3 Replicating Data Between On-Premises and Oracle Cloud Using Oracle GoldenGate

3-1 Configuring Oracle GoldenGate for Performance, Manageability, and Database Stability
3-1 Best Practices for Replication between Cloud and On-Premises Environments

4 Running Analytics in Oracle Analytics Cloud

4-1 Performance Impact on Business Intelligence Cloud Service Queries from Network Overhead
4-2 Latency Volumes for Oracle Analytics Cloud – Essbase Data Files
4-3 High-Memory Shapes for Oracle Analytics Cloud – Essbase Service Instances
4-3 HTTP Layer Cache Control Pragmas for Static Files
4-3 Validating Datasources and initblock During RPD Lift-and-Shift
4-4 Maintaining Minimum Number of Executors for Spark Dynamic Allocation
4-4 Tuning Spark Configuration to Work with Analytics
4-4 Optimizing the Buffer Size and Executor Memory for Spark SQL Queries
4-5 Optimizing Resource Usage by Disabling Spark History

5 Running Oracle Traffic Director on Oracle Compute Cloud Service Instances

5-1 Oracle Traffic Director Tuning for Performance
5-2 Placement of Oracle Compute Cloud Service Instances on Different Nodes for High Availability
5-2 HTTP Keep-Alive Timeout Parameter
5-2 Thread Pool Settings
5-3 Always Use Keep Alive Setting for Connections to Origin Servers
5-3 Static Files Cached at Oracle Traffic Director
5-4 Cache Headers for Static Files in the Web Tier
5-5 Compression of Text for Faster Request and Response Times
Running Node.js Applications on Application Container Cloud Service

- Single-Threaded Node for Performance
- Applications of Node.js on Application Container Cloud Service for High Availability
- Application Container Cloud Service AntiAffinity Setting for High Availability
- Application Container Cloud Service Rolling Restart Mode for High Availability during Upgrade
Preface

*Migrating to Oracle Cloud: Performance and Tuning Best Practices* includes best practices for migrating Compute systems, databases, Oracle GoldenGate, Oracle Business Intelligence Cloud Service, Oracle Essbase, JVMs, Oracle Traffic Director, and Node.js applications to Oracle Cloud.

The following best practices and parameter settings are based on our experience with various applications in the lab. We recommend that you thoroughly test these out for your application in your environment before going to production.

Topics:
- Audience
- Documentation Accessibility
- Related Resources
- Conventions

Audience

*Migrating to Oracle Cloud: Performance and Tuning Best Practices* is intended for Devops Engineers, System Administrators, Storage Administrators, Developers, Database Administrators, Business Intelligence (BI) Administrators, and Essbase Administrators who want to move their on-premises systems and resources to Oracle Cloud.

Documentation Accessibility


Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) or visit [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs) if you are hearing impaired.

Related Resources

For more information, see the Oracle Cloud website at [http://cloud.oracle.com](http://cloud.oracle.com).
The following text conventions are used in this document.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Running Workloads on Compute Cloud Service Instances with Oracle Linux

Some performance best practices should be followed when you are creating Compute resources like Compute Cloud Service Instances and Block Storage on Oracle Cloud. Apply the configurations listed in the following checklist before you start installing or setting up any applications on the Compute Cloud Service Instances:

## Quick Checklist

<table>
<thead>
<tr>
<th>Area</th>
<th>Optimizations</th>
<th>Persona (DevOps/Developer/Admin)</th>
</tr>
</thead>
</table>
| VM Size and Image  | • Garbage Collection Tuning Affected by Number of VCPUs in Compute Cloud Service Instance  
• Small Compute Shape and Heap Size for WebLogic Administration Server  
• Maximum Heap Size for Java Process | Developers  
Developers  
Developers                                                                 |
| Storage            | OS Bootable Storage Volume Using the Default Stripe Size When Creating a Volume Group  
Using SSD-Based Storage Volumes to Optimize I/O Bandwidth | DevOps Engineers, System Administrators, and Storage Administrators                                 |
| I/O                | DOS-Compatible Mode Disabled for Storage Volumes                              | DevOps Engineers, System Administrators, and Storage Administrators                                 |
| Generic Performance| • Placement of Services to Reduce Network Latency  
• Data Migration | DevOps Engineers and System Administrators  
DevOps Engineers, System Administrators, and Storage Administrators |
| High Availability  | High Availability and Instance Anti-affinity                                  | DevOps Engineers and System Administrators                                                        |
| Feature-specific   | • SSH Tunneling for Secure Connectivity and Two-Way Communication  
• Runtime Tools for Monitoring System and Processes  
• LUKS for File Encryption at Rest  
• Disabling IPv6 to Solve Connectivity Issues in Applications  
• Performing Additional Operating System Tuning When Using SPARC 300 | DevOps Engineers and Developers  
DevOps Engineers, System Administrators, and Developers  
DevOps Engineers and Developers  
DevOps Engineers and System Administrators  
DevOps Engineers and System Administrators |
Garbage Collection Tuning Affected by Number of VCPUs in Compute Cloud Service Instance

The number of VCPUs in a Compute Cloud Service instance affects the tuning of garbage collection. You must carefully tune the garbage collection threads in Oracle Private Cloud environment because the VM shapes in Oracle Private Cloud may differ from the on-premise source environment. The process of tuning garbage collection is the same between on-premise environment and Oracle Cloud.

If you have small heaps on large shapes, limiting the number of parallel threads can also help performance. A 2 GB heap cannot be effectively parallelized beyond about 8 garbage collection threads.

Best practice for:
Developers

Small Compute Shape and Heap Size for WebLogic Administration Server

You can place your WebLogic Administration Server in a smaller-shaped Compute Cloud Service instance.

Oracle WebLogic Server comes with Managed Servers as well as an Administration Server. In most cases the Administration Server is not running any large application, so there is no need to set a large maximum heap size for this server. If your topology has a separate Compute Cloud Service instance for the WebLogic Administration Server, then you should consider allocating a smaller memory shape for it.

Best practice for:
Developers

Maximum Heap Size for Java Process

Tune the maximum heap size for your Java process according to the maximum operating system memory available in the shape of your Compute Cloud Service instance.

The actual process memory utilized by the JVM process can be as high as 1.8 times to 2 times that of your maximum heap size defined by the \(-Xmx\) setting. Also, you need to take into account any other processes running in the same Compute Cloud Service Instance.

A large heap size may end up swapping in the OS, which can significantly degrade performance.
Best practice for:
Developers

OS Bootable Storage Volume

Use a bootable storage volume for a boot disk instead of local, ephemeral volumes.

While creating a Oracle Compute Cloud Service instance, you can have a bootable volume either on a local, ephemeral disk or on persistent block storage. Local drives are ephemeral drives. If you use them, you may lose data on the Oracle Compute Cloud Service instance restarts, and your orchestration may have to recreate the image, hence your restart time will be slower. An Oracle Compute Cloud Service instance reboot doesn't guarantee that the same ephemeral disk will be mounted.

See Creating a Bootable Storage Volume in Using Oracle Compute Cloud Service (IaaS).

Best practice for:
DevOps Engineers, System Administrators, and Storage Administrators

Using the Default Stripe Size When Creating a Volume Group

When binding multiple storage volumes into a volume group, changing the default stripe size and making it too small, such as 4KB, may lead to situation where striped data won’t be properly aligned with the rest of Oracle Cloud block storage infrastructure.

This could lead to noticeable I/O bandwidth degradation, especially for I/O-intensive workloads that involve random data access.

Oracle recommends that you use the default stripe size of 64KB when binding multiple storage volumes into a volume group.

Best practice for:
DevOps Engineers, System Administrators, and Storage Administrators

DOS-Compatible Mode Disabled for Storage Volumes

For better I/O performance, disable DOS-Compatible mode.

Creating disk partitions in DOS-Compatible mode can lead to a performance impact. Most disks, including those used in the Oracle Cloud, have 4096-byte sectors. When a partition is created in DOS-compatible mode, it attempts to start the partition at what
would have been the second cylinder of a disk with 512-byte sectors, which does not fall on an even physical sector boundary for a disk with 4096-byte sectors. Every logical read or write on such a partition ends up being two physical writes, with the first part of the logical sector in part of one physical sector and the second part in the next physical sector. The result is a doubling of physical I/O versus non-DOS compatible disks and a corresponding decrease in performance.

See Doc ID 2098513.1.

Best practice for:
DevOps Engineers, System Administrators, and Storage Administrators

Placement of Services to Reduce Network Latency

Reduced network traffic between components can improve performance. If network latency is critical for performance, then place all services in the same data center, if your account allows.

For example, for the Oracle E-Business Suite application, the database and the middle tier should not be in two different data centers.

Best practice for:
DevOps Engineers and System Administrators

Data Migration

If you plan on moving a large amount of data from an on-premises system to Oracle Cloud, the following tips can help improve the performance of data migration.

- Consider using parallel threads to move data in multiple pipes. For example, use multiple Secure Copy (SCP) processes to send multiple files in parallel as opposed to sending a single file in a single SCP process.
  
  If you send data in a single pipe, you may face various limitations, such as single threads read or write bandwidth or single session network cap. Using multiple threads or processes helps to solve it.

- Copying files into a latency volume gives better performance than copying files to a throughput volume.

- If you've configured Corente VPN, consider an alternative, yet secure approach to move big volumes of data, because Corente VPN may significantly slow down data transfer speed.
  
  Corente VPN works as designed for normal daily application data flows, but it can heavily affect initial, and often, large volume data transmission.

  You can use Oracle Storage Server (OSS) as intermediate storage connecting directly to Oracle Cloud, and not through Corente VPN. OSS endpoints are already secured, because OSS uses the HTTPS protocol.
High Availability and Instance Anti-affinity

You can set high availability for an instance as active. This ensures that if the instance goes down unexpectedly, it is re-created automatically.

Instance anti-affinity provides a fallback for cases where the node that your instance is on fails. In such cases, even if high availability is specified for all instances on the node, instances can't be re-created until the node failure is resolved. Instance anti-affinity allows you to create an instances on a different physical nodes. By specifying instance anti-affinity for instances in the same cluster, even if one instance becomes unavailable due to physical node failure, other instances in the cluster aren't affected.

Note:

If you require very low latency across instances, then don’t use instance anti-affinity. Instead, place the instances on the same node.

Example:

```
"ha_policy": "active",
..............
..............
"relationships": [ 
    { 
      "instances": [ 
        "A",
        "B"
      ],
      "type": "different_node"
    }
  ]
```

In this example, two instances A and B are created in a single orchestration. The example shows the high-availability policy (ha_policy) specified as active for an instance. In the relationships section of the orchestration, the relationship type specified is different node, indicating that instance A and instance B should be created on different nodes.

Note:

If you need low latency across instances, then to ensure that instances are created on the same node, specify the relationship type as same_node.

See About High-Availability Policies in an Orchestration and Object Type: Launch Plan in Using Oracle Compute Cloud Service (IaaS).
Best practice for:
DevOps Engineers and System Administrators

SSH Tunneling for Secure Connectivity and Two-Way Communication

Use SSH tunneling to set up ad-hoc and secure connectivity between an on-premises tool, such as SQL Developer and the services running on Oracle Cloud. You can also use two-way SSH tunneling when there's a need for a two-way communication and a VPN connection is not ready. Use an autossh package to establish a two-way channel that is more persistent and self-healing in case of a network failure. Autossh package also monitors the state of the channel and also automatically restarts them if needed.

All applications’ traffic is transported through an SSH session channel, and there’s no need to open more ports on Oracle Cloud VMs other than the SSH port (22).

One-Way SSH Tunneling (for secure communication from an on-premise application to Oracle Cloud)

Use one-way SSH tunneling to create an SSH tunnel connecting the SQLDeveloper tool on a client desktop to a database running in Oracle Cloud.

Syntax (Local)

```bash
ssh -L <local_host>:<local_host_port>:<remote_host>:<remote_host_port> <opcuser>@<IPAdress_of_OPC>
```

Example

```bash
ssh -L localhost:1521:localhost:1521 oracle@x.x.x.x
```

Two-Way SSH Tunneling (interim solution while corporate VPN is not yet set up)

Use two-way SSH tunneling to immediately connect to Oracle Cloud while the corporate VPN is still being configured.

Syntax

```bash
ssh -L -R <local_host>:<local_host_port>:<remote_host>:<remote_host_port> <opcuser>@<IPAdress_of_OPC>
```

Example

The following example shows how the internal traffic coming to port 9002 on Server A is forwarded to port 9001 on Server B:

```bash
ssh -L localhost:9002:localhost:9001 oracle@x.x.x.x
```

The following example shows how the internal traffic coming to port 9002 on Server B is forwarded to port 9001 on Server A:

```bash
ssh -R localhost:9002:localhost:9001 oracle@x.x.x.x
```
Two-Way SSH Tunneling with autossh Package (self-healing in case of network failure)

Use two-way SSH tunneling with autossh package to achieve extra stability to overcome possible network failures and restart SSH channels automatically in case of a network failure, in addition to connect to Oracle Cloud while the corporate VPN is still being configured.

Example

```
autossh -M 0 -q -f -N -o "ServerAliveInterval 60" -o "ServerAliveCountMax 3" -L localhost:9002:localhost:9001 oracle@x.x.x.x
autossh -M 0 -q -f -N -o "ServerAliveInterval 60" -o "ServerAliveCountMax 3" -R localhost:9002:localhost:9001 oracle@x.x.x.x
```

**Best practice for:**
DevOps Engineers and Developers

Runtime Tools for Monitoring System and Processes

Continuous monitoring of system and application resource usage is critical for any production system. It's important to install some form of monitoring tools inside your Compute Cloud Service instance and ensure that such tools are running in the background and saving logs for certain periods of time.

Oracle recommends that you use at least some basic monitoring tools, such as oswatcher, for monitoring system and process behaviors. See [Doc ID 301137.1](https://www.oracle.com).

**Best practice for:**
DevOps Engineers, System Administrators, and Developers

LUKS for File Encryption at Rest

If you need to have encryption at rest in a raw Compute Cloud Service instance, you can use the Linux Unified Key Setup (LUKS) technique.

Many PaaS services have encryption at rest already. Sometimes it's required to set up an encrypted data landing page in a raw Compute Cloud Service instance. That's where LUKS can be used.

**Best practice for:**
DevOps Engineers and Developers
Disabling IPv6 to Solve Connectivity Issues in Applications

Some applications may behave unpredictably when both IPv6 and IPv4 are enabled in your operating system.

For example, the curl utility may fail with a timeout, and it works only when you use IPv4 (curl -4). Another example may be a custom Java application cluster that is having trouble communicating between its nodes unless you explicitly set the IPv4 usage preference (-Djava.net.preferIPv4Stack=true).

Unless your application requires IPv6, you don't need to enable IPv6. Enabling IPv6 can cause connectivity issues in your application. A suggested workaround is to disable IPv6. Following are the sample commands to temporarily disable IPv6:

```bash
# sysctl -w net.ipv6.conf.all.disable_ipv6=1
# sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

To make this change persistent, edit the sysctl.conf file located in the /etc directory as shown in the following sample and reboot the system.

```bash
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Best practice for:
DevOps Engineers and System Administrators

Performing Additional Operating System Tuning When Using SPARC 300

If you encounter high network latency in SPARC 300 dedicated compute, you can improve maximum bandwidth by changing the TCP configuration inside the virtual machine.

In addition, you must restart the SSH service every time TCP tuning is applied.

Tuning the TCP send and receive buffers may improve the network throughput.

Example:
The following example lists the sample commands that you can use to perform these additional tuning:

```bash
ipdadm set-prop -p recv_buf=4194304 tcp
ipdadm set-prop -p max_buf=4194304 tcp
ipdadm set-prop -p send_buf=4194304 tcp
svcadm restart ssh
```

In the preceding example, you should evaluate and replace the exact values on a case-to-case basis.
Best practice for:
DevOps Engineers and System Administrators

Using SSD-Based Storage Volumes to Optimize I/O Bandwidth

You can optimize the I/O Compute workloads by creating storage volumes on Solid-State Drive (SSD) filers instead of a Hard Disk Drive (HDD) filer.

Some regional sites offer SSD-based filers in addition to HDD-based filers to create storage volumes. In addition to high I/O performance, such SSD volumes provide same benefits in regard to data durability, failover, and HA deployment.

If your region doesn’t offer an SSD-based filer option, or if you need to deploy your application using a non-persistent storage option, then use the local NVMe SSD disks available in high I/O shapes to optimize the I/O workload. If you choose to use non-persistent storage option, then you must devise a proper strategy to have data durability, failover, or HA for your data.

Best practice for:
DevOps Engineers and System Administrators
Running Oracle Databases on Compute Cloud Service Instances

Optimizations can help database performance in Oracle Cloud. You can use a database instance as part of a database service or create a database manually in a Compute Cloud Service instance. If you create your own database, you need to apply certain optimizations.

Quick Checklist

<table>
<thead>
<tr>
<th>Area</th>
<th>Optimizations</th>
<th>Persona (DevOps/Developer/Admin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System Kernel</td>
<td>Oracle Database Kernel Parameter Tuning</td>
<td>Database Administrators</td>
</tr>
<tr>
<td>Generic Performance</td>
<td>• Configuration of HugePages for Oracle in Linux</td>
<td>Database Administrators</td>
</tr>
<tr>
<td></td>
<td>• Oracle Row Prefetching</td>
<td>Developers</td>
</tr>
<tr>
<td></td>
<td>• Database Performance Bottleneck with UTF8 Character Set</td>
<td>Database Administrators</td>
</tr>
<tr>
<td>I/O</td>
<td>• Oracle DB Redo Logs and Data Files Mapped to Latency Volumes</td>
<td>Database Administrators</td>
</tr>
<tr>
<td></td>
<td>• Oracle init.ora I/O Performance Parameters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Improving Block I/O Performance</td>
<td></td>
</tr>
<tr>
<td>Backup and Recovery</td>
<td>• Disaster Recovery</td>
<td>Database Administrators</td>
</tr>
<tr>
<td></td>
<td>• RMAN Backup</td>
<td></td>
</tr>
<tr>
<td>Feature-specific</td>
<td>• Recommended DBaaS Software Edition</td>
<td>Database Administrators</td>
</tr>
<tr>
<td></td>
<td>• Oracle init.ora Diagnostics and Tuning Pack for Enterprise Edition</td>
<td></td>
</tr>
</tbody>
</table>

Oracle Database Kernel Parameter Tuning

Tune database-related kernel parameters.

See Configuring Kernel Parameters and Resource Limits in Oracle Database Quick Installation Guide.

Many of these parameter values are the defaults for DBaaS, but you may have to modify the values if you manually install the database on IaaS.
Configuration of HugePages for Oracle in Linux

Enable the HugePages feature for an Oracle E-Business Suite database so the operating system can support memory pages larger than the default (usually 4 KB). Other application databases might also benefit with HugePages.

The HugePages feature is integrated into the Linux kernel 2.6. Using very large page sizes can improve system performance by reducing the amount of system resources required to access page table entries. HugePages is useful for both 32-bit and 64-bit configurations. HugePage sizes vary from 2 MB to 256 MB, depending on the kernel version and the hardware architecture. For Oracle Databases, using HugePages reduces the operating system maintenance of page states and increases the TLB (Translation Lookaside Buffer) hit ratio.

See Overview of HugePages in Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems or HugePages on Oracle Linux 64-bit (Doc ID 361468.1).

Oracle Row Prefetching

Query performance may benefit if you increase the number of rows fetched from the database by each fetch call.

Increasing the Oracle row prefetch size helps in performance if you have relatively large latency between the database and middle tier (more than 20 msec) and your application is fetching more than 100 rows in result sets for your queries.

Increasing the Oracle row prefetch can help for an OLAP application and not necessarily for OLTP applications.

The default value of the JDBC prefetch size is 10. You can increase it to a large value, but increasing beyond 1000 is not recommended because the memory footprint in the middle tier can grow. See Class OracleDriver in the Oracle Database JDBC Java API Reference.
Database Performance Bottleneck with UTF8 Character Set

Use a different character set than the UTF8 character set, unless you need it, because UTF8 has a performance bottleneck.

Depending on your workload and how CPU intensive it is, the overhead of using a multibyte character set like UTF8 can be significant compared to using other character sets.

> **Best practice for:**
> Database Administrators

Oracle DB Redo Logs and Data Files Mapped to Latency Volumes

For improved I/O performance, ensure that redo logs and temporary data files are placed in latency volumes.

DBaaS by default uses latency volumes for redo logs and temporary data files. If you install Oracle Database Software Edition manually on IaaS, place the redo and temporary files in latency volumes. See [Store Data Files Separate from Redo Log Files](#) in Oracle Database Administration Guide.

> **Best practice for:**
> Database Administrators

Oracle init.ora I/O Performance Parameters

Ensure that the `filesystemio_options=SETALL` parameter is set for better database I/O performance.

DBaaS has this parameter setting by default. If you manually create a database, you need to ensure this setting is present.

> **Best practice for:**
> Database Administrators

Disaster Recovery

Data Guard and Active Data Guard provide disaster recovery (DR) for databases with recovery time objectives (RTOs) that cannot be met by restoring from backup.
See **Disaster Recovery to the Cloud with Data Guard and Active Data Guard**.

**RMAN Backup**

Use RMAN parallelism and compression to speed up cloud backups and restores.

See Doc ID **2078576.1**

- Set parallelism (for instance, 10):
  ```
  CONFIGURE DEVICE TYPE SBT PARALLELISM 10;
  ```

- Do not enable tracing (that is, no `_OPC_TRACE_LEVEL` parameter specified in the Oracle Cloud configuration file).

- If the database size is big in TBs, then use the `SECTION SIZE` option in the backup command to parallelize big file tablespace data files. For instance:
  ```
  BACKUP DEVICE TYPE sbt DATABASE TAG 'MYBACKUP' SECTION SIZE 100G;
  ```

  The preceding command will allocate 10 channels and split big files into 100 GB backup pieces.

See **Best Practices to Optimize Cloud Backup and Restore Rates** in *Using Oracle Database Backup Cloud Service*.

---

**Recommended DBaaS Software Edition**

Use the DBaaS High Performance version for production because it comes with a Diagnostics and Tuning pack that helps in debugging performance issues.

While creating a DBaaS instance, for **Software Edition** choose the **Enterprise Edition - High Performance** version, or above, for production installations. Standard Edition lacks AWR and other diagnostic and tuning features.

---

**Oracle init.ora Diagnostics and Tuning Pack for Enterprise Edition**

Verify that `control_management_pack_access = DIAGNOSTIC+TUNING`, to enable the Diagnostics and Tuning pack in Enterprise Edition.

If you create the database manually, you may have to change this setting.
Improving Block I/O Performance

If you notice poor I/O performance on the Compute storage volumes of your Database Cloud Service database deployment, you can download and run a script to enable the multi-page ring support feature of Linux to improve performance. Some of the latest DBaaS images may not have the multi-page ring support feature enabled.

To enable the multi-page ring support feature, see Improving Block I/O Performance by Enabling Multi-Page Ring Support in the Linux OS.
Oracle GoldenGate is a comprehensive software package for real-time data integration and replication in heterogeneous IT environments. The product set enables high-availability solutions, real-time data integration, transactional change data capture, data replication, transformations, and verification between operational and analytical enterprise systems.

Quick Checklist

<table>
<thead>
<tr>
<th>Area</th>
<th>Optimizations</th>
<th>Persona</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Performance</td>
<td>Configuring Oracle GoldenGate for Performance, Manageability, and Database Stability</td>
<td>DevOps Engineers</td>
</tr>
<tr>
<td>Feature-specific</td>
<td>Best Practices for Replication between Cloud and On-Premises Environments</td>
<td>DevOps Engineers</td>
</tr>
</tbody>
</table>

Configuring Oracle GoldenGate for Performance, Manageability, and Database Stability

Configure Oracle GoldenGate for the best performance, simple manageability, and stability for Oracle databases.

See Oracle GoldenGate Performance Best Practices.

Best practices for:
DevOps Engineers

Best Practices for Replication between Cloud and On-Premises Environments

Follow best practices for replication with Oracle GoldenGate.

See Doc ID 1996653.1.
### Running Analytics in Oracle Analytics Cloud

Oracle Analytics Cloud comprise of Oracle BI Cloud Service and Oracle Essbase Cloud. Some performance best practices should be followed for running analytics in Oracle Analytics Cloud.

#### Quick Checklist

<table>
<thead>
<tr>
<th>Area</th>
<th>Optimizations</th>
<th>Persona (DevOps/Developer/Admin)</th>
</tr>
</thead>
</table>
| Generic Performance| - Performance Impact on Business Intelligence Cloud Service Queries from Network Overhead  
- Validating Datasources and initblock During RPD Lift-and-Shift  
- Maintaining Minimum Number of Executors for Spark Dynamic Allocation  
- Tuning Spark Configuration to Work with Analytics  
- Optimizing the Buffer Size and Executor Memory for Spark SQL Queries  
- Optimizing Resource Usage by Disabling Spark History | - Developers and BI Administrators  
- Developers and BI Administrators  
- Administrators  
- Big Data Cloud Service Administrators and BI Administrators  
- Big Data Cloud Service Administrators and BI Administrators  
- Big Data Cloud Service Administrators and BI Administrators |
| I/O                | • Latency Volumes for Oracle Analytics Cloud – Essbase Data Files  
• High-Memory Shapes for Oracle Analytics Cloud – Essbase Service Instances         | Developers and Essbase Administrators                                                               |
| Feature-specific   | HTTP Layer Cache Control Pragmas for Static Files                              | Developers                                                                                           |

### Performance Impact on Business Intelligence Cloud Service Queries from Network Overhead

If your Oracle Business Intelligence Cloud Service instance points to a database that is on a different data center and has longer latency, then increase the set array size inside the repository to 1000.
Your Business Intelligence Cloud Service instance pointing to a database on a different data center with longer latency can have a performance impact on the queries due to network overhead. To minimize the performance impact, you can reduce the network round-trips between the Oracle BI server and the data source, which reduces the response time of the queries.

Best practice for:
Developers and BI Administrators

Latency Volumes for Oracle Analytics Cloud – Essbase Data Files

To improve I/O latency and performance for Oracle Analytics Cloud – Essbase queries, use latency-optimized volumes for your data files.

Oracle Cloud provides two types of block storage options: one storage is optimized for throughput, and the other one is optimized for latency. Because Oracle Analytics Cloud – Essbase is very I/O sensitive, Oracle recommends that you use the Latency option.

Best practice for:
Developers and Essbase Administrators
High-Memory Shapes for Oracle Analytics Cloud – Essbase Service Instances

Oracle Analytics Cloud – Essbase would greatly benefit from a high-memory shape in its Oracle Compute Cloud Service instance, so use high-memory shape Oracle Compute Cloud Service instances if possible.

Oracle Analytics Cloud – Essbase uses buffered I/O and not direct I/O, so a bigger OS file buffer cache can improve I/O latency and performance for Oracle Analytics Cloud – Essbase queries. A high-memory shape for an Oracle Compute Cloud Service instance allows more memory, hence more room for the buffer cache.

Best practice for:
Developers and Essbase Administrators

HTTP Layer Cache Control Pragmas for Static Files

If you manage an instance of Oracle BI Cloud Service, ensure that you have cache control pragmas set for your static files to be cached on the browser. Caching of static files on the client browser help in reducing the network latency to render the user interface.

Typically the images, CSS files, and JavaScript files are considered to be static files.

See Oracle Traffic Director Best Practices for Oracle Cloud.

These cache control pragmas are not required in a managed PaaS environment.

Best practice for:
Developers

Validating Datasources and initblock During RPD Lift-and-Shift

During the lift-and-shift of Oracle Business Intelligence (BI) repository (RPD), some datasources specified in an on-premises RPD may not be reachable or may be irrelevant in an Oracle Cloud setup, and this can slow down performance significantly.

During an RPD lift-and-shift, validate datasource and corresponding initblocks.

For example, in nqserver.log, look for the following error:

"[13011] Query for Initialization Block 'Loglevel' has failed. ["
Best practice for:
Developers and BI Administrators

Maintaining Minimum Number of Executors for Spark Dynamic Allocation

When using Oracle Business Intelligence Cloud Service with Oracle Big Data Cloud Service, by default Oracle Big Data Cloud Service allocates 0 executor. The executors are created on demand that adds more than 5 seconds to the time that the first SQL statement takes to execute when running SparkSQL with no executor allocated. So, maintaining minimum number of executors helps the SQL.

Define the initial or minimum number of executors for spark dynamic allocation in the Advanced spark-thrift-sparkconf section in Ambari UI.

```
spark.dynamicAllocation.initialExecutors = <# of nodes in the Spark cluster>
spark.dynamicAllocation.minExecutors = <# of nodes in the Spark cluster>
```

Best practice for:

Tuning Spark Configuration to Work with Analytics

When you use Oracle Business Intelligence Cloud Service with Oracle Big Data Cloud Service, the Oracle Business Intelligence Cloud Service datasets are typically not as large as other Spark scenarios. Reducing the shuffle partitions helps the performance of the Spark queries.

For configuring Spark in Ambari UI, specify the shuffle partitions in the Custom spark-thrift-sparkconf section.

```
spark.sql.shuffle.partitions = 4
```

Best practice for:
Big Data Cloud Service Administrators and BI Administrators

Optimizing the Buffer Size and Executor Memory for Spark SQL Queries

Spark SQL queries may fail because of insufficient buffer size and executor memory, insufficient capacity of Spark Kryo serializer, and short timeout setting for long queries.
To optimize the buffer size and executor memory for Spark SQL:

Set the following in the Custom spark-thrift-sparkconf section of Spark configuration in the Ambari UI:

```
spark.kryoserializer.buffer=128m
spark.kryoserializer.buffer.max=2047m
spark.network.timeout=600
```

Configure a proper Spark executor JVM heap size to ensure that sufficient memory is being allocated. You can increase this if the executor nodes have highly available memory.

```
spark.executor.memory=8g
```

---

**Best practice for:**

Big Data Cloud Service Administrators and BI Administrators

---

### Optimizing Resource Usage by Disabling Spark History

If you don't need to maintain Spark history data through event logging, consider disabling it to save disk and CPU usage.

For example, when Spark SQL request rate is high, Spark history data can pile up in the disk, and Spark history can occupy at least one core of the CPU to process the data. Disable the event logging to save the disk space and to optimize the processing power of the CPU.

To disable event logging, set the following in the Advanced spark-defaults and Advanced spark-thrift-sparkconf sections of Spark configuration in the Ambari UI:

```
spark.eventLog.enabled=false
```

---

**Best practice for:**

Big Data Cloud Service Administrators and BI Administrators
Running Oracle Traffic Director on Oracle Compute Cloud Service Instances

Oracle Traffic Director (OTD) is an on-board, highly available Application Delivery Controller (ADC) to optimize application-to-application communication. Several performance best practices should be followed for OTD on Oracle Compute Cloud Service.

Quick Checklist

<table>
<thead>
<tr>
<th>Area</th>
<th>Optimizations</th>
<th>Persona (DevOps/Developer/Admin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Performance</td>
<td>Oracle Traffic Director Tuning for Performance</td>
<td>DevOps Engineers</td>
</tr>
<tr>
<td>High Availability</td>
<td>Placement of Oracle Compute Cloud Service Instances on Different Nodes for High Availability</td>
<td>DevOps Engineers</td>
</tr>
<tr>
<td>Feature-specific</td>
<td>• HTTP Keep-Alive Timeout Parameter</td>
<td>DevOps Engineers</td>
</tr>
<tr>
<td></td>
<td>• Thread Pool Settings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Always Use Keep Alive Setting for Connections to Origin Servers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Static Files Cached at Oracle Traffic Director</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Cache Headers for Static Files in the Web Tier</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Compression of Text for Faster Request and Response Times</td>
<td></td>
</tr>
</tbody>
</table>

Oracle Traffic Director Tuning for Performance

For best performance, do the generic Oracle Traffic Director tuning.

See Tuning Oracle Traffic Director for Performance in Oracle Traffic Director Administrator's Guide.

Best practice for:

DevOps Engineers
Placement of Oracle Compute Cloud Service Instances on Different Nodes for High Availability

Placement of Oracle Traffic Director Compute Cloud Service instances on different physical nodes using the orchestration flags ensures high availability during node failures.

Such placement provides true HA capabilities.

While provisioning Oracle Traffic Director node Compute Cloud Service instances, make sure that the type attribute under relationships in a launch plan is used to increase high availability.

Best practice for:
DevOps Engineers

HTTP Keep-Alive Timeout Parameter

Make sure HTTP keep-alive is enabled to remove the cost of establishing connections in the high-latency network path.

In the Administration Console, select Advanced Settings > HTTP > KeepAlive and make sure HTTP keep-alive is enabled. Set the KeepAlive time as follows:

- 61 seconds if OTD supports a web-based UI that is not opening a large number of connections per user
- 5 seconds for most REST invocations and most JET-based UIs

```xml
<keep-alive>
  <timeout>5</timeout>
</keep-alive>
```

Establishing a connection is an expensive process that involves round trips between the client and server. The keep-alive timeout parameter sets the client-server connections to be preserved for the traffic to flow without connection cost. Too many keep-alive connections with high timeout values would let new users beyond threshold to starve for resources. You need to tune this for the workload requirements.

Best practice for:
DevOps Engineers

Thread Pool Settings

Tune the thread pool to allow the maximum number of threads that can run in parallel to avoid exhaustion of resources.
Queues are used to avoid errors during burst requests (that is, spikes). The queue size is defined to let the requests wait until a free thread is available to serve. This is done to avoid errors when all the threads are busy, avoid exhaustion of resources (throttling), and make pre-initialized threads available to serve requests.

In the Administration Console, select Advanced Settings > Thread Pool and tune as follows:

- Enable the thread pool.
- Change max-threads to 20480 (make sure that the OS File Descriptor number is tuned).
- Change queue-size to 3000.

In the instance server.xml file, this looks like:

```xml
<thread-pool>
  <queue-size>3000</queue-size>
  <max-threads>20480</max-threads>
</thread-pool>
```

---

**Always Use Keep Alive Setting for Connections to Origin Servers**

The keep_alive setting reduces response times and improves throughput by removing the cost of establishing a new connection for every request.

Connection establishment and SSH handshakes are costly operations. Maintaining a pool of connections between OTD and origin servers reduces the overall cost of serving client requests. Connections can be reused across requests from the same and different clients.

When the origin server is WebLogic Server, the connection between OTD and WLS will always be persisted and kept alive.

1. In the Administration Console, select Virtual Servers > Config > Routes > Default Route.
2. Set Always Use Keep Alive to true.

---

**Static Files Cached at Oracle Traffic Director**

Use static file caching to reduce the network and server processing cost of transferring static artifacts from the origin server to Oracle Traffic Director (OTD).
Static file caching allows OTD to cache static artifacts from origin servers on the web server to avoid a round trip to the origin server for every such request. Because the caching happens in-memory, the sizing has to be appropriate to avoid exhaustion of resources for other functions.

You can use OTD to cache static artifacts. Adjust the cache size to be able to cache the objects you want:

1. In the Administration Console, select **Advanced Settings > Cache**.
2. Make sure the cache is enabled and set the following values:
   - **Heap Size** to 104857600
   - **Heap Object size for caching** to 4194304
   - **Replacement algorithm** LRU
   - **Max Entries** to 4096

Best practice for: DevOps Engineers

### Cache Headers for Static Files in the Web Tier

Static file headers improve processing time and reduce network congestion between the web server and origin servers.

Caching static artifacts from the origin servers in the web tier helps with improved request processing. Configuring the right artifacts and the expiry of those artifacts is key.

You can enable caching and add cache headers like this:

```xml
<If $uri =~ ©.(js|css|woff)$©
    ObjectType fn="proxy-cache-config" id="catchcompressjaandcss" enabled="true" cache-https-response="true" compression="true"
    ObjectType fn="set-cache-control" control="public,max-age=31536000"
</If>
<If $uri =~ ©.(?:gif|jpe?g|png|bmp|ico)$©
    ObjectType fn="proxy-cache-config" id="cacheimages" enabled="true" cache-https-response="true"
    ObjectType fn="set-cache-control" control="public,max-age=31536000"
</If>
```

Best practice for: DevOps Engineers
Compression of Text for Faster Request and Response Times

You can use compression to reduce the size of text contents, resulting in reduced network transfer time for requests and responses.

Typically, the network path between the client and the web server is the slowest in the topology because it represents clients residing in the public Internet or WAN. You need to reduce the number of round trips between these two components.

Compression works well for text content and results in reduction of content size. This shouldn't be configured for images and binary files, which are already compressed, or would not benefit from compression. Because compression uses CPU cycles, it is a trade-off between CPU versus network resources.

OTD should do response compression (when the client indicates that it can receive a compressed response).

You can enable compression of resources (except for images, PDF files, and SWF) like this:

```html
<If $uri !~ '\.(?:gif|jpe?g|png|bmp|ico)$' and $uri !~ '\.(pdf)$' and $uri !~ '\.swf$'
    Output fn="insert-filter" id="response-compression-rule"
    filter="http-compression"
</If>
```

Best practice for:
DevOps Engineers
Running Node.JS Applications on Application Container Cloud Service

Some best practices apply for Node.js applications on Application Container Cloud Service in Oracle Cloud.

Quick Checklist

<table>
<thead>
<tr>
<th>Area</th>
<th>Optimizations</th>
<th>Persona (DevOps/Developer/Admin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Performance</td>
<td>Single-Threaded Node for Performance</td>
<td>Developers</td>
</tr>
<tr>
<td>High Availability</td>
<td>• Applications of Node.js on Application Container Cloud Service for High Availability</td>
<td>Developers</td>
</tr>
<tr>
<td></td>
<td>• Application Container Cloud Service AntiAffinity Setting for High Availability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Application Container Cloud Service Rolling Restart Mode for High Availability during Upgrade</td>
<td></td>
</tr>
</tbody>
</table>

Single-Threaded Node for Performance

For best performance, Node.js applications should never make blocking, sync calls. For example, never do sync file system calls like `readFileSync`. Instead, do `readFile`.

Best practice for:

Developers

Applications of Node.js on Application Container Cloud Service for High Availability

To support high availability for the Node.js part of a service, run applications of Node.js on the Application Container Cloud Service.

See Create Sample Node.js Applications in the Oracle Application Container Cloud Service documentation.
Best practice for:
Developers

Application Container Cloud Service AntiAffinity Setting for High Availability

To ensure high availability against node failure, use the `requiresAntiAffinity` setting provided by ACCS.

The following example is from `manifest.json`:

```json
{
  "runtime": {
    "majorVersion": "6.3"
  },
  "command": "node server.js",
  "startupTime": 30,
  "shutdownTime": 30,
  "isClustered": false,
  "requiresAntiAffinity": true,
  "release": {},
  "notes": "NodeJS Sample Application"
  "mode": "rolling"
}
```

Best practice for:
Developers

Application Container Cloud Service Rolling Restart Mode for High Availability during Upgrade

To support high availability during upgrade, use the rolling restart mode provided by ACCS.

Make sure instances are not stuck in start or stop. Use appropriate start and stop times in `manifest.json`.

Example from `manifest.json`:

```json
{
  "runtime": {
    "majorVersion": "6.3"
  },
  "command": "node server.js",
  "startupTime": 30,
  "shutdownTime": 30,
  "isClustered": false,
  "requiresAntiAffinity": true,
  "release": {},
  "notes": "NodeJS Sample Application"
}
```
"mode": "rolling"
}

Best practice for:

Developers