

Oracle  
**Primavera Unifier**  
**Performance and Sizing Guide for On-Premises**

**Version 22**  
December 2023

Oracle Primavera Unifier Performance and Sizing Guide for On-Premises

Copyright © 1998, 2023, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

# Contents

---

<b>Overview of the Performance and Sizing Guide .....</b>	<b>5</b>
<b>Architecture Overview .....</b>	<b>7</b>
<b>Performance and Scalability Considerations.....</b>	<b>9</b>
Vertical Scaling .....	9
JVM Heap Sizes .....	9
Hardware Upgrade .....	9
Horizontal Scaling.....	10
Application Scaling and Clustering.....	10
Database Scaling and Clustering.....	12
Database File Storage.....	12
Operating System Upgrade .....	12
<b>Deployment Considerations .....</b>	<b>12</b>
Deployment Categories.....	13
Configuration for Deployment Categories.....	13
<b>Network Bandwidth Estimation .....</b>	<b>17</b>
<b>Factors that Affect Application Performance.....</b>	<b>19</b>
Configuration, Hardware, and Environment Factors.....	19
Other Actions that Affect Performance .....	19
<b>Conclusion .....</b>	<b>20</b>
<b>Frequently Asked Questions .....</b>	<b>20</b>



# Overview of the Performance and Sizing Guide

---

This document provides administrators guidance on planning product deployment and estimating of hardware and software requirements, and it also provides recommendations for deploying small-scale to large-scale configurations, for the product.

Based on the number of end-users, three deployment scenarios (Small, Medium, and Large) are considered and recommendations for each are provided in this guide. See **Deployment Categories** (on page 13) for more details.

---

**Note:** The recommendations in this guide must be considered as guidance for planning product deployment.

---

## Assumptions

The following assumptions are made in this document:

- ▶ Leveraging a high-availability environment for deployment.
- ▶ Following database-specific best practices for high availability, backup, and recovery.

---

**Note:** Hardware and Software requirements for load balancers and network devices are beyond the scope of this document.

---



# Architecture Overview

---

Unifier is a Java 2 Enterprise Edition (J2EE platform) web application. Unifier uses J2EE platform specification to build flexible and scalable cross-platform solution. The J2EE platform consists of a set of industry-standard services, APIs, and protocols that collectively provide the functionality needed for developing multi-tiered, web-based, and enterprise applications.

The division of tiers enables Unifier to scale according to the performance demands. The main tiers of Unifier are:

- ▶ **Presentation tier**

A web server layer rendering HTML, JavaScript, and so forth that presents a feature-rich user interface accessible through various supported browsers.

- ▶ **Middle tier**

Business logic for Unifier and Unifier Services.

- ▶ **Data tier**

A standalone, or clustered, RDBMS and supported Content Repository environment that uses Java Database Connectivity (JDBC) to integrate with the middle tier.





# Performance and Scalability Considerations

---

While there are multiple ways to achieve the desired performance and scalability levels in Unifier, the recommended performance considerations can be grouped into two categories:

- ▶ Vertical scaling
- ▶ Horizontal scaling

There are several advantages and disadvantages for each category. Your organization can decide which category to use based on the:

- ▶ Desired level of performance
- ▶ Availability requirements
- ▶ Short-term or long-term outlook of system usage
- ▶ Number of concurrent users

## In This Section

---

Vertical Scaling.....	9
JVM Heap Sizes.....	9
Hardware Upgrade.....	9
Horizontal Scaling.....	10
Operating System Upgrade.....	12

## Vertical Scaling

Vertical scaling involves adding additional resources or upgrading resources on an existing server. Vertical scaling is a preferred approach if the application bottlenecks are related to processor and memory.

## JVM Heap Sizes

The application objects (such as shells, companies, and so forth) are stored in the Java Virtual Machine (JVM) Heap. Most of these objects are short-lived and are periodically cleaned up by the JVM garbage collection mechanism.

As the concurrent users increases, the number of objects in Heap increases, and performance and scalability will be affected due to the congestion and resulting higher frequency of garbage collections.

Increasing application JVM Heap (with adequate physical memory provisioned) is an efficient way to achieve the desired performance and scalability. See the **Deployment Categories** (on page 13) section of this guide for guidelines on configuring the optimal Java heap size.

## Hardware Upgrade

The desired performance and scalability can be achieved by upgrading the CPU through:

- ▶ Adding extra cores
- ▶ Adding physical memory
- ▶ Upgrading to faster I/O devices

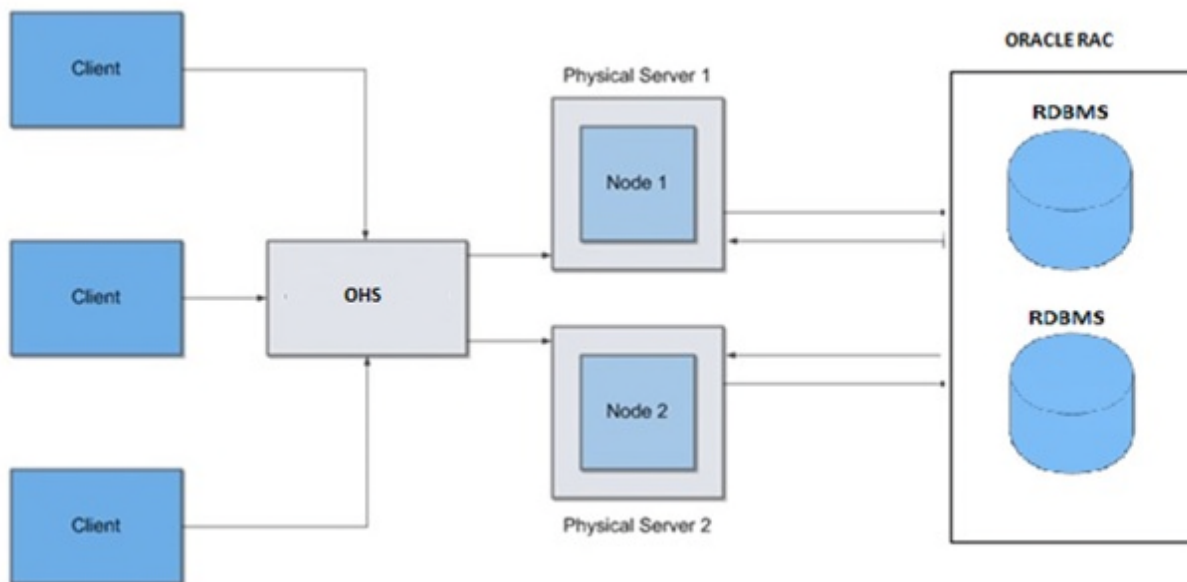
While vertical scaling is easier to achieve, it does not address the availability issue completely. Horizontal scaling can help achieve a higher availability.

## Horizontal Scaling

As the usage of application grows within the organization, additional server nodes can be added to an existing application server-cluster to handle the increased demand.

For high availability requirements, horizontal scaling is the better option.

The following figure explains how to scale a deployment:



## Application Scaling and Clustering

As the usage of application grows within the organization, adding additional server nodes is the best way to achieve the required performance and scalability.

If the application usage model exhibits seasonality, or periodic variations (for example, the average load on the system quadruples during month-end closing, or a plant may close for a week every quarter for maintenance), then consider adding, or removing, application server nodes in order to manage seasonality usage spikes.

To mitigate the risk of degraded performance and undesired downtime, it is crucial to understand the business cycles of your organization and to plan for the required level of performance, availability, and scalability.

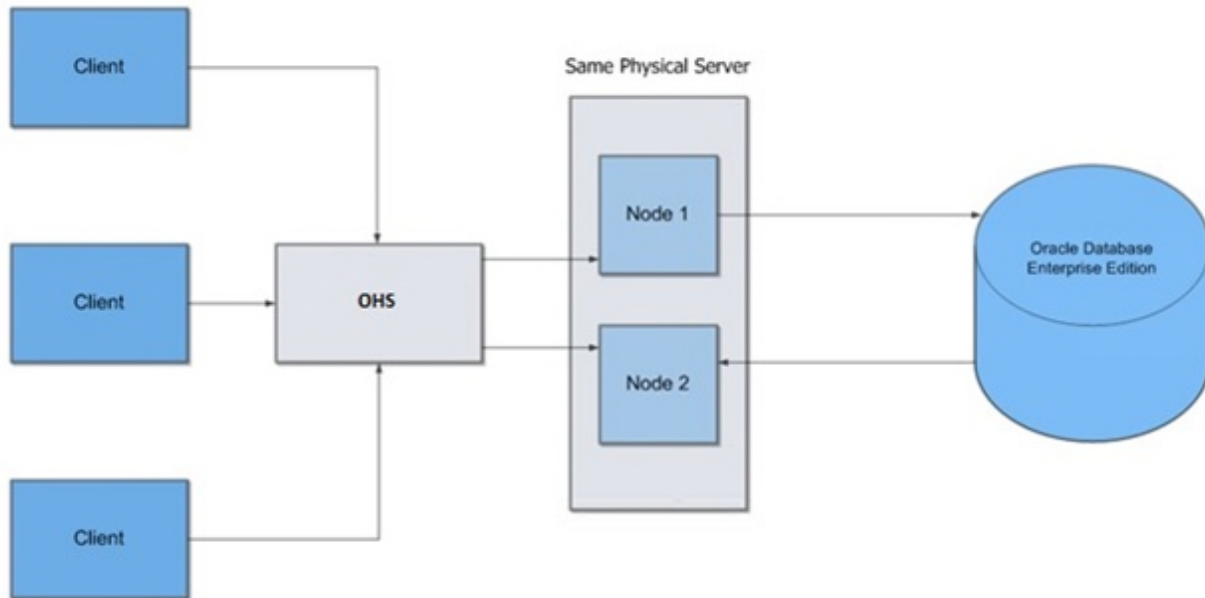
There are two ways to add application server nodes in a deployment:

- ▶ Vertical clustering
- ▶ Horizontal clustering

### Vertical clustering

In cases where you observe that the application transaction response times are degraded due to increased usage, if the hardware resources of the server (Memory and CPU) *have enough head room*, then you can implement *vertical clustering* by adding two, or more, server nodes of the application on the same physical server.

The following figure shows vertical clustering.



### Horizontal clustering

In cases where you observe that the application transaction response times are degraded due to increased usage, if the hardware resources of the server (Memory and CPU) *do not have enough head room*, you can implement *horizontal clustering* by adding another server and installing a Unifier instance on the added server.

The horizontal clustering is shown in the **Horizontal Scaling** (on page 10) section of this guide.

For high availability scenarios, Oracle recommends horizontal clustering in production systems. A mix of horizontal and vertical clustering is recommended for large deployments.

---

**Note:** While creating application clusters, the administrator must monitor the database server resource utilization. If the database performance worsens, then the administrator must tune the database or upgrade the hardware.

---

## Database Scaling and Clustering

Database server scaling options are available and have been widely adopted and implemented. Database clustering (Oracle Real Application Clusters) enables multiple server nodes in a clustered system to mount and open a single database that resides on shared disk storage. This configuration provides high availability in the database environment.

## Database File Storage

Unifier supports storing historical data in the database table. Over a period of time, data can grow in size and the database storage size will be varied, depending on deployment and usage.

Unifier provides an option for storing documents with various document management technologies.

When sizing storage, determine storage needs of your deployment.

Oracle recommends high performance I/O disks to optimize performance throughput. When creating database instance, ensure that you have enough redo log files that are sized according to your planned database activity.

## Operating System Upgrade

The recommended OS upgrades can improve the performance and scalability of Unifier. Ensure that you are:

- ▶ Upgrading to the latest versions of the operating system
- ▶ Installing the latest patch updates
- ▶ Upgrading to a 64-bit version

---

**Note:** For the full list of system requirements, applications, and application version levels, refer to the *Unifier Tested Configurations* document.

---

## Deployment Considerations

---

Unifier performance depends on the factors identified in the following sections of this guide.

Also, consider the following factors during deployment planning:

- ▶ Follow the instructions in the *Unifier Installation Guide*.
- ▶ Configure the recommended JVM Heap for application servers.
- ▶ Proper configuration to downstream applications and servers such as email and Oracle BI publisher.
- ▶ Establish proper access for the supporting servers such as email account and BI Publisher (BIP) report account.
- ▶ Define adequate table spaces for the database.
- ▶ Provide plenty of disk space for file repository.

## Deployment Categories

Unifier deployments can be classified into three categories:

- ▶ Small
- ▶ Medium
- ▶ Large

The following table shows the criteria used to classify Unifier deployments.

Deployment Category	Small	Medium	Large
Estimated Number of Named Users	<250	251 - 750	751 - 1500
Maximum Concurrent Users (Assumed 20% of named users, with an average of 20 seconds think time)	50	150	300
Estimated Number of Projects/Shells	<1,000	1,000 - 5,000	5,001 - 10,000

### Important information about Deployment Categories

- ▶ A named user is a user who has an account with the application but may not be logged in.
- ▶ A concurrent user is a user who logged in to the application with active interactions.
- ▶ This document considered 20 second intervals between interactions (the think time).
- ▶ The hardware sizing is based on maximum concurrent users estimated for each deployment categories. The maximum concurrency is assumed to be 20% of the named users.

The following section outlines server and storage requirements for the above defined deployment categories.

## Configuration for Deployment Categories

To achieve the appropriate application scalability, performance, and availability, Oracle recommends that you scale your hardware configuration by *at least matching the values* indicated in tables for the following configurations:

### Application Server Configuration

Deployment Category	Small	Medium	Large
JVM Heap	6 GB	12 GB	24 GB
CPU	1 Core	2 Core	4 Core

<b>Deployment Category</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
64 bit, 1 Core @ 2.90 GHz, Intel® Xeon® CPU E5-2690 or equivalent			
Server/VM Disk Space	40 GB	80 GB	120 GB
Managed Server Instances	1	2	4

### Database Server Configuration

<b>Deployment Category</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
CPU 64 bit, 1 Core @ 2.90 GHz, Intel® Xeon® CPU E5-2690 or equivalent	1 Core	2 Core	4 Core
Memory	15 GB	30 GB	60 GB
Database Storage Size	500 GB	1000 GB	1500 GB
Storage for Files	300 GB	900 GB	1800 GB

### OHS Configuration

<b>Deployment Category</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
CPU 64 bit, 1 Core @ 2.90 GHz, Intel® Xeon® CPU E5-2690 or equivalent	1 Core	1 Core	1 Core
Memory	4 GB	4 GB	4 GB
Storage	20 GB	20 GB	20 GB

**Unifier Configurator Guide**

<b>Configuration Parameter (Foreground)</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
Front-end Server Count	1	2	4
Database Connections	60	100	120
High Priority Job Thread Count	5	10	10
Normal Priority Job Thread Count	Disabled	Disabled	Disabled
<b>Note:</b> Servers scaled based on load.			

<b>Configuration Parameter (Background/Integration)</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
Background/Integration Server Count	1	2	4
Database Connections	60	100	120
High Priority Job Thread Count	5	10	10
Normal Priority Job Thread Count	10	20	40
<b>Note:</b> Servers scaled based on load.			

<b>Suggested Size Classification</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
Job Count	< 500 / day	500 - 1,000 / day	> 1000 / day
Assumptions: 65% High Priority Jobs ex:			

<b>Suggested Size Classification</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
Auto-creation, Reverse Autopopulate, Autopublish Documents 35% Scheduled Jobs ex: UDR, Cashflow, Activities			
<b>Note:</b> Monitor Background Job delay to observe job load.			

<b>Suggested Integration (SOAP/REST) Size Classification</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
Integration Request Count	< 5,000 / day	5,000 - 15,000 / day	> 15,000 / day
Assumption: Load is spread evenly, not considering peak loads.			
<b>Notes:</b> <ul style="list-style-type: none"> <li>▶ Processing intensive calls may require more resources, ex: getUDRData, createBPREcord, updateBPREcord</li> <li>▶ Large Integration Deployments advised to have dedicated Application Servers for Integration.</li> </ul>			



Deployment Category (Front-end Server)	Small	Medium	Large
Estimated Number of Named Users	< 250	251 - 750	> 750
Maximum Concurrent Users (Assumed 20% of named users, with an average of 20 seconds think time)	50	150	300
Estimated Number of Projects/Shells	< 1000	1000 - 5000	> 5000

### Important information about Configuration for Deployment Categories

- ▶ The recommended JVM Heap and CPU requirements are the overall sizing needs to accommodate the Deployment Category. These recommendations can be distributed across managed server instances in case of clustering.
- ▶ The CPU and JVM Heap recommendations are intended for the Unifier application instances only. Operating System and other services or processes demands must be sized separately.
- ▶ The CPU and JVM Heap recommendations are appropriate for the configurations supported as mentioned in the *Unifier Tested Configurations* document.
- ▶ For optimal system performance, Oracle recommends that you deploy Unifier on a 64-bit architecture. A 64-bit architecture includes a 64-bit hardware, 64-bit operation system, 64-bit application servers, and databases deployments using 64-bit Oracle JDK.
- ▶ If you use Oracle database for storing documents, you may need to increase storage space on the database server based on the expected number of documents stored.
- ▶ The recommended OHS configuration will support up to Large deployment category. Oracle recommends that you follow the OHS sizing specifications and tuning guidelines based on your usage patterns.

## Network Bandwidth Estimation

As Unifier users make requests to the Unifier server using various browsers, the browsers store static content in the cache and only dynamic requests will be sent to the server.

You can use the following table to estimate the amount of bandwidth that you may need for a set number of users; however, Oracle recommends that you calculate your application bandwidth requirements in order to better represent the number of people actually using the applications assuming varying levels of intensity.

Concurrent users	<= 50	51 - 100	101 - 150	151 - 200	201 - 250	250 - 300
Recommended Bandwidth (Mbps)	2	4	6	8	10	12

### Important information about Network Bandwidth Estimation

- ▶ Oracle recommends that you enable compression on the WebLogic/OHS server.
- ▶ The network bandwidth recommendations that are described above are based on compression being enabled.
- ▶ The recommended bandwidth estimates take caching into consideration.
- ▶ First page hits to the server are not taken into consideration for bandwidth estimation.
- ▶ The first hit to the server produces a spike in bandwidth because all of static web components will be fetched from the server.
- ▶ After static content is cached, all subsequent requests contact the server for dynamic content.

You can calculate the bandwidth requirements for an application using the following process:

- 1) Calculate the weighted average of the request and compressed response payload-sizes (s) in KB considering the frequency in which your organization views pages and performs actions on them.

- 2) Calculate the bandwidth of one user regarding transmission time (n):

$$\text{Bandwidth (Kbps)} = (8 * s) / n$$

- 3) Calculate the bandwidth for a percentage (c%) of the total number of users (u) that are concurrently logged in and are using the system, assuming that think-time and server-side or client-side processing times are negligible:

$$\text{Bandwidth for named users with zero think / processing time (Kbps)} = (8 * s * u * c\%) / n$$

- 4) Calculate the bandwidth for a percentage (c%) of the total number of users (u) that are concurrently logged in and are using the system, including think-time (t) and server-side / client-side processing times (p):

$$\text{Bandwidth for named users with think and processing time (Kbps)} = (8 * s * u * c\%) / (n + t + p)$$

#### Important information about calculating the bandwidth:

- ▶ If a high number of users can access the application, use a low percentage of the total users to estimate your bandwidth.
  - ▶ If a low number of users can access the application, use a high percentage of the total users to estimate your bandwidth.
  - ▶ Oracle recommends that you at least provision the bandwidth for a single user even if the value received in this step is smaller.
- 5) Repeat this process for each application that you deploy.
  - 6) Calculate your overall bandwidth requirement by taking the sum of the highest bandwidth estimates that you calculated for each application.

---

## Factors that Affect Application Performance

---

The following factors can impact application *server* performance:

- ▶ Number of worker threads configured in the application server
- ▶ Number of configured and available database connections
- ▶ Number of users that will be concurrently uploading data
- ▶ Other applications running on the application server (CPU utilization before Unifier is installed/started)

The following factors can impact *database* performance:

- ▶ Number of database instances on a server (dedicated versus shared)
- ▶ Disk storage system performance (I/O speed, buffer, mirroring)
- ▶ Table space layout and extent sizing
- ▶ Table data, index, and LOB distributions on table spaces
- ▶ Table and index fill factor definition
- ▶ Database block sizing
- ▶ Connection management (dedicated versus MTS)
- ▶ RAM allocations (automatic, SGA, PGA, shared pool, buffer pool)
- ▶ CBO optimizer parameter configuration setting
- ▶ Database table and index statistics gathering mechanism and frequency
- ▶ Additional database jobs

### Configuration, Hardware, and Environment Factors

The following factors can also impact the *application* performance:

- ▶ Hardware architecture and operating system
- ▶ Amount of memory and Swap/Virtual Memory configurations
- ▶ Anti-virus software
- ▶ Amount of I/O being performed by other applications running on the servers
- ▶ Network Interfaces (number of NICs, speed and duplex settings)
- ▶ Network throughput (for example, the time it takes to download a 5 K file between application server and browser)
- ▶ Network hops, latency between browser and application server
- ▶ Network Bandwidth consumed by other applications
- ▶ Amount of memory available on client for browser

### Other Actions that Affect Performance

Some of the other actions that can also impact Unifier application performance include the following:

- ▶ **User actions**

User actions play a key role in the scalability of the application. When sizing a configuration, you need to understand the operations users plan on doing. For example, if you have 200 users in the system all working and loading cost sheets/schedule sheets into the page, then you can expect the application to perform slowly. However, if you have 200 users who only login and look at task logs, custom dashboards the application will perform more quickly. You must consider user roles when determining your scaling options.

▶ **Server hardware**

You need to evaluate your hardware to see if it will work with the application. If the server is old, it will probably not handle as many users as a newer server. In some cases, the server may also be virtualized or segmented. In both cases, this means there are fewer resources for the application. This must be considered when planning for the number of users a configuration can handle.

▶ **Storage types**

All of Unifier tests are conducted with local disks. You can use server-side disk storage or a SAN configuration for your servers; however, a SAN configuration can be more complex to setup with your system. You need to ensure that the connections to the SAN are working.

▶ **Network**

You must ensure your network infrastructure is up-to-date and running efficiently. The application server and the database servers must be in the same location.

▶ **Network locations of end-users**

Performance can also be affected by the network location of the end user relative to the application server. Any user that has many network hops to the application server will likely experience poor performance. More hops and high latency are key factors that you need to consider when planning an installation. An environment that contains many hops and high latency will have the most effect on key areas.

## Conclusion

---

Following a systematic approach to evaluating, planning, and testing the architecture for your Unifier deployment is the only way to assure a successful deployment. With careful examination of the performance objectives, system availability requirements, and short-term versus long-term outlook of system usage, the appropriate hardware choices can be made early in the process.

## Frequently Asked Questions

---

1) How much hardware does a Unifier installation require?

See the **Deployment Categories** (on page 13) section of this guide for the recommended hardware for each deployment size.

2) How much disk space does Unifier require?

The Unifier application requires little space; however, you do need enough space to run the application server software (such as WebLogic) and to keep historic log files.

You must also ensure that you have the appropriate amount of disk space available on your database server. You can learn more about disk space recommendations in the **Deployment Categories** (on page 13) section of this guide.

- 3) Can the Unifier database be installed in a shared database environment?

For large deployments Oracle recommends a dedicated Unifier database server.

- 4) How is bandwidth requirement calculated?

See **Network Bandwidth Estimation** (on page 17).

- 5) Will there be an impact of caching on bandwidth?

Yes. Caching will certainly reduce network round trips thereby boosting the performance of Unifier. The proposed bandwidth estimates take caching into consideration. First page hits to the server are not taken into consideration for bandwidth estimation.

The first hit to the server will be costly as all static web components will be fetched from the server and thereafter subsequent requests will contact server only for dynamic content.

- 6) How much disk space will the database schema require for table spaces?

Tables that include the recommended disk space for different configurations can be found in the **Deployment Categories** (on page 13) section of this guide.

- 7) What is the best way to monitor performance for Unifier?

You can use Oracle Enterprise Manager to monitor many aspects of the database (Oracle database only), in addition to OS and Web Logic exposed metrics.

- 8) What is considered acceptable network latency for Unifier?

Unifier has been tested within simulated latency environments and offers acceptable performance up to 100 ms (round-trip time, from client machine to front end server). Higher latency environments have been tested, but higher network latency results in proportionally slower response times.

- 9) What is the best way to monitor performance for Unifier?

You can use Oracle Enterprise Manager (OEM) to monitor many aspects of the database (Oracle database only), in addition to OS and Web Logic exposed metrics.

- 10) What is considered acceptable network latency for Unifier?

Unifier has been tested within simulated latency environments and offers acceptable performance up to 100 ms (round-trip time, from client machine to front end server). Higher latency environments have been tested, but higher network latency results in proportionally slower response times.