For example:

```
96a68014-5762-4579-86ee-29eb743decbd:manage;scott:use;sue:inspect;dd7c8e35-3c8d-4441-a9b0-f58e959b84ba:read
```

A user is added to an ACL when they are assigned one of the ACL privileges. A user is removed from an ACL when they are assigned the none privilege. It is possible for a vault or file to have an empty list of user and privilege pairs, which is also known as a null ACL.

ACLs work in conjunction with user privileges, in particular vault top-level privileges. To perform an action on a vault or file, a user requires the appropriate ACL privilege or the appropriate vault top-level privilege. See Vault and File Access Control.

Related Topics

- Administer Access Control Lists

1.2.18 Vault and File Access Control

Access control lists (ACLs) work together with user privileges, in particular vault top-level privileges, to control access to Exascale vaults and files. To perform an action on a vault or file, a user requires the appropriate ACL privilege or the appropriate vault top-level user privilege. Because Exascale has no formal concept of vault or file ownership, all operations are governed by the combination of user privileges and ACLs.

The following table lists the minimum vault top-level user privilege, vault ACL privilege, or file ACL privilege that is required to perform various operations on Exascale vaults and files. Where relevant, associated ESCLI commands are listed along with each operation.

Operation	Required Vault Top-Level User Privilege	Required Vault ACL Privilege	Required File ACL Privilege
Create vault (mkvault)	vlt_inspect	Not applicable.	Not applicable.
List vaults (ls)	vlt_read	inspect	Not applicable.
List files in a vault (ls)	vlt_read	read	Not applicable.
Drop vault (rmvault)	vlt_manage	manage	Not applicable.
View vault attributes (lsacl, lsxattr, lstemplate)	vlt_read	inspect	Not applicable.
Alter vault attributes (chxattr, mktemplate, rmtemplate, rmxattr)	vlt_manage	manage	Not applicable.
Alter vault ACL (chacl)	vlt_manage	manage	Not applicable.
Create file (mkfile)	vlt_read	inspect	Not applicable.

Operation	Required Vault Top-Level User Privilege	Required Vault ACL Privilege	Required File ACL Privilege
Drop file (rmfile)	vlt_manage	manage	manage
Read and write file contents (putfile)	vlt_manage	manage	use
Read file contents (getfile)	vlt_use	use	read
View file attributes (lsacl, lsxattr)	vlt_read	read	inspect
Alter file attributes (chxattr, rmxattr)	vlt_manage	manage	use
Alter file ACL (chacl)	vlt_manage	manage	manage

To perform an operation, a user requires at least one of the privileges that is listed beside the operation. For example, to open a file for read-only access the requesting user must have at least one of the following:

- The vlt_use vault top-level user privilege.
- The use vault ACL privilege for the vault containing the file.
- The read file ACL privilege for the file being opened.

> **Note**
>
> - To create a snapshot or a clone, the user requires the privileges for the 'read file contents' operation to read the source file, and they also require the privileges for the 'create file' operation to create a file for the snapshot or clone. After creation, operations on snapshots and clones require the same privileges as for any other file.
> - Exascale ensures that users can manage the vaults and files that they create. During vault creation, if the creating user does not have the vlt_manage vault top-level user privilege, then Exascale adds the creating user to the vault ACL with the manage privilege. During file creation, if the creating user does not have the vlt_manage vault top-level user privilege and the user does not have the manage privilege in the vault ACL, then Exascale adds the creating user to the file ACL with the manage privilege.

Related Topics

- Modify User Privileges
- Administer Access Control Lists

1.2.19 Trust Store

The Exascale trust store is a group of digital certificates that facilitates trusted communication between servers running Exascale cluster services (EGS). The digital certificates ensure the identity of the EGS servers, which enables clients to trust the legitimacy of the Exascale cluster.

During the initial configuration of Exascale, the trust store is automatically created and distributed across the cluster.

The trust store is modified when EGS servers are added to, or deleted from, the Exascale cluster.

While most operations involving the trust store occur automatically, some manual operations are occasionally required. For example, the trust store must be manually fetched into a new user wallet.

Related Topics

- Exascale User Credentials
- Fetch the Trust Store

1.2.20 Resource Management

Resource management governs how various storage resources are consumed. Exascale resource management works in conjunction with existing Exadata I/O Resource Management (IORM) and Oracle Database Resource Manager (DBRM) capabilities.

Exascale resource management is implemented in a hierarchical manner:

- Firstly, inter-vault resource management governs how resources are allocated to different vaults using Exadata IORM.
- The resulting resource allocation for each vault is further divided among the different databases and block volumes using intra-vault resource management with Exascale resource profiles.
- Finally, the resulting resource allocation for each database is divided among PDBs and consumer groups using intra-database resource management with DBRM definitions.

The following topics describe each of these areas in greater detail.

- Inter-Vault Resource Management Using Exadata IORM
- Intra-Vault Resource Management Using Exascale Resource Profiles
- Intra-Database Resource Management Using DBRM

1.2.20.1 Inter-Vault Resource Management Using Exadata IORM

Inter-vault resource management defines how resources are allocated to different vaults. Inter-vault resource management utilizes Exadata IORM in conjunction with a vault plan, which controls resource allocation across multiple Exascale vaults.

By default, each vault has access to all of the resources in the Exascale cluster. However, an Exascale administrator can set various vault-specific resource provisioning attributes, which may limit the space, I/O bandwidth (IOPS), and caching resources available to the vault.

The vault plan is automatically derived from all the vault definitions and propagated to every storage server in the Exascale cluster. Furthermore, the vault plan automatically adjusts with changes to the resource provisioning attributes. No administrator intervention is required.

On a system that uses Exascale and traditional Exadata storage based on Oracle Automatic Storage Management (Oracle ASM), the vault plan coexists with other parts of the IORM plan. For example, an Exadata administrator can still define an inter-database plan (`dbplan`) to manage resource allocation across databases using Oracle ASM. In this case, resources are shared equally between Exascale and Oracle ASM.

1.2.20.2 Intra-Vault Resource Management Using Exascale Resource Profiles

Intra-vault resource management defines how resources are shared by the various Oracle databases and block store volumes that use a vault. Within Exascale, intra-vault resource management is governed by resource profiles.

By default, every Exascale client (Oracle database or block store volume) has access to all of the resources in their associated vault. Furthermore, I/O resources are shared equally when the system is under load.

To enable more granular I/O resource management, you can associate each Exascale client with a resource profile. You can define any number of resource profiles, but you can only associate an Exascale client with one resource profile at a time.

Each resource profile contains the following resource limits and settings:

- I/O Bandwidth (IOPS) — For each Exascale media type (HC and EF), you can define a limit value and a share value. While I/O bandwidth utilization is less than the vault capacity, the limit controls each client. However, when I/O bandwidth utilization reaches capacity, resource allocation across all clients is further governed by the share values.

  The limit value specifies the absolute limit of I/O bandwidth available to each client associated with the resource profile. The value represents a fraction out of 10000. For example, a value of 1 represents a limit of 1/10000 (0.01%), 5000 represents 5000/10000 (50%), 10000 represents 10000/10000 (100%, or effectively unlimited), and so on. If not specified, the default limit value is 10000 (effectively unlimited).

  The share value defines the proportional share of I/O bandwidth available to each client associated with the resource profile. Each client's share is relative to all other client shares. A higher share value implies higher priority. For example, consider a system with 2 clients, where client A has a share value of 2 and client B has a share value of 1. In this case, when I/O bandwidth utilization reaches capacity, client A gets 2/3 (66.67%) of the I/O bandwidth, which is twice as much as client B (1/3, or 33.33%). Now, consider adding client C with a share value of 7. After the addition of client C, when I/O bandwidth utilization reaches capacity, client A gets 2/10 (20%) of the I/O bandwidth, which is still twice as much as client B (1/10, or 10%), but client C gets 7/10 (70%) of the I/O bandwidth. The range of valid values is 1-100. If not specified, the default share value is 1.
- Flash Cache and Exadata RDMA Memory (XRMEM) Cache — For each type of cache, you can enable or disable use of the cache by clients associated with the resource profile. Then, for each enabled cache type, you can specify a minimum and maximum usage value in the range of 0 to 10000.

  The minimum value guarantees a portion of the cache for each client associated with the resource profile. Nominally, the value represents a fraction out of 10000. For example, a value of 1 represents a limit of 1/10000 (0.01%), 5000 represents 5000/10000 (50%), 10000 represents 10000/10000 (100%, or effectively unlimited), and so on. However, if the sum of all minimum values exceeds 10000 across all clients and resource profiles, then all

values are scaled down proportionally to ensure that minimum guarantees can be honored. If not specified, the default minimum value is 0 (no guaranteed minimum).

The maximum value specifies the absolute limit of cache space available to each client associated with the resource profile. The value represents a fraction out of 10000. For example, a value of 1 represents a limit of 1/10000 (0.01%), 5000 represents 5000/10000 (50%), 10000 represents 10000/10000 (100%, or effectively unlimited), and so on. If not specified, the default value is 10000 (effectively unlimited).

- Flash Log — You can enable or disable use of the flash log accelerator for clients associated with the resource profile. If not specified, the default setting enables the use of flash log.

You can also create a system-reserved resource profile named $UNASSIGNED. All Exascale clients not explicitly associated with a resource profile are automatically governed by the $UNASSIGNED profile. The $UNASSIGNED resource profile contains only two modifiable attributes, which specify the maximum fraction (out of 10000) of flash cache space and XRMEM cache space assigned to the profile. All other attributes of the $UNASSIGNED resource profile use the previously described default values.

All Exascale clients governed by the $UNASSIGNED profile share the specified cache resources. The behavior differs from regular resource profiles, where each application of the resource profile defines the resource allocation for one associated client.

If you do not create the $UNASSIGNED resource profile, all unassigned Exascale clients share any unassigned flash cache space and XRMEM cache space. If there is no unassigned space to share, the system automatically reserves 5% of the cache space for unassigned Exascale clients.

Resource over-provisioning across multiple clients and resource profiles is allowed. For over-provisioned resources, the system automatically adjusts the resource shares to maintain the relative proportions for each client.

For example, consider a system with the following resource profiles:

- GOLD resource profile:
  - EF IOPS: share=11, limit=5100
  - HC IOPS: share=40, limit=5200
  - Flash Cache: enabled=TRUE, minimum=500, maximum=1000
  - XRMEM Cache: enabled=TRUE, minimum=400, maximum=800
  - Flash Log: enabled=TRUE
- SILVER resource profile:
  - EF IOPS: share=6, limit=2500
  - HC IOPS: share=20, limit=2500
  - Flash Cache: enabled=TRUE, minimum=300, maximum=600
  - XRMEM Cache: enabled=TRUE, minimum=200, maximum=400
  - Flash Log: enabled=TRUE
- BRONZE resource profile:
  - EF IOPS: share=3, limit=900
  - HC IOPS: share=10, limit=1500
  - Flash Cache: enabled=TRUE, minimum=200, maximum=500

  - XRMEM Cache: enabled=TRUE, minimum=100, maximum=200
  - Flash Log: enabled=FALSE
- $UNASSIGNED resource profile:
  - Flash Cache: maximum=1000
  - XRMEM Cache: maximum=500

Also, imagine that the system hosts 8 databases, which are associated with the resource profiles as follows:

- DB1 and DB2 are associated with the GOLD resource profile.
- DB3 is associated with the SILVER resource profile.
- DB4 and DB5 are associated with the BRONZE resource profile.
- DB6, DB7, and DB8 are not explicitly associated with any resource profile. Therefore, they are implicitly associated with the $UNASSIGNED resource profile.

Now consider how the resource profiles are used to manage a specific resource, such as I/O bandwidth on high capacity storage media (HC IOPS):

- While the resource utilization is less than 100% of the vault capacity, each database is capped using the limit value. For example:
  - DB1: 5200 (52%)
  - DB2: 5200 (52%)
  - DB3: 2500 (25%)
  - DB4: 1500 (15%)
  - DB5: 1500 (15%)
  - DB6: 10000 (100%, default value, effectively unlimited)
  - DB7: 10000 (100%, default value, effectively unlimited)
  - DB8: 10000 (100%, default value, effectively unlimited)

  If the limits keep utilization to less than 100% of the vault capacity, then no further intervention is required.
- While the resource utilization is 100%, the resource is allocated proportionally using the share value. For example:
  - DB1: 40
  - DB2: 40
  - DB3: 20
  - DB4: 10
  - DB5: 10
  - DB6: 1 (default value)
  - DB7: 1 (default value)
  - DB8: 1 (default value)

  In this example, the sum of the shares is 123, resulting in the following resource allocations:
  - DB1: 40/123 (35.52%)
  - DB2: 40/123 (35.52%)

- DB3: 20/123 (16.26%)
- DB4: 10/123 (8.13%)
- DB5: 10/123 (8.13%)
- DB6: 1/123 (0.81%)
- DB7: 1/123 (0.81%)
- DB8: 1/123 (0.81%)

Related Topics

- Administer Resource Profiles

1.2.20.3 Intra-Database Resource Management Using DBRM

Intra-database resource management defines how resources are shared within an Oracle database, which may include multiple pluggable databases (PDBs). Intra-database resource management uses existing Oracle Database Resource Manager (DBRM) definitions, which are transparently propagated to Exadata storage server.

1.2.21 Exascale Block Store

The Exascale block store provides capabilities to create and manage arbitrary-sized raw block volumes based on Exascale storage. Each Exascale volume can be used as an Exascale Direct Volume (EDV) attachment.

The Exascale block store features:

- High availability - based on a clustered architecture with built-in redundancy and dynamic fail-over of software services
- High performance - utilizing Exadata RDMA memory and flash cache
- Reliability - using proven Exadata storage server technologies
- Security - employing standard security protocols and automatic volume encryption
- Rich functionality - including instantaneous volume snapshots and backups

The following topics introduce various Exascale block store concepts and features:

- Exascale Volumes
- Volume Attachments
- Volume Snapshots
- Volume Clones
- Volume Backups
- Volume Groups
- Oracle Advanced Cluster File System (ACFS) on Exascale
- Virtual Machine Images

1.2.21.1 Exascale Volumes

An Exascale block volume is an arbitrary-sized allocation of storage space, which can be used as an Exascale Direct Volume (EDV) attachment.

Internally, each volume is an Exascale file with special properties that identify it as block storage space. Like other Exascale files, physical space for the volume is only materialized when data is written to the volume.

Each volume is created in a user-specified vault. You can also optionally specify the physical media type used to store the volume. By default, volumes are stored on hard disk drives (HC media type).

Each volume can also be associated with a series of optional attributes, which define the detailed characteristics of the volume, such as the number of data copies (mirrors) that are maintained or the system resources that a volume is allowed to consume.

To use a volume, you must create a volume attachment.

Unlike regular Exascale files, volumes are not subject to access controls defined by user privileges and ACLs. Consequently, volume administration tasks (attachment, detachment, modification, and so on) must be performed by the volume owners or an Exascale cluster administrator (having the cl_admin privilege).

Related Topics

- Administer Exascale Volumes

1.2.21.2 Volume Attachments

To use a block volume, you must create a volume attachment.

An Exascale Direct Volume (EDV) attachment creates a association between the volume and an EDV device file on the Exadata compute node or Oracle Grid Infrastructure (GI) cluster hosting the attachment.

If you create a cluster-wide attachment, the EDV device file is created on every node in the GI cluster. If you create a node-specific attachment, the corresponding EDV device is only created on that node.

After attachment, the volume can be used as a raw block storage device or to support a file system. To implement Oracle Advanced Cluster File System (ACFS) on Exascale block volume storage, you must use an EDV attachment.

The I/O for each attachment is serviced by an EDV server process. Automatic data encryption and decryption occurs within the EDV process.

In addition to general file storage, an EDV attachment may be used to support Oracle Database data files for versions before Oracle Database 23ai. However, an EDV attachment cannot be used for a bootable volume.

Exascale supports the simultaneous use of multiple attachments for each volume, which can be used to support various types of clustered file systems.

Related Topics

- Administer Volume Attachments

1.2.21.3 Volume Snapshots

A volume snapshot is a thinly-provisioned read-only point-in-time copy of a volume.

Volume snapshots have the following uses:

- You can create attachments to a volume snapshot and use it as a read-only volume.
- You can use a volume snapshot as the source for a volume clone.

- You can use a volume snapshot as the source for a volume backup.

Related Topics

- Administer Volume Snapshots

1.2.21.4 Volume Clones

A volume clone is a thinly-provisioned read-write point-in-time copy of a volume snapshot.

A volume clone is functionally equivalent to a standard (non-cloned) volume.

Volume clones have the following uses:

- You can create attachments to a volume clone and use it like any other writable volume.
- You can create volume snapshots that are based on the volume clone.

Related Topics

- Administer Volume Clones

1.2.21.5 Volume Backups

A volume backup is a backup of an Exascale volume snapshot, which provides a consistent point-in-time copy of the volume.

The backup is stored in an Exascale vault, and the backup destination vault may differ from the vault containing the volume being backed up.

You can restore a volume backup into a new volume. When you restore a volume backup, the restored volume is separate from the volume backup and the original backup source volume. There is no ongoing association between a restored volume and the volume backup or the original backup source volume.

Related Topics

- Administer Volume Backups

1.2.21.6 Volume Groups

A volume group is a specific collection of Exascale volumes.

Volume groups can be used to:

- Easily identify volumes with a common purpose or inherent relationship.

  For example, you can use a volume group to collect all the volumes underpinning the image files associated with an individual virtual machine (VM). Also, you can optionally name the volume group, making it easier to understand its purpose and distinguish it from numerous other groups.

- Collectively manage I/O resources consumed by all volumes in the group.

  A volume group can optionally be configured as a resource-sharing volume group, using either aggregate or specified resource sharing.

  Under aggregate resource sharing, the volume group automatically shares the total I/Os per second (IOPS) bandwidth provisioned to all volumes in the group. In this case, an individual volume can exceed its volume-level limit by sharing unused IOPS provisioned to other volumes in the group. But the entire group is always governed by the aggregated limit.

With specified resource sharing, a specific IOPS bandwidth limit is associated with the volume group. In this case, the group is governed only by the specified limit.

In either case, a resource-sharing volume group also works in conjunction with I/O resource management (IORM) directives defined at other levels, such as the Exascale vault level.

- Efficiently create consistent snapshots of all volumes in a volume group.

  In one operation, you can create a set of volume snapshots with point-in-time consistency, which are easily identifiable by a unique batch ID.

Note the following details about volume groups:

- A volume group can contain only writable volumes, including volume clones. Volume snapshots cannot be part of a volume group.
- All volumes belonging to a volume group must reside in the same Exascale vault.
- A volume must always have at least one owner in common with each volume group it belongs to.
- A volume can be a member of up to five different volume groups. However, at any time, a volume can belong to only one resource-sharing volume group.

Related Topics

- Administer Volume Groups

1.2.21.7 Oracle Advanced Cluster File System (ACFS) on Exascale

Exascale contains integrated support for Oracle Advanced Cluster File System (ACFS) on Exascale block storage using Exascale Direct Volumes (EDV).

By using this facility, an Exascale administrator can quickly and easily create and manage ACFS files systems on Exascale block storage.

For example, using one simple command an Exascale administrator can:

1. Create an ACFS file system on the specified EDV block storage volume.
2. Register the ACFS details with the Oracle Grid Infrastructure (GI) cluster
3. Mount the file system on all nodes associated with the GI cluster.

Related Topics

- Administer an Advanced Cluster File System (ACFS) on Exascale
- About Oracle ACFS

1.2.21.8 Virtual Machine Images

Exadata systems configured with Exascale can use Exascale Direct Volumes (EDV) to store virtual machine (VM) guest image files for Exadata database server VMs.

This capability removes previous constraints imposed by the limited amount of local storage space available on each Exadata database server.

Furthermore, VM images on Exascale are easily accessible from any VM host. This capability effectively decouples VM guests and hosts, providing the infrastructure to enable quick and easy migration of a VM guest to another host.

Decoupling VM guests and hosts also opens up additional possibilities for:

- Balancing database server VM workloads by separating busy VMs onto different hosts.
- Reducing the impact of scheduled VM host maintenance by proactively moving VMs to another host.
- Reducing the impact of unscheduled VM host downtime by reactively moving VMs to another host.

1.2.22 Features

Exascale contains infrastructure to track and manage software features throughout the Exascale cluster, which is used automatically by Exascale software to ensure compatibility between all services across the Exascale cluster.

For example, consider the possible introduction of a new security protocol for communications between Exascale software services across different servers in an Exascale cluster. Clearly, communications would break if one server started using the new communications protocol before all servers were updated with the capability to understand it. By using the infrastructure to track and manage software features, the Exascale software can be updated to facilitate the new protocol, but the feature can remain disabled until the entire cluster is ready to support it.

The infrastructure to track and manage Exascale software features is mostly internal, but Exascale administrators can view the metadata associated with software features. Additionally, administrators have some manual controls, such as the ability to manually enable a disabled feature, but these controls are typically reserved for use under the guidance of Oracle Support.