

# Oracle® Collaboration Suite

Sizing and Performance Tuning

Release 2 (9.0.4)

Part No. B15608-01

November 2004

---

This guide helps you plan, size, and tune your Oracle Collaboration Suite deployment. This document includes new information as well as links to information that exists in the Oracle Collaboration Suite documentation library.

This document contains the following sections:

- [Section 1, "Capacity Planning and Sizing"](#)
- [Section 2, "Performance Tuning"](#)

## 1 Capacity Planning and Sizing

This section discusses capacity planning and sizing information for the Oracle Collaboration Suite components. Topics include:

- [Section 1.1, "Suite-Level Sizing Information"](#)
- [Section 1.2, "Oracle Calendar"](#)
- [Section 1.3, "Oracle Email"](#)
- [Section 1.4, "Oracle Files"](#)
- [Section 1.5, "Oracle Ultra Search"](#)
- [Section 1.6, "Oracle Voicemail & Fax"](#)
- [Section 1.7, "Oracle Web Conferencing"](#)
- [Section 1.8, "Oracle Wireless and Voice Access"](#)

### 1.1 Suite-Level Sizing Information

The following guidelines apply to the entire Oracle Collaboration Suite.

#### 1.1.1 Oracle Collaboration Suite Sizing Considerations

When determining the sizing requirements for Oracle Collaboration Suite, you need to consider not only the number of users, but also the usage patterns and the time zones in which your users are located. The distance of users from the Oracle Collaboration Suite servers has a large effect on system performance.

Oracle assumes the following for the peak usage level of Oracle Collaboration Suite:

- 50-75% of the total users in the largest time zone for Oracle Email and Oracle Calendar
- 5-10% of the total users in the largest time zone for Oracle Files

### 1.1.2 Single-Computer Installation Considerations

Oracle recommends the following usage parameters when deploying Oracle Collaboration Suite on a single computer:

- Two CPUs and at least 6 gigabytes of RAM
- Less than 1000 named users
- Oracle Calendar, Oracle Email, and Oracle Files installed and configured
- Light weight Portal and WebUI usage (less than 100 concurrent users)
- Light weight Oracle Web Conferencing usage (approximately thirty concurrent users)
- No built-in high availability options

### 1.1.3 Oracle Collaboration Suite Memory Requirements

For an Oracle Collaboration Suite deployment that supports less than 5000 users, Oracle recommends 4-6 gigabytes of RAM for the middle-tier computers, and 8 gigabytes for the computer hosting the Information Store database.

## 1.2 Oracle Calendar

The *Oracle Calendar Administrator's Guide* contains capacity planning and sizing information for Oracle Calendar. See Appendix A, "Disk Space and Memory" in the *Oracle Calendar Administrator's Guide*.

## 1.3 Oracle Email

This section provides initial planning considerations and hardware requirements for Oracle Email and Oracle Webmail.

This section contains the following topics:

- [Section 1.3.1, "Deployment Choices"](#)
- [Section 1.3.2, "Oracle Email Memory Requirements"](#)
- [Section 1.3.3, "Oracle Webmail Configuration Requirements"](#)

### 1.3.1 Deployment Choices

The following topics provide an overview of the basic deployment decisions that you must make when installing and configuring Oracle Email.

#### Single-Instance Deployment

This is the most basic deployment to install and manage. You can have a deployment consisting of multiple single instance computers where Oracle Internet Directory points user X to instance Y, and user A to instance B. However, there will be redundant copies of all shared messages (such as a mail that goes to the entire company) because instances Y and B will each need to have their own local copy.

## Middle Tier Deployments

The middle tier can scale horizontally to any number of servers. Deciding on the number of middle tiers is a function of balancing the lower cost of managing  $n$  larger boxes against the price-to-performance ratio of a larger number of cheaper, smaller boxes with the same processing power as  $n$  boxes. Two-CPU Linux boxes appear to be the most common combination of price-to-performance ratio and management cost.

### 1.3.2 Oracle Email Memory Requirements

Use the following formula to determine the total computer memory required by Oracle Email on the middle-tier computer:

$$(480 + \text{peak native users} * .25) + \text{peak web users}$$

### 1.3.3 Oracle Webmail Configuration Requirements

The following capacity planning recommendations apply to Oracle Webmail:

- The number of OC4J\_UM Java Virtual Machines should equal the number of CPUs on the server computer.
- The LDAP server should be physically close to the Oracle Webmail server, with low round-trip TCP/IP times.
- Monitor the system for CPU bottlenecks during peak periods.
- Turn off database-side sorting of messages. Database-side sorting is a much slower and more expensive query than fetching messages by `msg_id` order.
- If you are enabling Oracle Web Cache, do not configure it in a clustered environment. Configure a separate instance of Oracle Web Cache on each middle tier computer.
- Turn on HTTP compression either in the Apache Web server as a `gzip` module, or in Oracle Web Cache.

## 1.4 Oracle Files

See the *Oracle Files Planning Guide* for sizing information.

## 1.5 Oracle Ultra Search

See Chapter 2, "Installing and Configuring Ultra Search" in the *Oracle Ultra Search User's Guide* for sizing information.

## 1.6 Oracle Voicemail & Fax

Sizing requirements for Oracle Voicemail & Fax are based on the expected number of concurrent callers recording, retrieving messages, and receiving faxes. [Table 1](#) lists guidelines for configuring your hardware.

**Table 1 Recommended User-to-Port Ratios**

Number of Users	User-to-Port Ratio
Under 100	20:1
100 - 300	30:1
300 - 500	40:1

**Table 1 (Cont.) Recommended User-to-Port Ratios**

<b>Number of Users</b>	<b>User-to-Port Ratio</b>
500 - 1000	50:1
1000 or more	75-100:1

Actual requirements vary based on the roles and responsibilities of end users. For example, a site with call center users who receive large numbers of voice mail messages may require a lower ratio compared to a back-office site which receives little call activity.

The following considerations apply:

- A normal scenario is to size the system based on two calls per voice mail message: one for leaving a message and one for retrieving a message. The number of calls will normally be reduced to between 1.25 and 1.5 per voice mail message when deploying Oracle Voicemail & Fax and Oracle Email together, because some users will retrieve Voicemail through their e-mail client.
- Intel Dialogic/82JCTU telephony cards provide eight ports per card.
- Additional cards can be added to the configurations up to the number of free slots available in any server number. The number of slots available is dependent on server hardware.

### **1.6.1 Oracle Voicemail & Fax Hardware Recommendations**

The following tables list CPU and telephony card hardware configuration recommendations based on the expected number of users per site.

**Table 2 500 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	1	1
Dual-CPU servers	0	0
Intel Dialogic/82JCTU cards	2	2
VFX/PCI cards	1	1

**Table 3 1000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	0	0
Dual-CPU servers	1	1
Intel Dialogic/82JCTU cards	4	4
VFX/PCI cards	1	1

**Table 4 2000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	1	0
Dual-CPU servers	1	1

**Table 4 (Cont.) 2000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Intel Dialogic/82JCTU cards	7	5
VFX/PCI cards	2	1

**Table 5 3000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	0	0
Dual-CPU servers	2	1
Intel Dialogic/82JCTU cards	8	5
VFX/PCI cards	2	1

**Table 6 4000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	1	0
Dual-CPU servers	2	1
Intel Dialogic/82JCTU cards	11	5
VFX/PCI cards	3	1

**Table 7 5000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	0	0
Dual-CPU servers	3	1
Intel Dialogic/82JCTU cards	13	5
VFX/PCI cards	3	1

**Table 8 10000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	1	0
Dual-CPU servers	5	1
Intel Dialogic/82JCTU cards	26	5
VFX/PCI cards	6	1

**Table 9 20000 User Site**

<b>Resource</b>	<b>Primary System</b>	<b>Secondary Systems</b>
Single-CPU servers	1	0
Dual-CPU servers	10	1
Intel Dialogic/82JCTU cards	52	5
VFX/PCI cards	11	1

## 1.7 Oracle Web Conferencing

See the *Oracle Web Conferencing Sizing Guide* for sizing information.

## 1.8 Oracle Wireless and Voice Access

See Chapter 13, "Server Performance Monitoring" in the *Oracle Wireless Administrator's Guide* for sizing information.

# 2 Performance Tuning

This section discusses performance tuning information for the Oracle Collaboration Suite components. Topics include:

- [Section 2.1, "Oracle Calendar"](#)
- [Section 2.2, "Oracle Email"](#)
- [Section 2.3, "Oracle Files"](#)
- [Section 2.4, "Oracle Ultra Search"](#)
- [Section 2.5, "Oracle Web Conferencing"](#)
- [Section 2.6, "Oracle Wireless and Voice Access"](#)

## 2.1 Oracle Calendar

See Appendix B, "Adjusting Calendar Kernel Parameters" in the *Oracle Calendar Administrator's Guide* for performance tuning information for Oracle Calendar.

## 2.2 Oracle Email

This section provides an overview of Oracle Email architecture, details of the major tunable components, and guidelines for scaling the system. It also includes a section describing typical symptoms of performance problems, how to identify the limiting component, and solutions for solving the problems.

See the *Oracle Email Administrator's Guide* for information about monitoring Oracle Email processes.

This section contains the following topics:

- [Section 2.2.1, "Overview of the Oracle Email Architecture"](#)
- [Section 2.2.2, "Understanding the Breakdown of Costs"](#)
- [Section 2.2.3, "Recommended Database Connection Settings"](#)
- [Section 2.2.4, "Recommended LDAP Connection Settings"](#)
- [Section 2.2.5, "Oracle Email Parameter Recommendations"](#)
- [Section 2.2.7, "Monitoring CPU Usage for Oracle Webmail"](#)
- [Section 2.2.8, "Monitoring JavaMail API Response Time for Oracle Webmail"](#)
- [Section 2.2.9, "Determining the Optimum Database Pool Size for Oracle Webmail"](#)
- [Section 2.2.10, "Oracle Webmail Parameter Recommendations"](#)
- [Section 2.2.11, "Scaling Problems and Solutions"](#)

## 2.2.1 Overview of the Oracle Email Architecture

This section provides a brief overview of the Oracle Email architecture and processes. This information is intended to help you understand the tuning recommendations included in this section.

### Oracle Email Processes

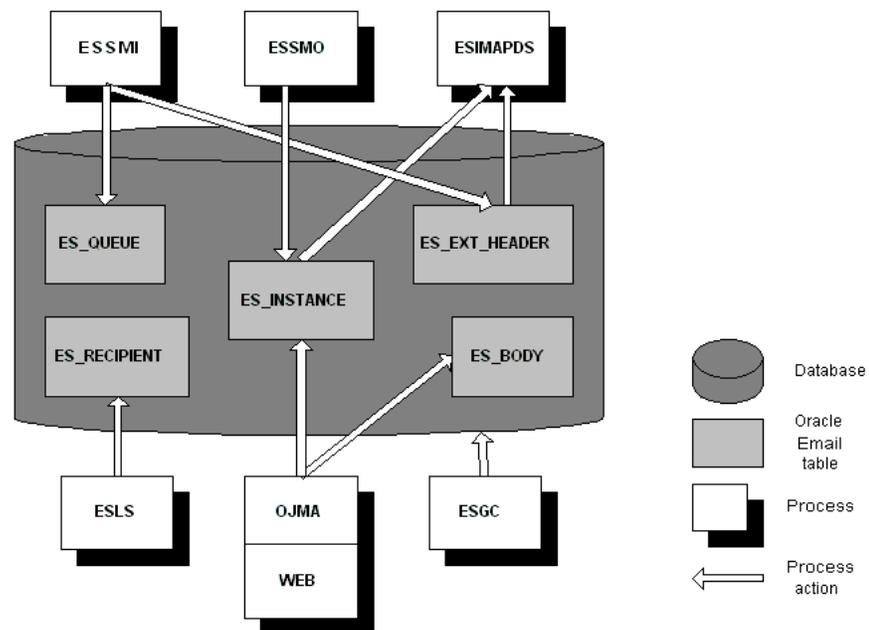
Table 10 describes the Oracle Email processes.

**Table 10 Oracle Email Processes**

Process	Description
ESSMI	The inbound SMTP process
ESSMO	The outbound SMTP process
ESIMAPDS	The IMAP process
ESLS	The list server process
OJMA	An underlying technical base used by Web clients and other applications to access the message store. This is not a server.
ESGC	The housekeeping process that cleans up all transient tables within the Oracle Email schema.

Figure 1 is an overview of the Oracle Email processes and their interactions with the database.

**Figure 1 Oracle Email Processes**



In Figure 1, the ESSMI process is acting in submit-only mode, allowing the outbound SMTP process ESSMO to perform message delivery. The LDAP server, Oracle Internet Directory, is not represented. The consumers of the Oracle Internet Directory server for user authentication purposes are the list server, the SMTP delivery module (ESSMO), and IMAP processes.

## Mail Flow

The following steps describe an overview of the flow for a message entering the system:

1. An incoming message is inserted by ESSMI into the queue tables, `ES_QUEUE` and `ES_RECIPIENT`, and into the tables handling the content and metadata, `ES_HEADER`, `ES_EXT_HEADER`, `ES_SHELL`, and `ES_BODY`.
2. The ESSMO process performs an LDAP lookup to determine whether the message recipient(s) are local or not.
  1. If the recipient(s) are local, ESSMO inserts a record into the pointer table, `ES_INSTANCE`, and updates the recipients' usage in the `ES_USER` and `ES_FOLDER` tables.
  2. If the recipient(s) are not local, ESSMO forwards the mail message.

The following steps describe the flow for a user reading a message:

1. User a Web, POP3, or IMAP client, a user requests a specific message to read.
2. The server collects the message body from the `ES_BODY` table and then finds the metadata (number of attachments, sender, and so on) of the body from the `ES_SHELL`, `ES_HEADER`, and `ES_EXT_HEADER` tables.
3. The server constructs the message in the format the protocol expects and sends it over the network for the user to view.

### 2.2.2 Understanding the Breakdown of Costs

The costs of an Oracle Email system can be broken into three tiers: very costly, moderately costly, and inexpensive. These cost values are relative to other operations on the system. The following descriptions are provided to help you understand how changes in user behavior or application design can impact the hardware tiers.

#### What is Very Costly

**For the database:** Opening a folder. The list of all the user's messages must be retrieved from the database. The cost increases linearly with the size of the folder. This cost is not as high when the table is reorganized as an index-oriented table.

**For the middle tiers:** IMAP fetch new headers operation. Fetching a new list of headers from the client requires the `ESIMAPDS` process to parse each message shell, which is an expensive CPU operation. This operation occurs when a user opens their inbox and downloads all unseen headers. This operation for a first-time user will be for all messages in their folder. For subsequent open folder calls, only new messages are downloaded.

#### What is Moderately Costly

**For the database:**

- Delivering a mail message. New records must be inserted into at least seven tables with the appropriate indexes maintained. Several of the queue tables are updated multiple times as the message moves through various states.
- Reading a mail message. The large object (LOB) in `ES_BODY` must be fetched and the other meta-tables queried to construct the message.
- The housekeeping process. The `ESGC` process marks and then sweeps the entire mail store for de-referenced bodies as well as clean up the queue and

instance tables. This is not inexpensive, but the limiting factor can often be physical input/output constraints, as the objects that the housekeeping process is interested in are unlikely to be in the buffer cache.

- Performing a `get new mail` or `noop:` call. This causes the server to check for new instances for a particular folder identifier.

#### **What is Inexpensive**

##### **For the database:**

- Changing flags of mail messages. This occurs during operations such as marking a mail message read or marking it deleted.
- Reading a mail message already in the buffer cache. The message is already in the buffer, so no additional retrieval is needed.
- Delivering a message to multiple recipients. A large amount of the necessary mail database work has already been performed for delivering the message to a single recipient. Delivery to any additional recipients, especially in the case of a distribution list, is very inexpensive because only new `ES_INSTANCE` records are created.

##### **For the middle tiers:**

- SMTP accepting a mail message. Given the efficiency of the Net8 Listener and Oracle Email SMTP code, little processing power is necessary from the ESSMI component to insert a mail message.
- SMTP performing message delivery. The cost here is the middle tier process, not the database. For example, a single E250 with two 300Mhz processors was able to easily handle delivery of 75,000 messages an hour.

### **2.2.3 Recommended Database Connection Settings**

Database connection limits have a large effect on system performance. Setting the limit too low can lead to unnecessary client failures. Setting it too high can lead to wasted memory on the middle tier and less efficient consumption of the physical memory on the back end database due to the size of the shadow processes.

Use the `oesmon` utility to obtain metric data about the database connections. Use the following command to determine the database connection usage per IMAP instance. For example:

```
oesmon get rgmum4:um_system:imap.DUMP.DBconnections.dump
```

The database connection pool algorithm starts at the top of the pool for each request and moves down the list until an available connection is located. Because of this, the database connection usage is pyramidal in shape: The first connection handles a large percentage of the load, the next connection handles a smaller percentage, and so on. The slope of this shape indicates the rate at which the database is servicing requests: A steep slope indicates a quick response, while a flat slope indicates a slow response. When evaluating this data, look for the connection at which the execution count becomes zero. This indicates that the pool at this point is large enough to accommodate the demands made on it. Resize the maximum database connection pool to this value or greater.

## 2.2.4 Recommended LDAP Connection Settings

The best way to determine the usage statistics for the LDAP connection pool is to set the minimum pool setting to a small number, increment the count by one, and then use the following command to determine how many LDAP connections were created:

```
netstat -a | grep ldap | grep ESTABLISHED | wc -l
```

Note that for some operating systems, you must specify the LDAP port instead of ldap in the command. The default LDAP port is 389.

## 2.2.5 Oracle Email Parameter Recommendations

[Table 11](#) describes the parameters that affect the performance of Oracle Email. These parameters can be accessed and edited through Oracle Enterprise Manager. They are located on the server pages listed under Unified Messaging.

**Table 11** *Parameter Recommendations for Oracle Email*

Parameter	Description	Recommended Value
LDAP Maximum Connection Pool	Determines the maximum number of LDAP connections.	20
LDAP Minimum Connection Pool	Determines the minimum number of LDAP connections. Specify the number of connections required to handle 60% of the peak load.	5
LDAP Connection Pool Increment	The size increment by which to increase the pool of LDAP connections. If a large number of requests arrive at the same time, the pool may not be able to grow quickly enough and some connections will be refused.	1
LDAP Connection Retry Interval	The time to sleep (in microseconds) between attempts to make an LDAP connection. If the value is too large, response time can suffer because the process will sleep for too long a time interval even if an LDAP connection is available.	100000
LDAP Number of Retry Before Erroring	Number of time to try making an LDAP connection before ending the query. If the pool is full, the process will sleep for the value specified for the LDAP Connection Retry Interval parameter.	10
Get New Mail Interval	The time to sleep (in seconds) before entering administrative statistics into the database. If the value is too low, the frequent writes to the database will slow system performance.	600 (7200 for housekeeping processes)

**Table 11 (Cont.) Parameter Recommendations for Oracle Email**

Parameter	Description	Recommended Value
Protocol Server Maximum Threads	The maximum number of threads available for client connection handling.	50
Protocol Server Minimum Threads	The minimum number of threads available for client connection handling.	10
Protocol Server Increment Thread	The number of threads to add to the client connection pool.	1
Maximum Number of Clients	The maximum number of concurrent connections.  If the value is too large, IMAP will attempt to handle connections that the listener is rejecting.  If the value is too small, an unnecessary number of IMAP processes will need to be configured.	1000
Timeout Interval	The length of time (in seconds) until the server disconnects idle sockets	1800
Submit Only	The delivery behavior of the SMTP inbound server.  Setting this value to FALSE can cause long insert times on the inbound Oracle Email server.	TRUE

## 2.2.6 Recommended Database Process Parameter Settings

Table 12 describes the database process parameters that you can change in order to affect the performance of Oracle Email.

**Table 12 Database Process Parameter Recommendations for Oracle Email**

Parameter	Minimum Value	Maximum Value	Increment
ESIMAPDS	10	50	2
ESSMI	10	50	2
ESSMO	10	50	2

## 2.2.7 Monitoring CPU Usage for Oracle Webmail

Use the `sar` or `vmstat` operating system utilities to determine CPU load by averaging the values over ten minute periods. No more than 90% of the CPU resources should be used. Depending on the platform, a Java Virtual Machine might not be able to schedule threads on other processors. Because of this, when running one Java Virtual Machine on a two CPU computer you might only see 50% CPU utilization. However, that Java Virtual Machine might be using 100% of one of the CPUs.

## 2.2.8 Monitoring JavaMail API Response Time for Oracle Webmail

Set the parameter `oracle.mail.sdk.esmail.timing` to `true` to display timing data. Use this information to determine whether database or LDAP connection times are slow. If database connection response times are slow, as shown by the timing data and by IMAP client observation, database performance must be tuned.

If LDAP connection times are slow, use the Oracle Internet Directory Statistics Collection tool (`oidstats.sh`) to generate data required by the Oracle Optimizer to choose an optimal plan to execute the queries corresponding to the LDAP operations. If LDAP connection times are still slow, monitor the CPU usage of the machine and check the network round-trip times.

## 2.2.9 Determining the Optimum Database Pool Size for Oracle Webmail

Set the Oracle Webmail parameter `oracle.mail.sdk.esmail.dbpool_timing` to `true` in order to generate a line marked `Active` in the Oracle Webmail log files. If you set the parameter to `true` for a single day and then search for the `Active` lines in the log files, you can determine the number of daily active database connections. By correlating this information with the timestamp, you can determine the peak number of database connections used. Use this information to determine whether the database connection pool is set correctly.

## 2.2.10 Oracle Webmail Parameter Recommendations

Table 13 describes the Oracle Webmail process parameters that you can change in order to affect the performance of your system. These values are located in the file `oc4j.properties` for the `OC4J_UM` container.

**Table 13** *Parameter Recommendations for Oracle Webmail*

Parameter	Description	Recommended Value
<code>client.mail.defaultsort</code>	If <code>TRUE</code> , the Oracle Webmail client automatically sorts by the default sort field and order, when user first logs in.	<code>FALSE</code>
<code>client.esdsconnpoolparam.incrementsize</code>	Number of connections to add to the ESDS client connection pool	5
<code>client.esdsconnpoolparam.initialsize</code>	Initial number of connections in the ESDS client connection pool	30
<code>client.esdsconnpoolparam.maxsize</code>	Maximum number of connections in the ESDS client connection pool	60
<code>client.esdsconnpoolparam.minsize</code>	Minimum number of connections in the ESDS client connection pool	30
<code>client.esdsconnpoolparam.shrinkingtimeoutinterval</code>	Time delay before ESDS client connection pool can be shrunk	1800
<code>client.esdsconnpoolparam.timeoutinterval</code>	Maximum number of seconds the ESDS client waits for a free connection in the pool. If no connections are released back to the pool within that time, the directory server code throws an exception.	30

**Table 13 (Cont.) Parameter Recommendations for Oracle Webmail**

<b>Parameter</b>	<b>Description</b>	<b>Recommended Value</b>
<code>jdbc.connection.debug</code>	Enables or disables debugging JDBC connections	FALSE
<code>mail.debug</code>	Enables or disables debugging OJMA API for Oracle Email. If enabled, this adversely effects performance.	FALSE
<code>oracle.mail.ldap.reconnecttime</code>	The amount of time in seconds the server waits to reconnect to the LDAP server if it is unavailable.	4
<code>oracle.mail.sdk.esmail.timing</code>	Prints timing data. This is used to determine slow LDAP and database access times.	FALSE Set to TRUE only when debugging performance issues. See <a href="#">Section 2.2.8, "Monitoring JavaMail API Response Time for Oracle Webmail"</a> on page 12 for more information.
<code>oracle.mail.sdk.esmail.ldap_debug</code>	Enables or disables debugging OJMA API for LDAP.	FALSE
<code>oracle.mail.sdk.esmail.dbpool_timing</code>	Prints the active and total count of database connections. This is used to determine if the database pool is sized correctly.	FALSE Set to TRUE when sizing the database connection pool. See <a href="#">"Determining the Optimum Database Pool Size for Oracle Webmail"</a> on page 12 for more information.
<code>oracle.mail.sdk.esmail.cache_inactivity_timeout</code>	Number of seconds to wait for a connection before the ESMS client connection pool times out	600
<code>oracle.mail.sdk.esmail.connpool_max_limit</code>	Maximum number of connections in the Oracle mail sdk esmail connection pool	20
<code>oracle.mail.sdk.esmail.connpool_min_limit</code>	Determines the initial or minimum number of connections created in the connection pool.  Oracle recommends keeping this limit as low as possible to avoid holding on to unused database connections.	5

**Table 13 (Cont.) Parameter Recommendations for Oracle Webmail**

Parameter	Description	Recommended Value
<code>oracle.mail.sdk.esmail.cache_scheme</code>	Determines the cache scheme of the database connection pool.  A value of 3 sets the parameter to <code>FIXED-NO-WAIT</code> , which returns null if a connection is not available in the pool.	3

### 2.2.11 Scaling Problems and Solutions

This section describes typical scaling problems that could occur, their symptoms, and solutions. For the database-related components, this document assumes database administrator skills and a familiarity with Statspack. The resolution component of this document assumes that the affected middle tiers have the recommended CPU and memory resources.

#### Messages in the Delivery Queues Stay High

If the `ES_QUEUE` table has a large number of records in `state=1` (unprocessed), the problem could be one of the following:

- An insufficient number of ESSMO processes doing delivery
- A problem with name resolution (LDAP)
- Contention between processes performing insertions into the `ES_INSTANCE` table

To resolve this problem:

- Check the ESSMO log files for errors (such as storage capacity problems in the database)
- If there are no errors recorded in the ESSMO log, add two to four new ESSMO processes and observe the change in the queue processing.
- If the number of unprocessed `ES_QUEUE` records is consistently very high (more than 1,000 messages awaiting delivery), check the CPU usage on the LDAP server being used for name resolution. Also check that the LDAP server has been configured to accept as many socket connections as ESSMO can potentially create. If the LDAP server is not the bottleneck, run a Statspack snap on the database during the peak period. Check for contention on inserting into the tables `ES_INSTANCE`, `ES_QUEUE`, `ES_RECIPIENT`, `ES_EXT_HEADER`, `ES_HEADER`, `ES_SHELL`, and `ES_BODY`.

#### The Housekeeping Process Cannot Clean Up All of the Old Mail

For performance reasons, it is best to run the housekeeping process only at night. If the number of messages in `msg_state=3` waiting to be cleaned up from `ES_QUEUE` or in either `folder_id=3` ("pending prunable") or `folder_id=4` ("pending collectable") in `ES_INSTANCE` continues to grow, then the housekeeper should be run during the day as well. If this does not allow the housekeeper to keep up, then find out whether the system is falling behind in pruning mail messages or collecting them. Create additional housekeeper processes with settings to work only on the task that the system is unable to keep up with.

### **Sending Messages Takes a Long Time**

The problems could be either network congestion or other network problems. If the ESSMI process is configured with `submit_only=false`, then ESSMI performs a name resolution prior to inserting the mail message and the slow response time could be either in LDAP or the database insert.

If ESSMI is configured with `submit_only=false`:

1. Check the log files for errors that may indicate exceeding the LDAP connection pool limit. Also check that the LDAP server is not CPU-bound and is configured to accept the maximum number of connections that ESSMI can request.
2. You can also check the resolution time by checking the time it takes for name resolution to be done by simulating sending a mail message. The time it takes for the server to respond to the `rcpt to:` call is the time needed to perform the LDAP lookup. For example:

```
telnet localhost 25
220 rgmum12.us.oracle.com ESMTP Oracle Email SMTP Inbound Server Ready
helo test
mail from: me@yahoo.com
250 2.1.0 Sender OK
rcpt to: tuser1@oracle.com
250 2.1.5 Recipient ok
```

3. If the LDAP response times are acceptable, then inserting mail into the database is probably the slow component. When a mail message is inserted into the system one or more rows are inserted into the tables `ES_QUEUE`, `ES_RECIPIENT`, `ES_EXT_HEADER`, `ES_HEADER`, `ES_SHELL`, `ES_BODY` and `ES_INSTANCE`. Contention on any of these tables can cause slow inserts. Run a Statspack level 7 report on the database to check for enqueue contention on these tables and associated indexes. If there is contention, partition the tables. If the contention is on the `msg_id` index on the `ES_INSTANCE` table, then change the index to be a reverse key index. Of course, if there are other specific database issues not related to Oracle Email (for example, a large number of sessions waiting for disk writes or index reads from disk to complete) then you must first troubleshoot those issues.

If the ESSMI process is configured with `submit_only=true`, ESSMI does not perform the LDAP lookups prior to inserting into the database. Because of this, if sending a mail message is slow, then the problem is the network or the database. Follow the steps in the previous section to determine whether the database is a bottleneck, and if so, partition the tables as required.

### **It Takes a Long Time to Authenticate IMAP or POP3**

No database work is done during authentication. Only LDAP queries are performed, so this is a symptom that either the network or LDAP is slow. Check the LDAP server and database for resource contention. Also check whether `OIDStats.sh` has been run since a major data change.

### **It Takes a Long Time to Open the Inbox**

One query for opening the inbox goes through the `ES_INSTANCE` table to gather all of the rows for a particular folder identifier. As the size of the inbox folder increases, the number of blocks that must be fetched also increases. The database query is the following:

```
SELECT msg_id, msg_uid, msg_size, flags, unread, to_char(internal_date,
'DD-Mon-YYYY HH24:MI:SS'), msg_type FROM es_instance WHERE folder_id = :b1
ORDER BY msg_uid ASC
```

If the number of buffer gets per execution exceeds 200, consider using an index-organized table, which will cluster the same folder identifiers into a set of similar blocks. Performance can also be improved temporarily by rebuilding the table physically sorted by `folder_id`, although this optimization will degrade over time. If the query is extremely expensive (greater than 3000 buffer gets per execution) and the folder inbox sizes are small (fewer than 2000 messages per inbox), then there is most likely a bad execution plan and the `ES_INSTANCE` table should be analyzed.

### **It Takes a Long Time to Read an E-Mail Message**

The query for reading a mail message fetches the BLOB body from the `ES_BODY` table, the shell from the `ES_SHELL` table, and the header data, and then updates the folder with a modification date. If any of the selects are slow, then the end read will be slow. Check Statspack for Wait events on these objects to display the problem. Slow reads rarely occur unless there is a resource limitation such as disk input/output throughput.

### **At Peak Periods Valid Users Receive an "Invalid Username or Password" Error**

This is an erroneous error message; the real problem is that the LDAP pool has been exceeded. If there are available CPU resources on the LDAP server, increase the LDAP pool parameters for the IMAP processes.

### **The Oracle Listener Process, TNSLSNR, Exits with No Core File and All Later Requests to the Socket Fail**

This occurs when the number of file descriptors held by the listener exceeds 1024. After the listener accepts a socket connection, it hands off the file descriptor to an Oracle Email server process. The listener holds onto the file descriptor until it is handed off to the Oracle Email process. If the Oracle Email process server cannot accept the handoff quickly, the number of file descriptors held by the listener will grow. This typically only happens if the middle tier does not have sufficient CPU resources.

## **2.3 Oracle Files**

The *Oracle Files Administrator's Guide* contains performance tuning information for Oracle Files. See the following chapters:

- Chapter 7, "Monitoring Domain, Node, and Service Performance"
- Chapter 9, "Maintenance and Tuning"
- Chapter 11, "Troubleshooting"

## **2.4 Oracle Ultra Search**

See Appendix A, "Tuning the Web Crawling Process" and Appendix B, "Tuning Query Performance" in the *Oracle Ultra Search User's Guide* for performance tuning information.

## 2.5 Oracle Web Conferencing

See the *Oracle Web Conferencing Frequently-Asked Questions and Troubleshooting* document for information about minimum system requirements. This document is located at Oracle Technology Network:

[http://www.oracle.com/technology/products/ortc/pdfs/webconferencing\\_troubleshooting.pdf](http://www.oracle.com/technology/products/ortc/pdfs/webconferencing_troubleshooting.pdf)

By default, one Web Conferencing Listener and four Web Conferencing Server processes are started when Oracle Web Conferencing is installed and configured on a middle tier computer. If your computer exceeds these requirements, you can improve performance by increasing the number of processes.

The number of listener processes should equal the number of CPUs in the middle tier computer.

To determine the recommended number of server processes, use the following formula:

$$\text{Number of server processes} = (\text{amount of physical memory}) / 256$$

## 2.6 Oracle Wireless and Voice Access

See Chapter 13, "Server Performance Monitoring" in the *Oracle Wireless Administrator's Guide* for performance monitoring information.

## 3 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

