Oracle Maximum
Availability Architecture

Oracle Sharding MAA Best Practices

Oracle Database 12*c* Release 2

ORACLE WHITE PAPER | JUNE 2017

Table of Contents

.

Introduction

With an increasing number of applications requiring 24x7 availability, reducing downtime for both unplanned outages and planned downtime becomes a significant concern and top priority. Oracle Sharding in Oracle Database 12*c* Release 2 provides a sharded database architecture with unlimited scalability and with the highest application availability for any “shard-ready” application. This MAA paper describes the high availability architecture options, their corresponding configurations, and operational practices for a shard-ready application. The Oracle MAA sharding solution provides the following features and benefits:

- Fault tolerance with zero points of failure
- Fault isolation in which any shard failure or maintenance has zero or very minimal impact on the overall application and sharded database
- Fail over each shard quickly for local high availability or remote disaster recovery
- Apply changes online, in a rolling manner, or switch over to an upgraded shard, with zero or minimal downtime for planned maintenance activities
- Migrate, split, and rebalance existing shards with zero or minimal application impact.
- Grow and scale the application by adding shards with zero or minimal application impact
- Route and load balance across various shards and across geographic regions
- Reduce manageability costs with centralized management interface for the sharded database

Overview of Oracle Sharding

Oracle Sharding is a true shared-nothing hardware architecture that provides linear scalability and high availability by distributing data and workloads across a pool of independent Oracle databases known as shards.

The pool of shards is presented to the application as a single logical database. The single logical database is known as a sharded database. Applications elastically scale (data, transactions, and users) to any level, on any platform, simply by adding shards to the sharded database. Data and workloads are automatically balanced across the shards transparent to the application. Scaling a sharded database up to 1,000 shards is supported in the first release of Oracle Database 12c Release 2.

Oracle Sharding uses a sharded database to provide linear scalability and fault isolation for suitable applications. A sharded database eliminates the possibility of a single physical database being unable to scale to meet application requirements. Similarly, a sharded database prevents a physical database from being a single point of failure for an application due to unplanned outages or planned maintenance.

The Oracle Sharding MAA reference architecture uses the Bronze, Silver, Gold, and Platinum MAA reference architectures as building blocks to provide shard-level high availability given that each shard is a standalone Oracle Database:

- **Bronze**: Oracle restart and backups for recovery of a shard
- **Silver**: Bronze, plus Oracle RAC or Oracle Active Data Guard for shard-level high availability

- **Gold**: Silver, plus Oracle Active Data Guard or Oracle GoldenGate[1] for shard-level high availability and disaster recovery
- **Platinum**: Gold, plus advanced Oracle features for shard-level high availability to make all unplanned outages, and even the most complex planned maintenance tasks, completely transparent to an application.

The sharding reference architecture also includes best practices that address any unique considerations for a sharded database. Refer to *Oracle Database High Availability Overview Guide* for more information about MAA reference architectures.

Components of the Oracle Sharding Architecture

The following figure illustrates the major architectural components of Oracle Sharding:

- Sharded database (SDB) – a single logical Oracle Database that is horizontally partitioned across a pool of physical Oracle Databases (shards) that share no hardware or software
- Shards - independent physical Oracle databases that host a subset of the sharded database
- Global service - database services that provide access to data in an SDB
- Shard catalog – an Oracle Database that supports automated shard deployment, centralized management of a sharded database, and multi-shard queries
- Shard directors – network listeners that enable high performance connection routing based on a sharding key
- Connection pools - at runtime, act as shard directors by routing database requests across pooled connections
- Management interfaces - GDSCTL (command-line utility) and Oracle Enterprise Manager (GUI)

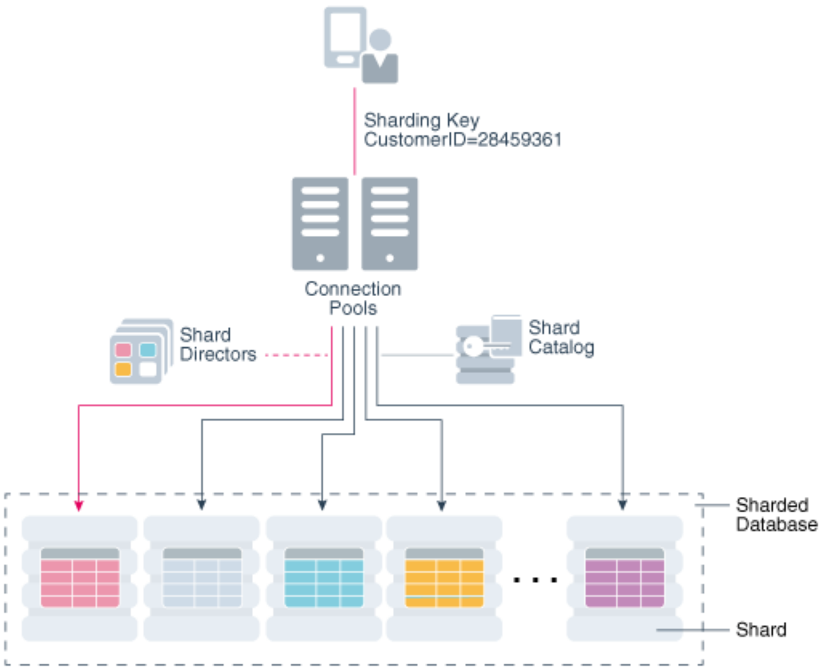

Figure 1. Oracle Sharding Architecture

For more information about Oracle Sharding architecture, see the *Oracle Database Administrator's Guide*.

[1] GoldenGate will be supported with Oracle Sharding in a future release

Application Requirements for Oracle Sharding

Applications must have the following characteristics in order to benefit from Oracle Sharding:

- OLTP applications with high transaction volumes which require low latency and extreme fault isolation. The current release of Oracle Sharding is not intended for data warehouse or analytical applications.
- Must be able to partition data for OLTP applications on a stable sharding key, for example, customer ID, and the data must mostly be accessed using the key.
- Must use Oracle integrated connection pools (UCP, OCI, ODP.NET, JDBC).
- Must be able to separate workloads that use direct routing from those that use proxy routing, that is, the each use separate connection pools.
- Separate global services for read/write and read-only workloads.
- For each key-based request, the application should check out a new connection from the pool by specifying the sharding key using the API provided with Oracle Sharding.

The effort required to use Oracle Sharding depends on the design of the application and the data model. For example:

- New OLTP applications can be easy to build. Oracle Sharding provides a simple declarative way of specifying sharded table families and duplicated tables. Administrators can add or subtract shards, and the sharding infrastructure rebalances data and workload automatically (for system-managed sharding). Applications never need to know how many shards there are or how data is distributed across them. Oracle Sharding provides a convenient API for providing the sharding key, load balancing across shard replicas, and so on.
- Home grown OLTP applications that were designed for sharding will require some amount of change to achieve the benefits of Oracle Sharding. Instead of using existing routing code, the application should use Oracle Sharding APIs. This may be a simple or more complex change depending on how closely integrated the home-grown routing code is with the application.
- Commercial Off-The-Shelf (COTs) or home grown OLTP applications that were never designed for sharding may prove challenging to convert. Such applications must change their database requests to access data by sharding key. They should also eliminate global secondary indexes and integrity constraints that must be enforced across shards and global sequences. Existing databases may require denormalization. The root table and all child tables must contain the sharding key. In spite of these challenges, customers with existing Oracle applications who wish to migrate to a sharded architecture will find it easier to move to Oracle Sharding than to alternative sharding solutions from various NoSQL vendors.

For more information about design considerations for sharded database applications, see the *Oracle Database Administrator’s Guide*.

MAA Best Practices for Oracle Sharding

The MAA Oracle Sharding reference architecture is different from the metal MAA reference architectures and solutions because it must address failures and maintenance activities to existing shard directors, shard catalogs, and shard databases. The MAA Oracle Sharding reference architecture does take advantage of existing MAA configuration and operational best practices when using Oracle's high availability technologies such as transparent client failover, Active Data Guard, and Oracle Real Application Clusters (RAC).

Figure 2 provides an overview of the MAA Oracle Sharding reference architecture.

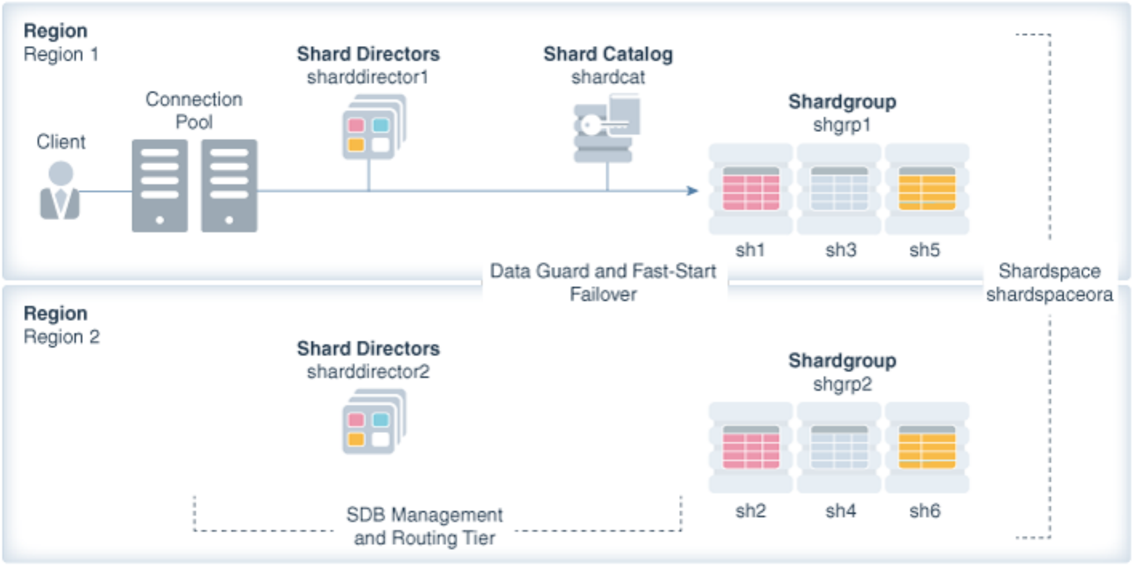

Figure 2. Oracle Sharding MAA Reference Architecture

The following are requirements of the MAA Oracle Sharding reference architecture.

- **Each client must communicate using an Oracle integrated connection pool (such as UCP, OCI, ODP.NET) and is directed to the appropriate shard for its transactions.** The connection pools are notified by Fast Application Notification events or timeouts when service for any data is moved to a different shard for unplanned outages, planned maintenance activities, or data movement due to elasticity. These connection pools are essentially for routing requests to the appropriate shards.
- **Install three shard directors per region**. Shard directors enable high performance data-dependent routing and aid in the management of the sharded databases. Each shard director is a stateless, light weight and intelligent listener that can repopulate its meta data from the shard catalog. When a connection pool establishes the initial connection, it may check with the shard director for the current shard data mapping. Once that relationship is established, no other connection pool or shard director communication is required until the mapping changes. Oracle recommends that you include at least three shard directors in your sharding environment so that if a shard director fails high availability is maintained among the remaining two shards directors. Each shard director should run on separate hardware.
- **Protect the shard catalog database with Data Guard.** The shard catalog is a very important database that contains centralized meta-data mapping of all the shards, and the materialized views for any duplicated tables. In general the shard catalog database is small (< 100 GBs) and read-only. Updates to the shard catalog database occur during 1) initial instantiation, deployment, and data load of the shard; 2) shard life cycle management, which includes any chunk movement; or 3) duplicated table changes. MAA recommends that you set up a local Data

Guard standby database configured with Maximum Availability database protection mode and Data Guard Fast-Start failover and a remote physical standby database. All shard catalog standby databases should use Active Data Guard for the best data protection, and they should reside on separate hardware and storage.

- For each additional remote region, create a remote physical standby database using ASYNC transport.
- Prior to any catalog database changes initiated by adding, dropping, or shard chunk movement, change redo transport to SYNC to at least one remote standby.
- **Provide high availability and data protection for the sharded database (SDB) using one of the MAA reference architectures**. Each shard contains a subset, or partition, of your critical sharded database and should be protected. The SDB architecture uses the same standard Bronze, Silver, Gold, and Platinum reference architectures as building blocks to provide shard-level high availability given that each shard is a standalone Oracle Database.
  - Bronze: Database restart and backups for recovery
  - Silver: Bronze, plus Oracle RAC or Active Data Guard for shard-level high availability
  - Gold: Silver, plus Active Data Guard if not already used for shard-level high availability and disaster recovery
  - Platinum: Gold, plus advanced Oracle features for shard-level high availability to make all unplanned outages, and even the most complex planned maintenance tasks, completely transparent to an application

Sharded Database Deployment Best Practices

At a high level the steps involved in deploying a sharded configuration are:

1. Create a database that hosts shard catalog
2. Grant roles and privileges on user accounts used by shard directors
3. Configure scheduler on the shard catalog database
4. Install shard directors (global service managers)
5. Install Oracle Database software and remote scheduler agents on the shard nodes
6. Configure scheduler agents on all of the shard nodes
7. Create the shard catalog
8. Add shard directors
9. Add credentials
10. Add shardgroups and shards
11. Execute DEPLOY
12. Add global services

For more information about sharded database deployment, see the *Oracle Database Administrator's Guide*

Configuration Best Practices

The following sections describe configuration best practices:

- Connection Pool and Client Failover
- All Metals Configuration Best Practices
- Metal-Specific Sharded Database Configuration Best Practices

- » Elasticity and Scalability
- » Operational Best Practices
- » Backup and Recovery Best Practices

Connection Pool and Client Failover

GLOBAL Services and FAN ONS

FAN uses the Oracle Notification Service (ONS) for event propagation to all clients from Oracle Database 12*c* and later and for JDBC, Tuxedo, and listener clients earlier than 12*c*. ONS is installed as part of Sharded Database deployment. ONS is responsible for propagating FAN events to all other ONS daemons it is registered with. You do not need to configure or enable FAN on the server side, with one small exception: OCI FAN and ODP FAN require –*notification* is set to TRUE for the Global Service by GDSCTL. With FAN auto-configuration on the client, ONS must be on the CLASSPATH or in the ORACLE_HOME dependent on your client.

General Steps for Configuring FCF Clients

Follow these steps before progressing to driver specific instructions:

1. Use a Dynamic Database Service

Using FAN requires that the application connects to the database using a dynamic global database service. This is a service created using GDSCTL. Do not connect using the database service or PDB service – these services are for administration only and are not supported for FAN. The TNSnames entry or URL must use the service name syntax and follow best practices by specifying a dynamic database service name. Refer to the examples later in this section. Use the Oracle Notification Service when you use FAN with JDBC thin or Oracle Database 12*c* Release 1 (12.1.0.1) OCI or ODP.Net clients, because FAN is received over ONS.

In Oracle Database 12*c* Release 1 and above ONS FAN auto-configuration is introduced so that FCF clients discover the server-side ONS networks and self configure. FAN is automatically enabled when ONS libraries or jars are present.

FAN auto-configuration removes the need to list the shard directors that an FCF client needs. Listing server hosts is incompatible with location transparency and causes issues with updating clients when the server configuration changes. Clients already use a TNS address string or URL to locate the shard director listeners. FAN auto-configuration uses the TNS addresses to locate the shard director listeners and then asks each server database for the ONS server-side addresses. When there is more than one shard director, for example, FAN auto-configuration contacts each and obtains an ONS configuration for each one. When using Oracle Database 12*c* Release 1, the ONS network is discovered from the URL. An ONS node group is automatically obtained for each address list when LOAD_BALANCE is off across the address lists.

By default the FCF client maintains three hosts for redundancy in each node group in the ONS configuration. Each node group corresponds to each GLOBAL data center. For example, if there is a primary database and several Data Guard standbys there are by default 3 ONS connections maintained at each node group. The node groups are discovered when using FAN auto-configuration, or for Oracle Database 12*c* Release 1 and earlier, use ons.configuration. With NODE_GROUPS defined by FAN auto-configuration, LOAD_BALANCE=OFF (the default), and more ONS end points are not required. If you want to increase the number of end points you can do this by increasing MAXCONNECTIONS. This applies to each node group. Increasing to 4 in this example maintains four ONS connections at each node. Increasing this value consumes more sockets.

```
oracle.ons.maxconnections=4 ONS
```

If the client is to connect to multiple clusters, and receive FAN events from both, for example in an Oracle RAC with Data Guard case, then multiple ONS node groups are required. FAN auto-configuration creates these node groups using the URL or TNS names for 12*c* client and 12*c* database. If not using auto-ons, specify the node groups in the Grid Infrastructure or oraaccess.xml configuration files.

2.Client Side Configuration

As a best practice add multiple shard directors to provide high availability. Configure clients for multiple connection endpoints where these endpoints are shard directors rather than local, remote, or single client access name (SCAN) listeners. For OCI clients use the following TNS names structure:

```
(DESCRIPTION=(CONNECT_TIMEOUT=90)(RETRY_COUNT=30)(RETRY_DELAY=3)
(TRANSPORT_CONNECT_TIMEOUT=3)
  (ADDRESS_LIST =
   (LOAD_BALANCE=on)
   (ADDRESS=(PROTOCOL=TCP)(HOST=SHDIR1)(PORT=1522)))
  (ADDRESS_LIST=
   (LOAD_BALANCE=on)
   (ADDRESS=(PROTOCOL=TCP)(HOST=SHDIR2)(PORT=1522)))
 (CONNECT_DATA=(SERVICE_NAME=sales)))
```

For JDBC thin clients use the following URL structure in Oracle Database 12*c* Release 1 (12.1.0.2) or earlier:

```
jdbc:oracle:thin =
(CONNECT_TIMEOUT=90)(RETRY_COUNT=30)(RETRY_DELAY=3)
  (ADDRESS_LIST=
   (LOAD_BALANCE=on)
   (ADDRESS=(PROTOCOL=TCP)(HOST=SHDIR1)(PORT=1522)))
  (ADDRESS_LIST=
   (LOAD_BALANCE=on)
   (ADDRESS=(PROTOCOL=TCP)(HOST=SHDIR2)(PORT=1522)))
 (CONNECT_DATA=(SERVICE_NAME=sales)))
```

Note that after Oracle Database 12*c* Release 1 (12.1.0.2), JDBC and OCI align and you should use the OCI version for all connection descriptions. Here are some guidelines:

- Always use dynamic global services created using GDSCTL to connect to the database.
- Do not use the database service or PDB service – these services are for administration only, not for application usage, and do not provide FAN and many other features because they are available at mount.
- For JDBC, use the current client driver (Oracle Database 12*c*) with current or older RDBMS.
- Use one DESCRIPTION in the TNS names entry or URL – using more causes long delays connecting when RETRY_COUNT and RETRY_DELAY are used.
- Set CONNECT_TIMEOUT=90 or higher to prevent logon storms for OCI and ODP clients.

- Use a lower setting for JDBC clients, CONNECT_TIMEOUT=4, as a temporary measure until TRANSPORT_CONNECT_TIMEOUT is available.
- Do not also set JDBC property oracle.net.ns.SQLnetDef.TCP_CONNTIMEOUT_STR because it overrides CONNECT_TIMEOUT.
- Set LOAD_BALANCE=ON per address to expand SCAN names.
- Do not use Easy*Connect syntax (EZConnect) because it has no high availability capabilities.

3.Application Level Configuration

This section covers the configuration guidelines for various clients.

3.1 Configuring FAN for 12*c* Java Clients Using Universal Connection Pool

The best way to take advantage of FCF with the Oracle Database JDBC thin driver is to use either the Universal Connection Pool (UCP) or WebLogic Server Active GridLink. Setting the pool property FastConnectionFailoverEnabled on the Universal Connection Pool enables Fast Connection Failover (FCF). Active GridLink always has FCF enabled by default. Third party application servers including IBM WebSphere and Apache Tomcat support UCP as a connection pool replacement. For more information on embedding UCP with other web servers refer to the following white papers.

“Design and Deploy WebSphere Applications for Planned, Unplanned Database Downtimes and Runtime Load Balancing with UCP” at

http://www.oracle.com/technetwork/database/application-development/planned-unplannedrlb-ucp-websphere-2409214.pdf

“Design and deploy Tomcat Applications for Planned, Unplanned Database Downtimes and Runtime Load Balancing with UCP” at

http://www.oracle.com/technetwork/database/application-development/planned-unplanned-rlb-ucptomcat-2265175.pdf

Follow these configuration steps to enable Fast Connection Failover:

1. Specify a dynamic database service name and the JDBC URL structure. The connection URL of a connection factory must use the service name syntax and follow best practice by specifying a dynamic database service name and the JDBC URL structure (see examples above and below). No other URL formats provide high availability. The URL may use JDBC thin or JDBC OCI.

2. If wallet authentication has not previously been established, then configure remote. Use the pool property setONSConfiguration, which can be set in a property file as shown in the following example. The property file must contain an oracle.ons.nodes property, and optionally, properties for oracle.ons.walletfile and oracle.ons.walletpassword. An example of an ons.properties file is shown here.

```
PoolDataSource pds =
PoolDataSourceFactory.getPoolDataSource();
pds.setConnectionPoolName("FCFSamplePool");
pds.setFastConnectionFailoverEnabled(true);
pds.setONSConfiguration("propertiesfile=/usr/ons/ons.prope
rties");
pds.setConnectionFactoryClassName("oracle.jdbc.pool.Oracle
DataSource");
```

```
pds.setURL("jdbc:oracle:thin@((CONNECT_TIMEOUT=4)(RETRY_CO
UNT=30)(RETRY_DELAY=3) "+ " (ADDRESS_LIST = "+ "
(LOAD_BALANCE=on) "+ " ( ADDRESS = (PROTOCOL =
TCP)(HOST=SHDIR1)(PORT=1522))) "+ " (ADDRESS_LIST = "+ "
(LOAD_BALANCE=on) "+ "( ADDRESS = (PROTOCOL =
TCP)(HOST=SHDIR2)(PORT=1522)))"+
"(CONNECT_DATA=(SERVICE_NAME=service_name)))");
```

3. Ensure the pool property `setFastConnectionFailoverEnabled=true` is set.

4. The CLASSPATH must contain ons.jar, ucp.jar, and the jdbc driver jar file, for example ojdbc7.jar.

5. If you are using JDBC thin with Oracle Database 12*c* Release 1, Application Continuity can be configured to failover the connections after FAN is received.

6. If the database is earlier than Oracle Database 12*c* Release 1, or if the configuration needs different ONS end points than those auto-configured, the ONS end points can be enabled.

3.2 Configuring 12c OCI Clients

1. Starting with 12*c* Release 2, the client install comes with ONS linked into the client library. Using ONS auto-configuration, the ONS end points are discovered from the TNS address. This automatic method is the recommended approach. Like ODP.Net, manual ONS configuration is also supported using oraaccess.xml.

2. Enable FAN high availability events for the OCI connections. To enable FAN requires editing the OCI file oraaccess.xml to specify the global parameter events. This file is located in $ORACLE_HOME/network/admin. See the following whitepaper for additional information: http://www.oracle.com/technetwork/database/options/clustering/overview/fastapplicationnotification12c-2538999.pdf.

3.3 Controlling Logon Storms

Small connection pools are strongly recommended, but when you have many connections controlling logon storms can be done by tuning servers that host shard directors. To tune the servers:

» Increase the Listen backlog at the OS level. To have the new value take effect without rebooting the server, run the following as root:

```
echo 8000 > /proc/sys/net/core/somaxconn
```

» To persist the value across reboots, also add the following setting to /etc/sysctl.conf.

```
net.core.somaxconn=6000
```

» Increase QUEUESIZE for the shard director. Update sqlnet.ora in the Oracle home that the listeners are running from to increase the QUEUESIZE parameter:

```
TCP.QUEUESIZE=6000
```

Metal-Specific Sharded Database Configuration Best Practices

Each MAA reference architecture, or high availability tier, utilizes an optimal set of Oracle Database capabilities that, when deployed together, reliably achieve a given service level for high availability and data protection.

Oracle MAA offers a choice of architecture patterns for high availability and scalability:

- A set of standard reference architectures, Bronze, Silver, Gold, and Platinum, that provide application transparent scalability (with Oracle RAC), data protection, high availability, and disaster recovery for the Oracle Database.
- A special-purpose reference architecture that uses Oracle Sharding for linear scalability with complete fault isolation. The Oracle Sharding MAA reference architecture, introduced in Oracle Database 12*c* Release 2, is a separate MAA reference architecture that is only applicable to shard-ready applications. The Oracle Sharding reference architecture uses these same standard Bronze, Silver, Gold, and Platinum reference architectures as building blocks to provide shard-level high availability, given that each shard is a standalone Oracle Database.

Bronze: Oracle Restart (or Single Instance with Oracle Clusterware) Configuration Best Practices

The Bronze architecture uses a single instance Oracle Database; there is no clustering technology used for automatic failover if there is an outage of the server on which the Oracle Database instance is running. When a server becomes unusable or the database unrecoverable, RTO is a function of how quickly a replacement system can be provisioned or a backup restored. In a worst case scenario of a complete site outage there will be additional time required to perform these tasks at a secondary location, and in some cases this can take days.

Oracle Recovery Manager (RMAN) is used to perform regular backups of the Oracle Database. The RPO, if there is an unrecoverable outage, is equal to the data generated since the last backup was taken. Copies of database backups are also retained at a remote location or on the Cloud for the dual purpose of archival and disaster recovery should a disaster strike the primary data center.

Silver: Configuration Best Practices

The Silver tier builds upon Bronze by incorporating clustering technology for improved availability for both unplanned outages and planned maintenance. Silver uses Oracle RAC or Oracle RAC One Node for high availability within a data center by providing automatic failover should there be an unrecoverable outage of a database instance or a complete failure of the server on which it runs. Oracle RAC also delivers substantial benefits by eliminating many types of planned downtime by performing maintenance in a rolling manner across Oracle RAC nodes.

Silver, Gold, and Platinum: Data Guard Configuration Best Practices

All other configuration best practices will be common between the Oracle Sharding MAA reference architectures and standard MAA reference architectures. Refer to 12.2 HA Overview or MAA reference architecture white paper. http://www.oracle.com/technetwork/database/availability/maximum-availability-wp-12c-1896116.pdf

Elasticity and Scalability

For the steps to add or remove shards, refer to the Sharded Database Lifecycle Management chapter in the Administrator's Guide. - http://docs.oracle.com/database/122/ADMIN/sharding-lifecycle-management.htm#ADMIN-GUID-1A3B887E-148D-4167-81B5-B0FA35746E4B

Operational Best Practices

Standard operational practices still apply. For more information about these best practices, see *Oracle Database High Availability Overview* and *Oracle Database Administrator's Guide*.

Use operational best practices to provide a successful MAA implementation.

- Understand Availability and Performance SLAs
- Implement and Validate a High Availability Architecture That Meets Your SLAs
- Establish Test Practices and Environment
- Set up and Use Security Best Practices
- Establish Change Control Procedures
- Apply Recommended Patches and Software Periodically
- Execute Disaster Recovery Validation
- Establish Escalation Management Procedures
- Configure Monitoring and Service Request Infrastructure for High Availability
- Check the Latest MAA Best Practices

Backup and Recovery Best Practices

This section discusses the motivation and tools for maintaining good database backups, for using Oracle database recovery features, and for using backup options and strategies made possible with Oracle database features.

Use Recovery Manager (RMAN) to Backup Database Files

Recovery Manager (RMAN) is Oracle's utility to backup and recover the Oracle Database. Because of its tight integration with the database, RMAN determines automatically what files must be backed up. More importantly, RMAN knows what files must be restored for media-recovery operations. RMAN uses server sessions to perform backup and recovery operations and stores metadata about backups in a repository. RMAN offers many advantages over typical user-managed backup methods, including:

- Online database backups without placing tablespaces in backup mode
- Efficient block-level incremental backups
- Data block integrity checks during backup and restore operations
- Test backups and restores without actually performing the operation
- Synchronize a physical standby database with the primary database

Use Oracle Secure Backup for Backups to Tape

Oracle Secure Backup delivers unified data protection for heterogeneous environments with a common management interface across the spectrum of servers. Protecting both Oracle databases and unstructured data, Oracle Secure Backup provides centralized tape backup management for your entire IT environment, including:

- Oracle database through the Oracle Secure Backup built-in integration with Recovery Manager (RMAN)
- File system data protection: For UNIX, Windows, and Linux servers
- Network Attached Storage (NAS) data protection leveraging the Network Data Management Protocol (NDMP)

Oracle Secure Backup is integrated with RMAN providing the media management layer (MML) for Oracle database tape backup and restore operations. The tight integration between these two products delivers high-performance Oracle database tape backup.

Specific performance optimizations between RMAN and Oracle Secure Backup that reduce tape consumption and improve backup performance are:

- Unused block compression: Eliminates the time and space usage needed to backup unused blocks
- Backup undo optimization: Eliminates the time and space usage needed to backup undo that is not required to recover the current backup.

You can manage the Oracle Secure Backup environment using the command line, the Oracle Secure Backup Web tool, and Oracle Enterprise Manager Cloud Control.

Using the combination of RMAN and Oracle Secure Backup provides an end-to-end tape backup solution, eliminating the need for third-party backup software.

Optionally, during Zero Data Loss Recovery Appliance deployment, Oracle Secure Backup can be configured and integrated automatically with the Recovery Appliance.

Use Zero Data Loss Recovery Appliance to Back Up Database Files

Zero Data Loss Recovery Appliance is a new data protection solution integrated with the Oracle Database that eliminates data loss exposure and dramatically reduces backup and data protection overhead on production servers. The recovery appliance easily protects all databases in the data center with a scalable cloud architecture, ensures end-to-end data validation, and automates the management of the entire data protection lifecycle for all Oracle databases through a unified Enterprise Manager Cloud Control interface.

The recovery appliance serves as a destination for real-time redo transport for all Oracle Database 11*g* and 12*c* databases, thus providing data loss protection until the last sub-second for transactional data.

In conjunction with Oracle Recovery Manager (RMAN), the Recovery Appliance minimizes the impact of running backups against the database by only requiring a single incremental Level 0 (full) database backup on day 1, and then Incremental Level 1 database backups thereafter.

When a database needs to be restored, the Recovery Appliance constructs a level 0 copy of the data file as it would have been at the time of the most recent incremental level 1 backup, thereby reducing the amount of data that needs to be recovered after the database restore is completed.

The Recovery Appliance also manages the transfer of data from the appliance to a tape library using Oracle Secure Backup (OSB) for purposes of tape vaulting, as well as replicating backups to another recovery appliance for faster offsite data protection.

MAA Best Practices for Database Backup and Recovery

All of the backup MAA best practices for single-instance databases are applicable to a sharded database. For complete information about MAA best practices for backup and recovery of Oracle database see *Oracle Database High Availability Best Practices*. The following list summaries these best practices at a high level:

- Determine backup frequency and retention policy based on transactions per second and your business requirements
- To improve the speed of RMAN incremental backups enable block change tracing (BCT)
- Enable autobackup for the control file and server parameter file

- To minimize the primary shards' load and effectively use the standby resources, offload the backups to physical standby databases
- Determine the undo retention based on flashback query and flashback table needs; determine the optimal number of RMAN channels
- Monitor memory and I/O usage during backups
- Optimize the number of channels for best performance
- Use the RMAN RESTORE VALIDATE command to test recovery procedures and to assess the time needed to restore (minus the time to write to disk)
- Use data protection parameters discussed in MOS Note "Best Practices for Corruption Detection, Prevention, and Automatic Repair - in a Data Guard Configuration" (MOS Doc ID 1302539 in https://support.oracle.com).

MAA Best Practices for Backup and Recovery Specific to Sharded Databases

As mentioned previously, all generic backup and recovery best practices apply to sharded databases. The following are additional best practices specific to sharded databases:

- Prevent backups from occurring during chunk movement. Chunk movement occurs automatically when a shard is added, or when CHUNK movement is performed by the DBA. Avoid running RMAN backups during chunk movements because backups taken during chunk movement will have incorrect layout of DATA.

  To determine if chunk movement is occurring or is scheduled, use the GDSCTL CONFIG CHUNKS command. Determine if chunk movement is scheduled. If necessary, suspend chunk movement so that a backup can be taken, using the following commands.

  ```
  GDSCTL> alter move -suspend -chunk 3,4
  GDSCTL> alter move -resume -chunk 3,4
  ```

  As a best practice always perform backups before and after periods of large chunk movement.

- In general restore and recovery of a sharded database is the same as for normal databases, and all existing MAA best practices apply. However, a point in time restore of a shard may lead to incorrect chunk layout and missing DDL statements. This results in the individual shard being out of sync with the shard catalog. To correct this inconsistency, run GDSCTL VALIDATE and RECOVER SHARD commands for the shard that was just restored to first identify any inconsistency and secondly to reconcile them with the shard catalog. Consider the following process flow whenever a shard is restored:

  1. Disable the global service for the shard to be restored and recovered.
  2. Perform database restore.
  3. Identify any issues to be corrected by using the GDSCTL VALIDATE command against the shard .

     ```
     GDSCTL> validate
     ```

  4. Sync the restored shard with the shard catalog.

     ```
     GDSCTL> recover shard -full
     ```

  5. Once recovered, identify any additional issues that may need to be corrected.

     ```
     GDSCTL> validate
     ```

  6. Enable the global service once all issues have been corrected.

Recover FULL mode performs a complete recovery which covers DDL operations, failed chunk migration, tablespace sets reconstruction, and database parameters.

MAA Best Practices for Backup and Recovery Specific to Shard Catalog Databases

As mentioned above, all generic backup and recovery best practices apply to shard catalog databases. This is true as long as the shard catalog database is restored and recovered to a consistent point in time with the shards within the configuration. For example, a restore and recovery of the shard catalog that results in no data loss. However, performing a point in time recovery of the shard catalog can result in the catalog having an inconsistent view of the sharded configuration. There is no ability to reconcile or rebuild an accurate view of shard configuration based on information from the shards in the initial release.

The best practice to protect the shard catalog database is to have a standby database configured with Data Guard Max Availability mode. Optionally use a FAR/FAST sync instance to minimize the performance impact of shard catalog (primary) database failure. Protecting the shard catalog database with Max Availability mode ensures that any failover results in a zero data loss failover, and will not result in any inconsistency with the sharded configuration.

Outage Types and Application Impact Analysis

Using an MAA Oracle Sharding solution versus a central consolidated database, higher application availability is achievable using our recommended MAA reference architectures. The table below gives examples of the potential application impact for various unplanned sharded database outages. The outage table matches a similar format as Tables 4-1 “Outage Types and Oracle High Availability Solution for Unplanned Downtime” (also refer to Table 4-2) in *Oracle Database High Availability Overview*. Note that the percentage of application impact varies depending on the total number of SDBs and regions. The examples of application impact are based on a sharded database configuration with 100 shards evenly distributed across 5 regions.

TABLE 1. UNPLANNED OUTAGESOLUTIONS FOR MAA ORACLE SHARDING REFERENCE ARCHITECTURE

Shard Outage Scope	Oracle Sharding MAA Solution	Application Impact With 100 Shards Evenly Distributed Across 5 Regions	Application Impact With Non-Sharded Application Using MAA Reference Architectures
Instance or node failure (but restartable)	Bronze: Oracle Restart Silver/Gold: Oracle RAC or Data Guard failover Platinum: Application Continuity and Oracle RAC or Data Guard failover	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Minutes Silver: Seconds Gold: Seconds Platinum: Zero application outage	**Entire application is impacted.** Bronze: Minutes to an hour Silver: Seconds Gold: Seconds Platinum: Zero application outage
Permanent node failure (but storage available)	Bronze: Restore and recovery Silver/Gold: Oracle RAC or Data Guard failover Platinum: Application Continuity and Oracle RAC or Data Guard failover	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Hours to Day Silver: Seconds Gold: Seconds Platinum: Zero application outage Refer to Example 1: Impact of Sharded Database Failure	**Entire application is impacted.** Bronze: Hours to a day Silver: Seconds Gold: Seconds Platinum: Zero application outage
Storage failures (not complete storage failure)	Oracle Automatic Storage Management and Storage Redundancy	No application downtime	No application downtime
Data corruptions	Bronze: Basic protection. Some corruptions require restore and recovery. Silver: If using Oracle RAC, then same as Bronze. If using Active Data Guard (ADG), comprehensive corruption protection and Auto Block Repair Gold/Platinum: Comprehensive corruption protection and Automatic Block Repair with Oracle Active Data Guard	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Hours to a day Silver: If using Oracle RAC, hours to a day. If ADG, zero to seconds. Gold: Zero to seconds Platinum: Zero application outage	**Entire application is impacted.** Bronze: Hours to a day Silver: Hours to a day Gold: Zero to seconds Platinum: Zero application outage

Human errors resulting in incorrect results	Oracle Security Features Oracle Flashback Technology	**100% availability for 99% of application.** For 1% of application impacted, dependent on logical failure	Dependent on logical failure
Database unusable, system or storage failures, or wide spread corruptions	Bronze: Restore and recovery Silver: If uaing Oracle RAC, then same as Bronze. If using ADG, Data Guard failover Gold/Platinum: Data Guard failover	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Hours to a day Silver: If using Oracle RAC, hours to a day. If using ADG, zero with auto-block repair of physical data corruptions or seconds for other failures. Gold: Seconds Platinum: Zero application outage Refer to Example 1: Impact of Sharded Database Failure	**Entire application is impacted.** Bronze: Hours to a day Silver: Hours to a day Gold: Seconds Platinum: Zero application outage
Site failure Assumption:100 shards spread out among 5 regions. i.e., 20 shards per region (Site is mapped to a region)	Bronze/Silver: Restore and recovery Gold/Platinum: DNS and Data Guard failover	100% availability for 80% of application and shards. For 20% that are impacted: Bronze/Silver: Hours to a day Gold: Seconds Platinum: Zero application outage	**Entire application is impacted.** Bronze: Hours to days Silver: Hours to days Gold: Seconds Platinum: Zero application outage

Example 1: Impact of Shard Failure

A specific shard failure in a sharded database has limited overall application impact because the remaining shards continue to function. Depending on the number of shards and percentage of the data that failed shard contains, the overall application impact can be very small. When a shard fails (for example, a non-RAC node, database, cluster, site, or certain corruption), Data Guard Fast-Start failover can be initiated and the application using the shard can resume in seconds as observed in the following graph.

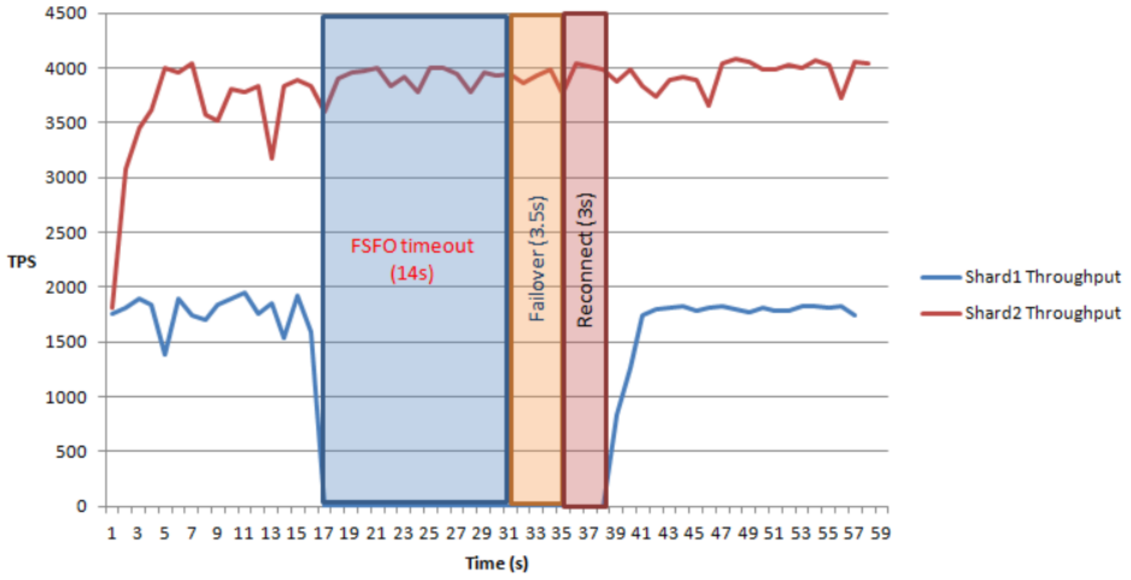

Figure 3. Shard outage has zero impact on surviving shards

For the additional sharded database architecture components, Oracle MAA recommends that you configure them as described in this document to ensure zero data loss and zero application impact. Refer to Table 2 below for the outages, recommended MAA solution, and expected application impact.

TABLE 2 UNPLANNED OUTAGE SOLUTIONS FOR ORACLE SHARDING INFRASTRUCTURE COMPONENTS

Outage Scope	Oracle Sharding MAA Solution	Application Impact With 100 Shards Evenly Distributed Across 5 Regions
Shard director failure (light weight, no database)	3 shard directors per region Failover to shard director Restart failed shard director	No application downtime Refer to Example 3: Impact of Shard Director Failure
Shard catalog failure (small, light weight database)	Active Data Guard Fast Start Failover Maximum Availability protection mode Local Data Guard using SYNC transport and remote Data Guard in each region	No application downtime for workloads using direct routing During the failover, application workloads using proxy routing are impacted Shard chunk movement is restricted for seconds to minutes Refer to Example 4: Impact of Shard Catalog Failure

Example 3: Impact of a Shard Director Failure

A shard director failure, or adding, removing, or restarting a shard director, has zero application impact, as shown in the following graph.

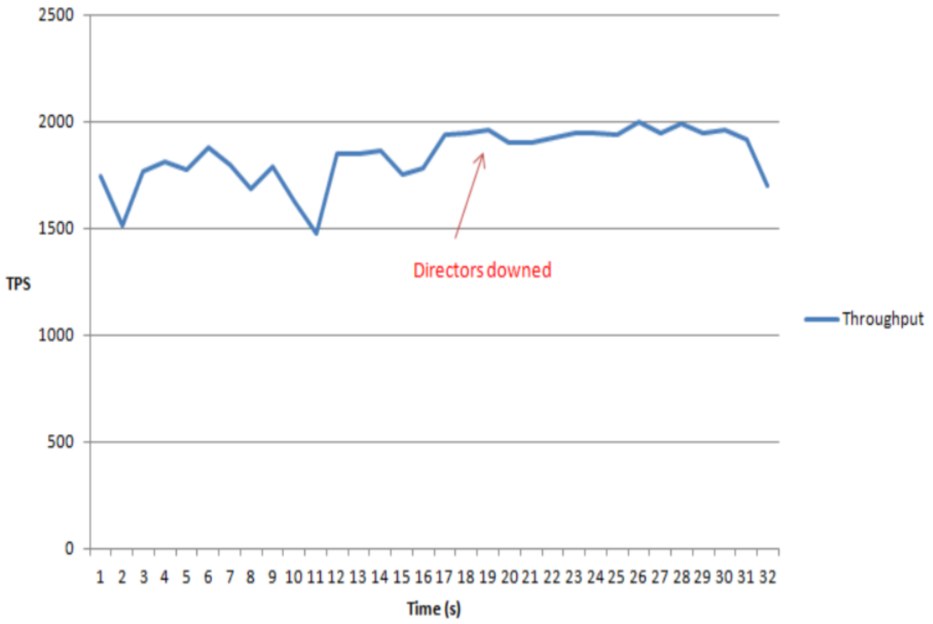

Figure 4. Shard Director Outage Has Zero Impact on Availability

Example 4: Impact of Shard Catalog Failure

A failure of the shard catalog results in zero application impact and a zero data loss failover of a shard catalog database using the Data Guard Fast-Start failover solution. While the shard catalog database is down for that short period, shard directors cannot be restarted and no ongoing chunk movement or new chunk movement can take place. to the following graph shows the application impact when the shard database is down temporarily.

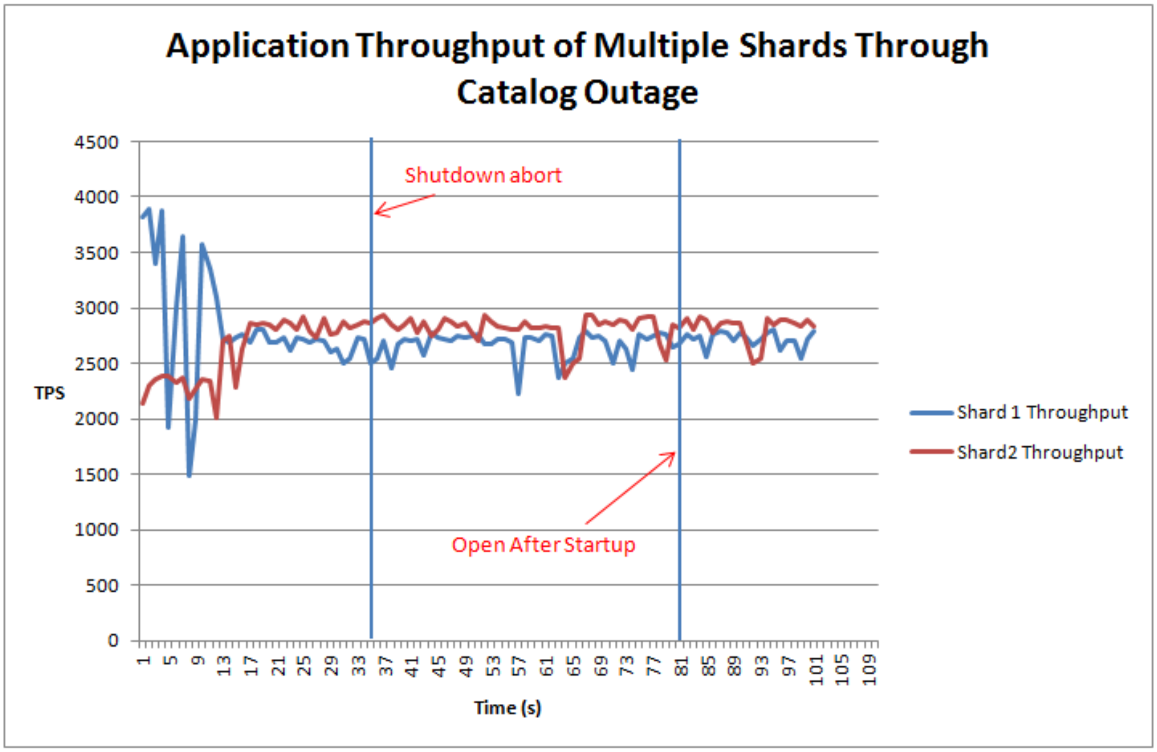

Figure 5. Shard Catalog Outage Has Zero Impact on Availability for OLTP

Planned Maintenance Types and Application Impact Analysis

Similar to the unplanned outage solutions described above, using the MAA Oracle Sharding solution rather than our recommended MAA reference architectures provides much higher availability for various planned maintenance activities . Refer to Table 3 below for examples of the potential application impact for various planned maintenance activities on the sharded database. The table matches a similar format to Tables 5-7, “Oracle High Availability Solutions for System and Software Maintenance,” in *Oracle Database High Availability Overview*. Note that the percentage of application impact varies depending on the total number of shards and regions.

TABLE 3 SHARDED DATABASE PLANNED MAINTENANCE

Shard Maintenance Scope	Oracle Sharding MAA Solution	Application Impact With 100 Shards Evenly Distributed Across 5 Regions	Application Impact With Non-Sharded Application Using MAA Reference Architectures
Operating system and hardware upgrades	Bronze: Upgrade and restart Silver/Gold: Oracle RAC or Data Guard rolling upgrade Platinum: Application Continuity and Oracle RAC or Data Guard failover	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Minutes to an hour Silver: Zero to seconds Gold: Zero to seconds Platinum: Zero application outage	**Entire application is impacted.** Bronze: Minutes to an hour Silver: Seconds Gold: Seconds Platinum: Zero application outage

Oracle interim patches or diagnostic patches	All Metals: Online patching	**100% availability for application.**	All Metals: Zero application impact
Oracle Database and Oracle Grid Infrastructure bundle patches, Patch Set Updates (PSU), Critical Patch Updates (CPU)	Bronze: Upgrade and restart Silver/Gold: Oracle RAC or Data Guard rolling upgrade Platinum: Application Continuity and Oracle RAC or Data Guard failover	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Minutes to an hour Silver: Zero to seconds Gold: Zero to seconds Platinum: Zero application outage	**Entire application is impacted.** Bronze: Minutes to Hour Silver: Seconds Gold: Seconds Platinum: Zero application outage
Oracle Database and Oracle Grid Infrastructure Patch Set (for example, Oracle Database 12.2.0.1 or 12.2.0.2) and Major Upgrade (for example, Oracle Database 12.2 to future release)	Bronze: Upgrade and restart Silver with Oracle RAC: Upgrade and restart Silver with ADG: Data Guard rolling upgrade with transient logical standby Gold/Platinum: Data Guard rolling upgrade with transient logical standby	**100% availability for 99% of application.** For 1% of application impacted: Bronze: Minutes to an hour Silver if RAC: Minutes to an hour Silver with ADG: Zero to seconds Gold: Zero to seconds Platinum: Zero to seconds	**Entire application is impacted.** Bronze: Hours to a day Silver: Hours to a day Gold: Zero to seconds Platinum: Zero with Oracle GoldenGate

Oracle Sharding Benchmark on Oracle Bare Metal Cloud

This section covers the results of the Oracle Sharding MAA benchmark on Oracle Bare Metal Cloud.

The objectives of this benchmark are to:

- Elastically scale-out the Sharded Database (SDB) on Oracle Bare Metal IaaS Cloud – following the Oracle Maximum Availability Architecture (MAA) best practices
- Demonstrate the linear scalability of relational transactions with Oracle Sharding
- Observe the fault isolation aspects of a sharded database

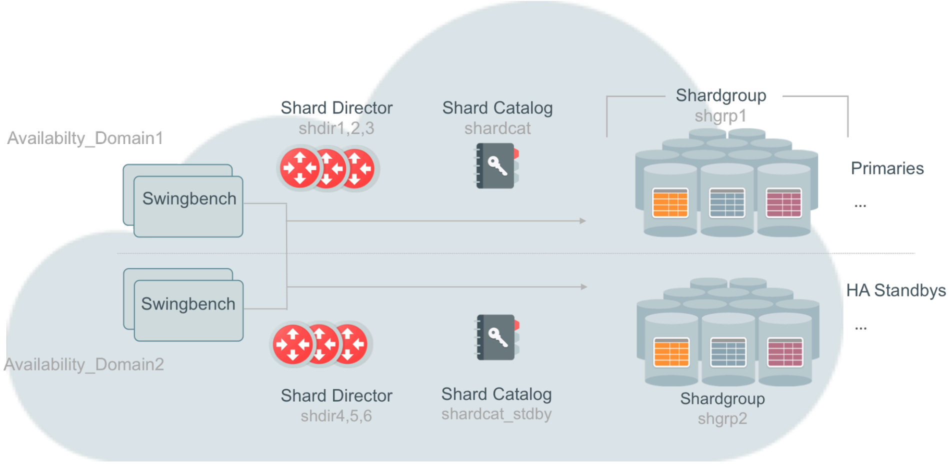

Figure 6. Sharded Database Topology on Oracle Bare Metal Cloud

As shown in Figure 6, the sharded database used in this benchmark has two shard groups - one in each of the availability domains of the Oracle Bare Metal Cloud. The network latency between the availability domains is less than 1ms. Shardgroup1 in Availability_Domain1 has 100 primary shards and the shardgroup2 in Availability_Domain2 has 100 HA standby shards. Each shard is hosted on dedicated bare metal server, which has 36 cores, 512G RAM and four 12.8TB NVMe flash disks. These flash disks are used for the creation of two ASM Diskgroups (+DATA, +RECO) with normal redundancy. Three shard directors were deployed in each of the Availability Domains for high availability. The shard catalog is placed in Availability_Domain1 and its standby is located in Availability_Domain2.

This benchmark uses the Swingbench Order-entry application. Customer_id column is defined as the sharding key. The data model has been modified so that every sharded table in the table family contains the sharding key. Tables used for reference data have been defined as duplicated tables. The connection pooling code has been modified to use the Oracle Sharding API calls - to check out a connection on a given shard based the sharding key.

Role-based global services have been created so that read-write transactions run on primary shards and queries run on standby shards. The total size of the sharded database is 50TB with 500G on each of the 200 shards.

This sharded database uses Active Data Guard in Max Availability with Fast-Start Failover for replication. The SDB is deployed automatically using the “CREATE SHARD” and Oracle Sharding automatically configured the replication. The FSFO observers are automatically started on the regional shard director.

Here are the steps taken for the execution of the benchmark:

1) Begin application workload on 25 primary shards and 25 standby shards

2) Bring up 25 more primary and 25 more standby shards
3) Repeat step #2 until all the 200 shards are online on both Availability Domains
4) Observe linear scaling of transactions per second (TPS) and queries per second (QPS)
5) Compute the total read-write transaction per second and queries per second across all shards in the sharded database

Following is the table that shows the TPS and QPS observed, as the SDB is elastically scaled-out.

TABLE 1 LINEAR SCALABILITY OF TRANSACTIONS AS SHARDS ARE ADDED

Primary shards	Standby shards	Read/Write Transactions per Second	Ready Only Queries per Second
25	25	1,180,000	1,625,000
50	50	2,110,000	3,260,000
75	75	3,570,000	5,050,000
100	100	4,380,000	6,820,000

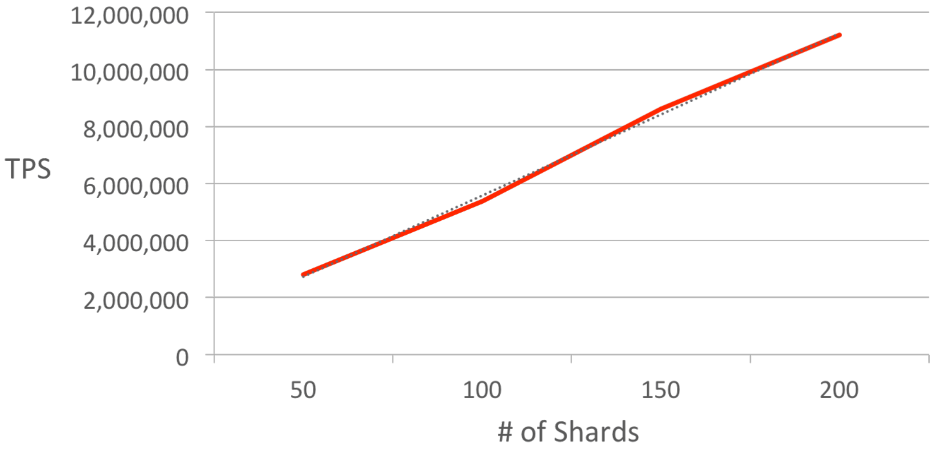

Figure 7. Transactions and queries scaled linearly as shards are added

Figure 7 illustrates that as shards were doubled, tripled and quadrupled, we were able to observe that the rate of transactions and queries doubled, tripled and quadrupled accordingly. This demonstrated the frictionless linear scaling due to shared-nothing hardware architecture of Oracle Sharding. Altogether we were able to execute 11.2 Million transactions per second that includes 4.38 Million read-write transactions per second across all the 100 primary shards and 6.82 Million read-only transactions per second across all the 100 active standby shards.

The study also illustrated that Oracle Sharding provided extreme data availability. When a failure was induced on a given shard, there was absolutely no impact to the other shards in the SDB. This is due to zero shared hardware or software among the shards.

As can be seen, Oracle Sharding provides both linear scalability and fault isolation.

Summary

Oracle Sharding is a marquee feature of Oracle Database 12c Release 2, which enables distribution, and replication of data across a pool of discrete Oracle databases that share no hardware or software. Oracle Sharding provides linear scalability, fault containment and geo-distribution benefits for suitable web-scale OLTP applications. It also supports on-premises, cloud and hybrid deployments. This paper included an in-depth coverage of the MAA best practices when deploying an Oracle Sharded Database. This paper also presented Oracle Sharding benchmark on Oracle Bare Metal Cloud and showcased the linear scaling and extreme data availability characteristics of Oracle Sharding.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Integrated Cloud Applications & Platform Services

Oracle Sharding MAA Best Practices
June 2017