Compaq ActiveAnswers

Solution Guide

December 2000
13XV-1200A-WWEN
Prepared by:
Enterprise and Mid-Market
Solutions Engineering
Compaq Computer Corporation

Contents

Introduction 3
TPC Benchmark C (TPC-C) 3
Oracle8*i* Server Overview 4
Key Features of Oracle8*i* Server 10
Tuning Goals 11
Basic Tuning 13
I/O Tuning 24
Processor Scaling 33
Memory Tuning 34
Updating Init*SID*.ora 39
Network Tuning 41
System Management 41
Conclusion 41

OLTP Configuration and Tuning of Oracle8*i* for Windows 2000 on Compaq Servers

Abstract: This Technical Guide provides guidelines for configuring and tuning the performance of Oracle8*i* Server and Microsoft Windows 2000 Server and Windows 2000 Advanced Server on Compaq servers in an Online Transaction Processing (OLTP) environment.

The system tested by Compaq represents a single node, Transaction Processing Performance Council (TPC) Benchmark C on the *Compaq ProLiant* family of servers.

ESE_AA_SOLUTIONS@COMPAQ.COM

Notice

OLTP Configuration and Tuning of Oracle8*i* for Windows 2000 on Compaq Servers
Solution Guide prepared by Enterprise and Mid-Market Solutions Engineering

First Edition (December 2000)
Document Number 13XV-1200A-WWEN

Introduction

This Technical Guide contains information to help you optimize the performance of your On-line Transaction Processing (OLTP) system. This guide lists tuning goals, and provides recommendations for basic tuning and for tuning your I/O performance, processors, and memory. An introduction to the Transaction Processing Performance Council (TPC) Benchmark C and an overview of the key features and architecture of Oracle8*i* server are provided.

When tuning an OLTP system, it is important to realize that optimizing parameters for one application does not necessarily improve performance for other applications. Tuning is an iterative process; requirements can change as your workload changes.

TPC Benchmark C (TPC-C)

The TPC-C benchmark is a workload that consists of read-only and update-intensive transactions simulating the activities found in a complex OLTP application environment. TPC-C exercises the simultaneous execution of multiple transaction types spanning a broad spectrum of complexity. These transaction types include:

- Online and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity
- Non-uniform distribution of data access through primary and secondary keys
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention on data access and update

While TPC-C offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. Compaq does not recommend extrapolation to any other environment.

Note: For more information on TPC-C, contact the Transaction Processing Performance Council on http://www.tpc.org/cspec.html or phone 408-295-8894.

Oracle8*i* Server Overview

The information in this guide relates to Oracle8*i* Server Enterprise Edition 8.1.6 for Microsoft Windows 2000. Because information on the generic tuning of Oracle8*i* Server is already available, this document focuses on specific tuning suggestions for OLTP on Compaq servers and Windows 2000 Server and Windows 2000 Advanced Server. All references to Windows 2000 are applicable to both Windows 2000 Server and Windows 2000 Advanced Server. Wherever possible, this document references additional tuning documentation.

This section provides general information on the architecture of Oracle8*i* Server.

Oracle8*i* Server for Windows 2000 is a 32-bit application that is implemented on Windows 2000 as a single process, multi-threaded architecture. Each Oracle8*i* Server instance consists of:

- A single Windows 2000 process with multiple Oracle8*i* server threads
- A System Global Area (SGA)
- A Process Global Area (PGA)
- A redo log
- Control and configuration files

This section provides basic information on the Oracle8*i* Server architecture. A more in-depth discussion is available in the *Oracle8i Server Concepts Manual.*

Threads

Note: Under the multithreaded architecture of Windows 2000, an Oracle8*i* Server instance is considered to be a Windows 2000 process while Oracle8*i* Server processes[1] are considered to be Windows 2000 threads. Therefore, for the purpose of this Technical Guide, Compaq refers to an Oracle8*i* Server instance as a process and an Oracle8*i* Server process as a thread.

An Oracle process consists of **server threads** (handling requests from user connections), **background threads** (consolidating and handling functions that would otherwise be carried out by multiple management tasks running for each user connection), and **shadow threads**[2] (reflecting the number of user connections to the database).

The number and type of threads associated with an Oracle process depends on the number of Oracle8*i* Server options selected (background threads) and the number of user connections (shadow threads).

Server Threads

Each Oracle process spawns two server threads, regardless of the options selected and the number of user connections. Thread 0, the Oracle process main thread, acts as a service dispatcher, creating thread 1 to handle the service.

[1] Server, background, and shadow processes.
[2] There is one shadow thread for each user connection when not using a Multi-Threaded Server.

Background Threads

Each background thread is associated with an Oracle8*i* Server background process: **PMON, DBWn, LGWR, CKPT, SMON,** or **RECO**. In addition, if you enable **ARCHIVELOG** mode, Oracle8*i* Server creates the **ARCn** background thread. For more information on background threads, see Table 1.

Table 1. Oracle8*i* Server Background Threads

Thread	Mnemonic	Description
Process Monitor	PMON	Process Monitor is responsible for the cleanup of abnormally terminated connections.
Database Writer	DBWn	Database Writer writes modified database blocks to data files.
Log Writer	LGWR	Log Writer writes the redo log entries to logfile.
Checkpoint Process	CKPT	Checkpoint Process signals **DBWn** to perform updates on all database data and control files. If **CKPT** is not present, **LGWR** signals **DBWn** to perform the updates.
System Monitor	SMON	System Monitor performs instance recovery and cleanup.
Recovery Process	RECO	Recovery Process resolves failures associated with the distributed database option.
Archival Process	ARCn	If you enabled **ARCHIVELOG** mode, Archival Process copies a full redo log file to the archive device.

Note: Since Compaq disk subsystems provide asynchronous input/output, Oracle8*i* Server requires only one Database Writer (**DBWn**) thread and one Log Writer (**LGWR**) thread.

Shadow Threads

A shadow thread is a separate, dedicated server thread that acts on behalf of a particular user. Oracle8*i* Server creates a shadow thread for every user who connects to the database, and uses this thread to perform database requests made by the user.

Note: Oracle8*i* Server supports a Multi-Threaded Server (MTS) environment. An MTS server thread can service requests from multiple clients.

When performing a database request, the shadow thread first checks to see if the required information is present in the System Global Area (SGA); if the information is not present, the shadow thread reads the appropriate data from data files into the SGA. When writing information to the database, the shadow thread writes into the SGA only, **DBWn** later writes this “dirty” data to disk.

Shadow threads can operate in parallel. Parallel execution begins when a shadow thread executes a SQL statement containing operations that can be performed in parallel. This shadow thread becomes the **query coordinator**, which dispatches the execution of the SQL statement to several parallel server threads, coordinates the results, and sends the results to the user. The number of parallel server threads assigned to a single operation is known as the “degree of parallelism.”

The maximum number of shadow threads (user connections) depends on the size and configuration of system memory available to Oracle8*i* Server.

Monitoring Process and Thread Activity

This section discusses using Windows 2000 Advanced Server to monitor process and thread activity.

Process Status

Microsoft Windows 2000 represents the Oracle8*i* Server process as a service named OracleServiceSSSS (where SSSS is the system ID) associated with the executable "ORACLE.EXE". Each Oracle8*i* Server instance, by which an Oracle8*i* database is created and accessed, is associated with a specific Windows 2000 service.

In Windows 2000, select Start →Programs→Administrative Tools→Computer Management→Services and Applications→Services to display the status of services as shown in Figure 1 where "OracleServiceORCL" represents Oracle8*i* Server instance ORCL.

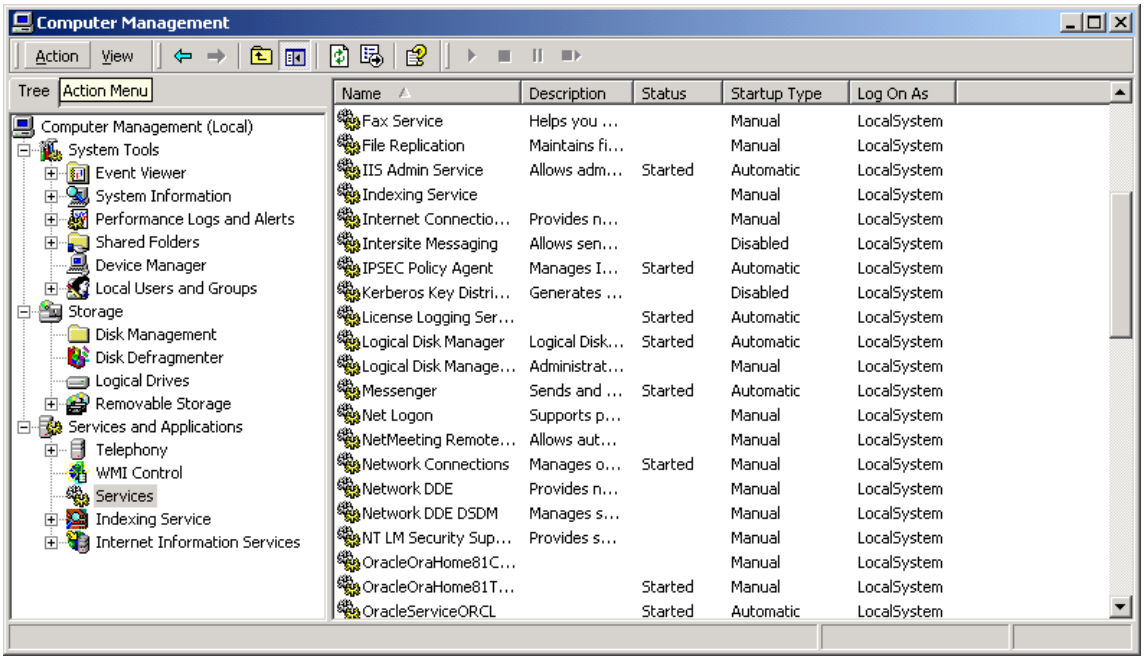

Figure 1. The Services screen showing an Oracle8*i* Server instance

Thread Activity

The Windows 2000 Performance Monitor allows you to monitor Oracle8*i* Server thread activity.

Start Performance Monitor and select the button identified by the plus sign ('+') to obtain the Performance Monitor screen illustrated in Figure 2. Select "Thread" from the Performance object list and use this screen to associate Performance Monitor counters with particular thread instances. For example, selecting "ORACLE ==> 3" allows you to add counters for Oracle ID Thread 3.

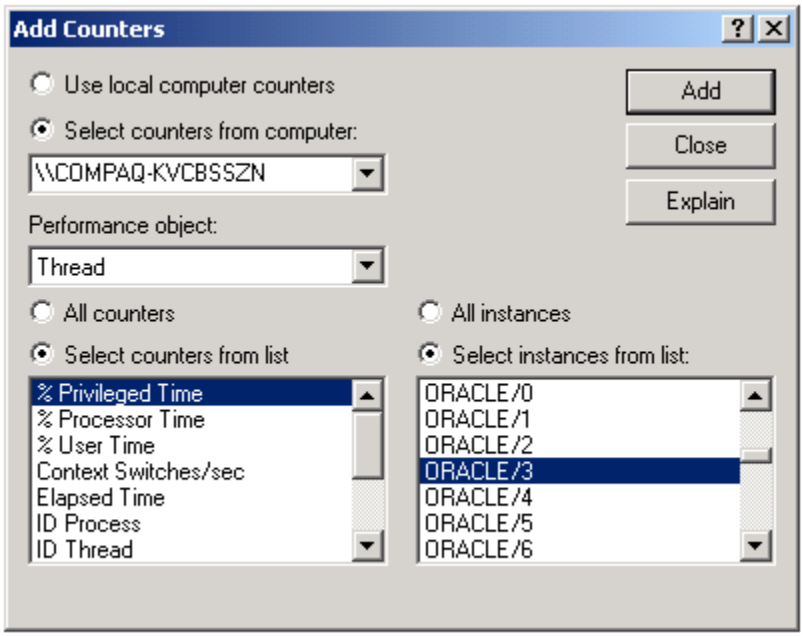

Figure 2.The Performance Monitor Thread Object Screen

You can identify a particular Oracle process from the value of the ID Thread Counter. From the screen illustrated in Figure 3, follow these steps:

1. Select “ID Thread” from the Counter list.
2. Select all Oracle instances from Instance list.

The resulting screen provides information on all Oracle threads.

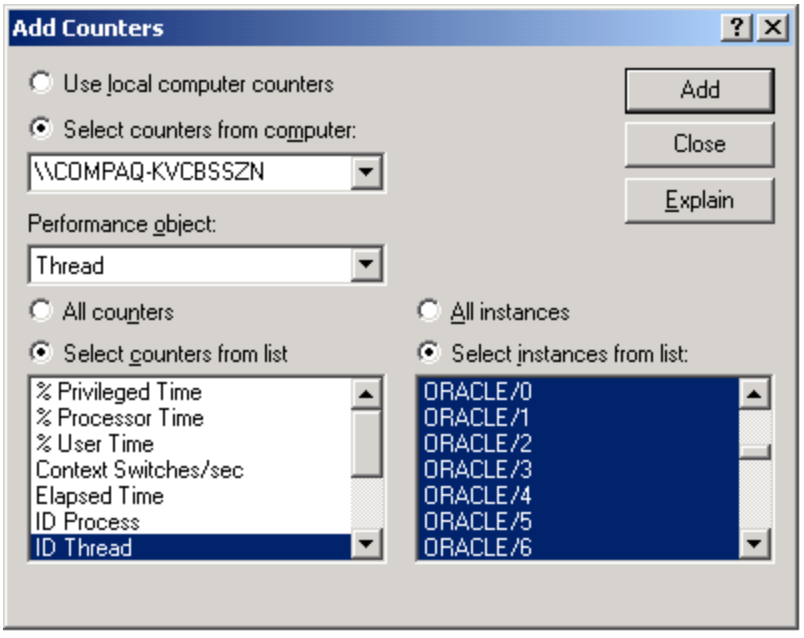

Figure 3. Identifying Oracle Processes

In Figure 4, the user has selected ID Thread 3. The value of the selected ID Thread Counter is 1256 (as displayed in the "Last," "Average," "Min," and "Max" boxes). To determine which background process corresponds to a particular ID Thread Counter value, you can use this SQL statement:

```
SELECT bg.paddr, bg.name, bg.description, p.spid, p.pid
FROM v_$bgprocess bg, v_$process p
WHERE bg.paddr = p.addr;
```

This statement displays all background processes running on the Oracle8*i* Server instance. See Table 2 for an example of the information displayed.

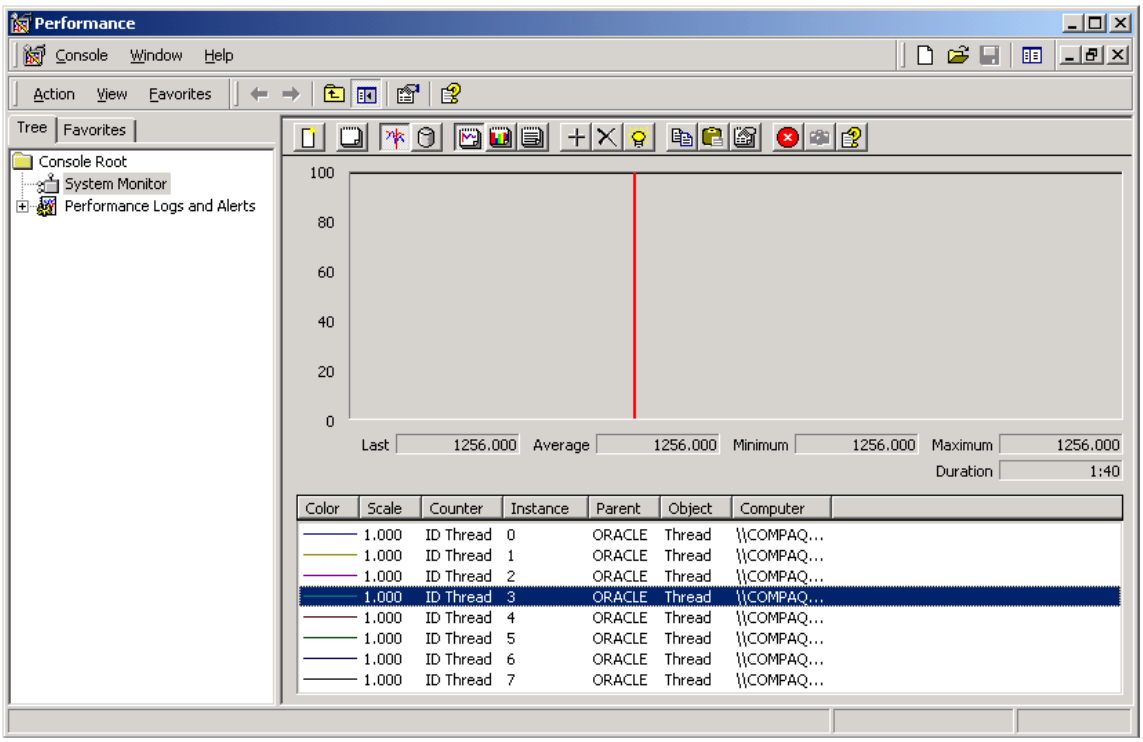

Figure 4. Performance Monitor Showing the Oracle8*i* Process and Threads

Note: Performance Monitor separates Oracle8*i* Server instances on the chart, log, and report but identifies each instance as "ORACLE," making it difficult to monitor multiple instances.

Table 2. Sample Background Process Display

p.addr	bg.name	bg.description	p.spid	p.pid
1C169E2C	**PMON**	process cleanup	1256	2
1C16A11C	**DBW0**	db writer process 0	1309	3
1C16A40C	**LGWR**	Redo etc.	1288	4
1C16A6FC	**CKPT**	checkpoint	1249	5
1C16A9EC	**SMON**	System Monitor Process	1277	6

The **p.spid** values in the display correspond to values specified in the ID Thread Counter; **bg.name** identifies the background process associated with the particular ID Thread Counter value.

From Figure 3, the ID Thread Counter value associated with ID Thread 3 is 1256. From Table 2, a **p.spid** value of 1256 corresponds to the **PMON** process.

Knowing which background process is associated with a particular ID Thread allows you to monitor Oracle8*i* Server threads using any of the thread counters in Windows 2000 Performance Monitor. See Figure 4.

System Global Area (SGA)

Table 3 lists the contents of the SGA sharable memory construct.

Table 3. SGA Contents

Contents	Description
Database Buffers	The Database Buffers contain the most-recently-used database blocks.
Shared Pool	The Shared Pool contains shared SQL areas and the data dictionary cache.
Redo Log Buffer	The Redo Log buffer records all changes made to the database.
Large Pool	The Large Pool is an optional memory area that can provide large memory allocations for Multi-Threaded Server (MTS) session memory, the Oracle8*i* Server XA interface, the I/O server processes, and Oracle8*i* Server backup and restore operations.

Redo Log File

To facilitate the recovery of a particular instance, the Oracle8*i* Server redo log contains a history of the database transactions committed by each instance. Oracle8*i* Server requires at least two redo logs and, if desired, you can use more.

When a redo log fills, a **log switch** occurs. After the log switch, Oracle8*i* Server records all new redo information in the next redo logfile in line. If the system is running in **ARCHIVELOG** mode[3], Oracle8*i* Server copies the previous logfile to an **archive logfile**.

With a recent database backup, the redo logfiles, and the archive logfiles, you can recover an Oracle8*i* Server instance, if needed.

[3] Recommended by Compaq.

Control and Configuration Files

Oracle8*i* Server uses control and configuration files to store information on the state and layout of the database and system tunables. For more information on OLTP configuration, see the Updating Init*SID*.ora section of this guide.

Note: It is very important to backup your control file when backing up the Oracle8*i* database.

Key Features of Oracle8*i* Server

Table 4 presents key features of Oracle8*i* Server 8.1, which can help improve the performance of your OLTP application. For more information on these features, refer to the : *Getting to Know Oracle8i* Manual.

Table 4. Features of Oracle8*i* Server

Feature	Description
Partitioned Tables and Indexes	Oracle8*i* Server can divide large tables and indexes into smaller, more manageable segments that are known as **partitions**. Oracle8*i* Server can create, manage, backup, restore, and utilize partitions to improve query and Data Manipulation Language (DML) performance. Oracle8*i* Server supports range, hash, and composite partitioning.
ROWID Enhancement	Oracle8*i* Server **ROWID**s are 10-byte IDs that contain the data object number, the data block address, and row number of a particular row. **ROWID**s can accommodate partitioned tables and tablespace-relative data block addresses.
Index-Organized Tables	Oracle8*i* Server stores table data in a B*-tree index structure for better performance and reduced storage.
Data Dictionary	The data dictionary includes Oracle8*i* Server views that provide additional information on partitions, parallel server, and latches/locks. Some views have additional columns defined to include new features and functionality.
Reverse-Key Indexes	A reverse-key index reverses the bytes of each column indexed, except the **ROWID**, but maintains the column order. This allows Oracle8*i* Server to distribute insertions across all leaf keys in the index.
SQL*Loader Partitioned Object Support	SQL*Loader partitioned object support facilitates the loading of partitioned objects into the database. The direct path has been changed to accommodate the mapping of rows to table partitions and to support local and global indexes. Parallel direct-path includes concurrent loading of an individual partition as well as support for concurrent loading of a partitioned table.
Multiple Buffer Pools	Oracle8*i* Server supports the configuration of database buffer cache into multiple buffer pools. You can assign schema objects to specific buffer pools. The **KEEP** and **RECYCLE** buffer pools can retain data blocks in memory or eliminate these data blocks from memory when they are no longer needed. The **DEFAULT** buffer pool contains data blocks from schema objects that are not assigned to the **KEEP** or **RECYCLE** buffer pools.
Locally-Managed Tablespaces	The capability to manage tablespaces locally improves reliability and reduces fragmentation by avoiding the recursive space management that can occur in dictionary-managed tablespaces. Locally-managed Tablespaces eliminate the need for system calls to manage dictionary tables. The use of locally-managed tablespaces automatically tracks adjacent free space, eliminating the need to coalesce free extents.

Tuning Goals

There are several factors involved in achieving the best performance from your system, including the optimization of the hardware, the Oracle8*i* Server, the operating system, and the application software. This guide focuses on optimizing these system components:

- Hardware
- Oracle8*i* Server
- Operating system

Note: It is important to tune database application software to take full advantage of system resources. However, due to the diversity of database applications, tuning application software is beyond the scope of this document. For more information on tuning applications, refer to the *Oracle8i Application Developers Guide*.

A well-tuned Oracle8*i* Server system has these characteristics:

- I/O wait time is minimal
- Most of the system processor utilization is allocated to Oracle shadow threads and not the server or background threads
- Most of the system processor utilization is in user mode
- Users see good response times

This section discusses the characteristics of a well-tuned system; subsequent sections provide guidelines for achieving these goals.

Waiting on I/O

Minimal waiting on I/O indicates that the system processors always have work to do while there are outstanding I/Os.

Note: You can use the Windows 2000 Performance Monitor to monitor I/O performance. Performance Monitor statistics appear in this format: (XXXX:YYYY) where XXXX is an object, such as Processor, and YYYY is an associated counter, such as %Processor Time.

If the processor utilization (Processor:%Processor Time) is 100%, then there is no disk I/O contention.

If the %Idle (100 minus (Processor:%Processor Time)) is more than 15% to 20% while the system is under load, you should check the number of outstanding I/Os. You can estimate the number of outstanding I/Os by running Performance Monitor and observing the number of queued transactions. To estimate the disk queue length for each drive, multiply the I/Os per second per drive by the number of seconds per read (PhysicalDisk:Avg. Disk sec/Read). For calculating I/Os per second per drive, see the I/O Limits section. This number should be less than or equal to two for each disk drive.

Note: To observe disk activity using Performance Monitor, run the **diskperf** utility to activate disk performance measurement by the system (**diskperf -YD**). Since **diskperf** degrades system performance by up to 1%, you should disable **diskperf** (**diskperf -N**) after you have completed disk monitoring. This is to restore maximum performance.

You must restart your system for the **diskperf** option to become effective.

Processor Utilization

Most of the system processor resources should be allocated to Oracle. For best performance, Oracle works well when it does not have to share resources with other applications. When monitoring processor utilization, it is important most of the system processor resources are specifically allocated to the shadow threads or user connections, and not the server and background threads. Use Performance Monitor to check the system processor utilization at the Oracle thread level (Thread:% Processor Time). If the performance of your Oracle system is not acceptable, you will have to determine if any applications can be moved to another server, if applications could be rewritten to use less processor time, if you need to add faster processors, or if you need to add more processors. The choice of options varies and is dependent upon the needs of each individual Oracle system.

User Mode

Most of the system processor resources should be available for user applications rather than Windows 2000 kernel applications. Check the system processor utilization by running Performance Monitor and reviewing the percentage of processor time in privileged time[4] (Process:% Privileged Time) and the percentage of processor time in user time for the Oracle process (Process:% User Time). In cases where the level of Oracle performance is unacceptable and the percentage of privileged time is more than the percentage of user time, TKPROF can be used to identify areas within your application that may be causing extra system overhead. Information on using and interpreting TKPROF can be found in the *Oracle8i Tuning Manual*.

[4] Think of privileged time as operating system overhead such as time spent in the I/O subsystem or in making system calls.

Response Times

Users should see good response times. A system that appears to be well-tuned but has poor response times could have an inefficient statement in the application or excess latencies in the network or I/O subsystems. Processor utilization over 80% may indicate a processor bottleneck, and a faster processor may improve response times.

Basic Tuning

This section contains basic tuning information relating to the installation and configuration of the *Compaq ProLiant* server hardware, operating system, and Oracle8*i* Server.

Compaq provides guidelines for:

- Tuning your system at installation time
- Installing Oracle8*i* Server
- Making changes to your Windows 2000 configuration:
  - Installing the latest fixes and drivers
  - Configuring physical, virtual, and server memory
  - Removing unwanted network protocols
  - Configuring server and workstation bindings
  - Stopping unwanted services
  - Configuring the file system

Later sections of this guide discuss the tuning of your OLTP system.

Installation Time

Table 5 provides guidelines for tuning your system at installation time.

Table 5. System Tuning at Installation Time

Action	Description
Updating the flashable ROM	Apply the latest *Compaq ROMPaq* to your *ProLiant* after configuring your system. Applying the *ROMPaq* can take some time if your system has a large number of drives configured.
Configuring the system	Use the *Compaq SmartStart* CD-ROM to configure your system and create your system partition. After creating the system partition, *SmartStart* copies the system configuration files to your system partition after rebooting. This allows you to run **SysConfig** upon bootup (select F10 at the prompt) at any later date. To review and edit your system configuration, reboot and run **SysConfig**. Select System Configuration→Configure Hardware→Review Details→View or Edit Details to review your boot order. Set the order of your boot drive in "Controller Order" to "First". Note that the CD-ROM (Embedded IDE Controller) is always "Enabled as Second Controller".
Configuring the array controllers	Run the Compaq Array Configuration Utility (on the *SmartStart* CD-ROM) against all controllers to be used in your OLTP system. See Figure 5. "Controller Selection" displays the controller and slot number currently viewed on the screen. Apply these settings: **Accelerator Ratio** The accelerator ratio allocates memory reserved for the accelerator's read ahead/write-posting functionality. The amount of memory for the accelerator and the ratio for the accelerator cache depend on the controller board. For more information, see "Array Controllers" later in this Technical Guide. Use Controller Settings→Accelerator Ratio to review and edit the accelerator ratio. **Array Accelerator Cache** Enable the array accelerator cache to increase controller performance by writing data to RAM on the controller rather than directly to disk. The RAM on the controller is ECC-protected and battery-backed for increased reliability. Enable the Array Accelerator Cache for both log drives and non-log drives under Logical Drive→Array Accelerator.

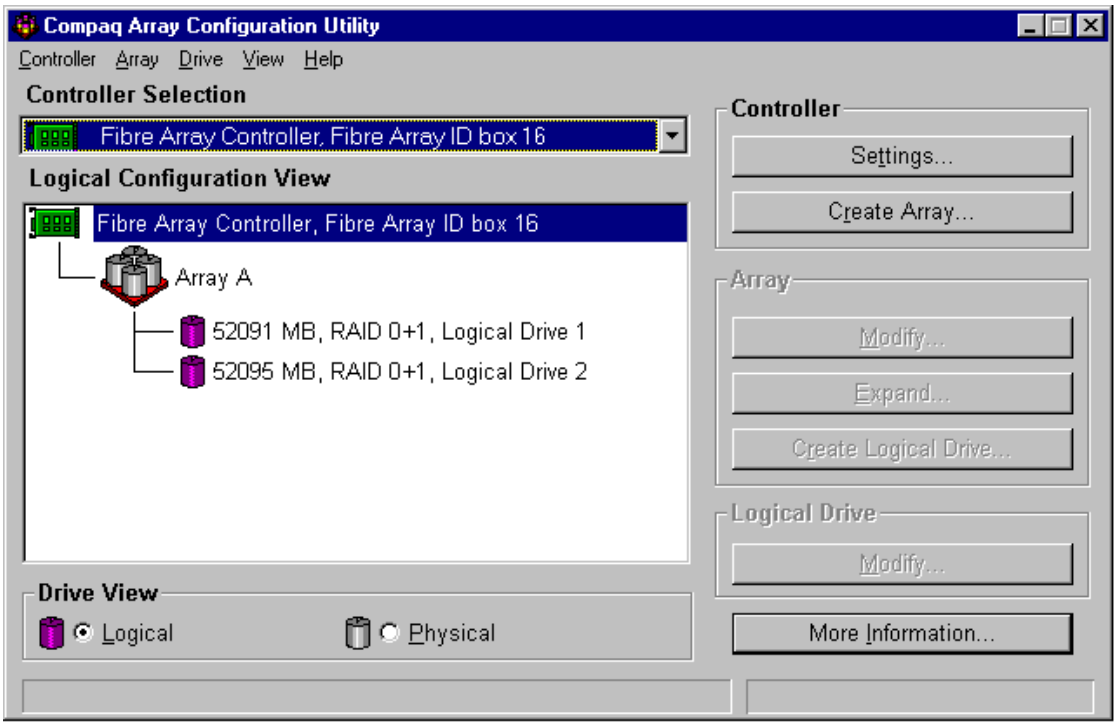

Figure 5. The Compaq Array Configuration Utility Screen

Installing Oracle8*i* Server

Follow these guidelines when installing Oracle8*i* Server:

- To enable Oracle8*i* Server partition functionality, install the partition option.
- Use one of these methods to create an Oracle8*i* Server instance:
  - Use the Oracle Database Configuration Assistant to create the instance automatically.
  - Use the Oracle Database Instance Manager (ORADIM) utility to create the instance manually and assign the administrator ID and password. If you are recreating an instance, you should first delete the old instance using:

    ```
    oradim -delete -sid SSSS
    ```

    before creating the new instance using:

    ```
    oradim -new -sid SSSS  -intpwd PWD -maxusers U
       -startmode manual
    ```

    where **SSSS** is your instance ID, **PWD** is your internal password and **U** is the maximum number of internal users.
- Use the Windows 2000 Registry Editor to make any necessary changes to Oracle8*i* Server registry parameters such as **`ORACLE_SID`**, **`LOCAL`**, and **`ORACLE_HOME`**, as well as other Oracle registry parameters. You should make the changes to: **\SOFTWARE\ORACLE\HOME(OracleID)** where **OracleID** is the ID number found in the **oracle.key** file in the **oracle_home\bin** directory. See Figure 6.

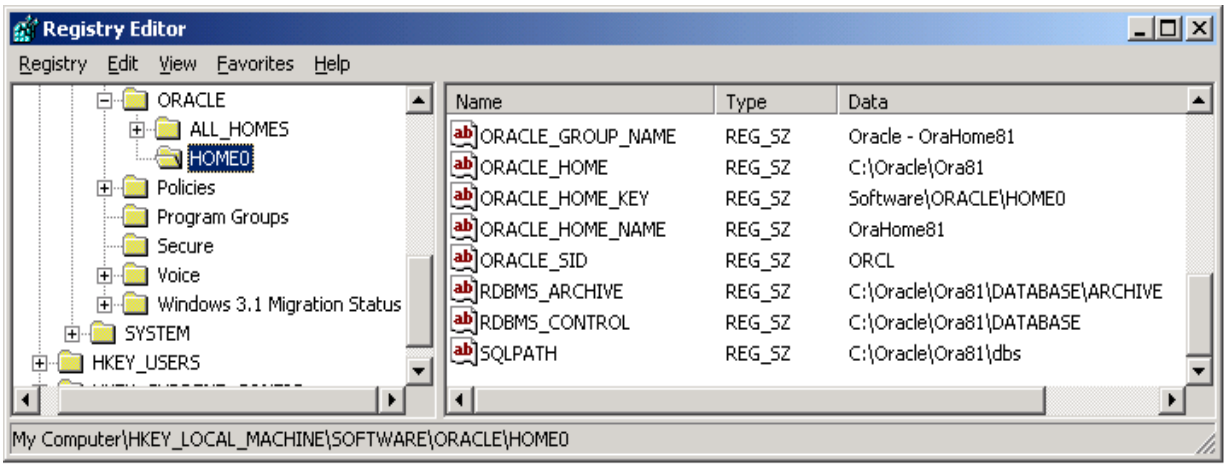

Figure 6. Registry Editor Showing Oracle8*i* Server Configuration Parameters

Windows 2000 Tuning

Before or immediately following the installation of Oracle8*i* Server, Compaq recommends making these changes to Windows 2000:

- Applying the latest operating system fixes from Microsoft
- Installing the latest Compaq drivers
- Installing the appropriate *Compaq SMART Array* controller drivers

Applying the Latest Fixes

Install the latest version of the Microsoft Windows 2000 Service Pack to apply the latest operating system fixes from Microsoft.

Installing the Latest Compaq Drivers

Install the latest version of the Compaq Software Support Diskette (SSD) to apply the latest drivers for Windows 2000 on Compaq hardware. You can obtain a complete listing of available drivers from: http://www.compaq.com/support/files.

Installing Compaq SMART Array Controller Drivers

Microsoft distributes a default **Miniport** driver for *Compaq SMART Array* controllers with Windows 2000; Compaq distributes drivers and updates on the SSD.

Compaq has developed a **Non-Miniport** driver for its *SMART Array* 53xx controller family, which in some cases, can improve performance. The Non-Miniport driver, contained in SoftPaq 14450, can be obtained from: www.compaq.com.

- The Non-Miniport driver requires less processor overhead for performing I/O and can decrease overall processor use, allowing the processor(s) to accommodate a heavier workload. If your system is processor-bound, you should consider using the Non-Miniport driver.

You should be aware of these restrictions on the use of Non-Miniport drivers:

- The Non-Miniport driver currently only supports the *SMART Array* 53xx controllers for Windows 2000.
- These applications do not currently support the Non-Miniport driver:
  - *Compaq Insight Manager* agents for Windows 2000 and *Compaq Insight Manager*
  - Compaq Standby Recovery Server Option
  - Compaq Online Recovery Server Option
  - *Compaq Array Configuration Utility* (when used online under Windows 2000)

Note: If you require any of these applications to support your *SMART Array* controllers, use the Miniport driver.

Configuring Physical Memory

Windows 2000 Server and Windows 2000 Advanced Server continue to support the increase in the per-process address limit from 2GB to 3 GB. Windows 2000 Server supports 4196MB of RAM. Windows 2000 Advanced Server supports 8192MB of RAM and uses Physical Address Extensions (PAE) to access memory above 4GB of RAM. This benefits applications that run on *Compaq ProLiant* servers with more than 2GB of physical RAM, and can take advantage of a larger address space.

If you have more than 2GB of RAM, modify the **boot.ini** file by adding the **/3GB** parameter to the startup line. If you have 4GB of RAM or less, you should add /NOPAE to the startup line. If you will be using more than 4GB of RAM, you should add /PAE to the startup line.

For example:

Using 4GB of RAM or less:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT=">3GB Microsoft Windows
2000 Advanced Server" /fastdetect /3GB /NOPAE
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows 2000
Advanced Server" /fastdetect
```

Using greater than 4GB of RAM:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT=">3GB Microsoft Windows
2000 Advanced Server" /fastdetect /3GB /PAE
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows 2000
Advanced Server" /fastdetect
```

You must re-boot after modifying the **boot.ini** file.

If you are using more than 4GB of RAM, you must also grant the "Lock memory pages" Windows 2000 Advanced Server privilege to the Oracle user. In Windows 2000 Advanced Server, Start→Run→gpedit.msc will display the Group Policy screen. Select Local Computer Policy→Windows Settings→Security Settings→Local Policies→User Rights Assignment. Double click on "Lock pages in memory" policy and assign the Local Setting to Administrators. See Figure 7.

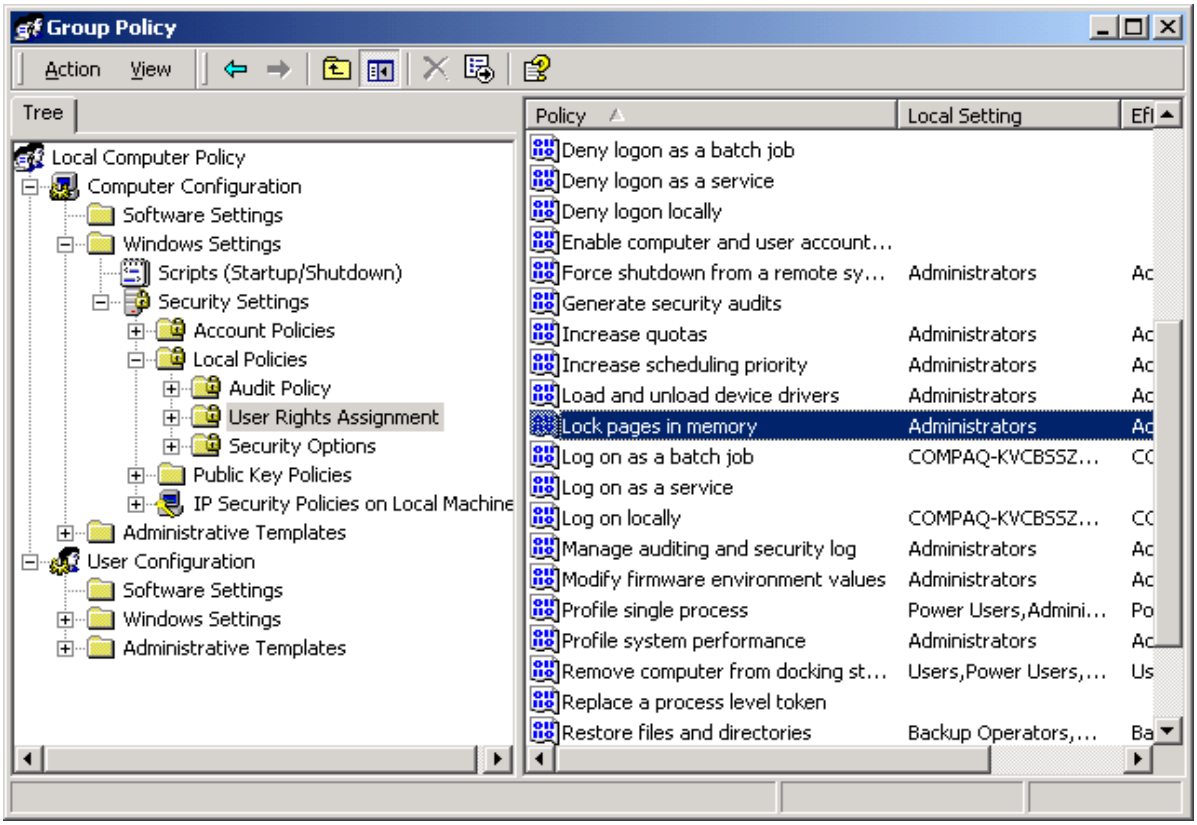

Figure 7. Windows 2000 Advanced Server Group Policy Screen

To take advantage of the increased Windows 2000 address limit, run **imagecfg** with the **-l** option on the Oracle8*i* Server executable. This enables Oracle8*i* Server to use an address space larger than 2GB:

```
imagecfg -l oracle.exe
```

Note: You can find **imagecfg** in the Support folder of the Windows 2000 Server or Windows 2000 Advanced Server CD-ROM.

Configuring Virtual Memory

The size of your virtual memory is the size of the physical memory (RAM) plus your swap file size. Ideally, the size of the swap file should be approximately one to one and a half times that of the RAM to ensure that there is enough space to hold a memory dump.

You can set virtual memory from Control Panel→System. Select the Advanced tab→Performance Options and click on "Change to Set Virtual Memory Size".

Note: If you are using amounts of RAM larger than the disk space that can be spared, it may not be practical to set your swap file to one to one and a half times that of your physical memory.

Configuring Server Memory

To optimize server memory for Oracle8*i* Server, use Control Panel→Network and Dial-Up Connection→Local Area Network→Properties. Highlight 'File and Printer Sharing for Microsoft Networks' and select Properties to re-configure server memory. Change the relationship[5] between the memory allocated to network connections and the memory allocated to applications running on the server by selecting "Maximize Throughput for Network Applications." See Figure 8.

Note: Oracle8*i* Server performs its own memory management for caching file and network I/O.

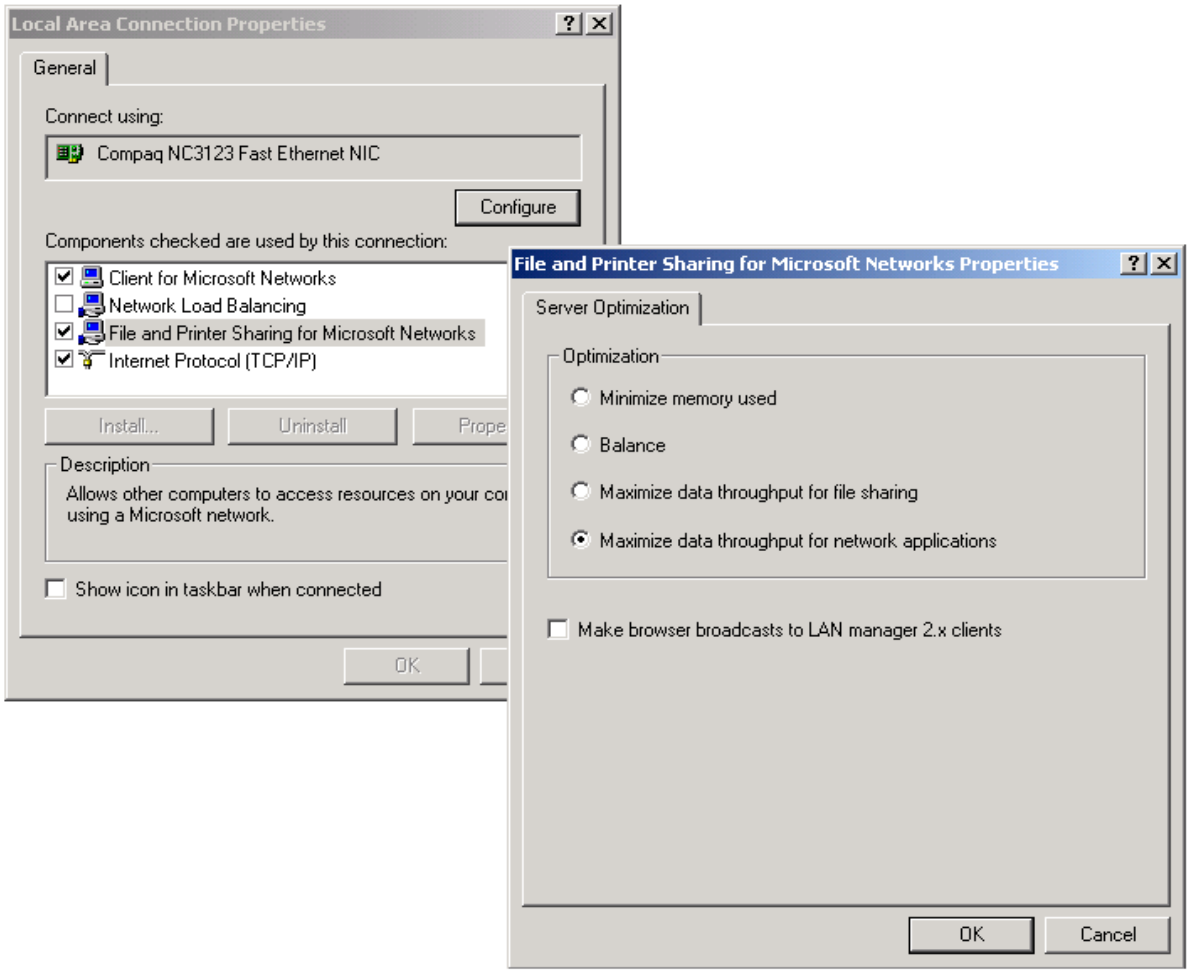

Figure 8. Configuring Server Memory

[5] The default is "Maximize Throughput for File Sharing."

Removing Network Protocols

Use Control Panel→Network and Dial-Up Connection→Local Area Connection→Properties to obtain the Network Protocols list. Deselect components in the "Components checked are used by this connection" list to remove any protocol not required for server activities. See Figure 9.

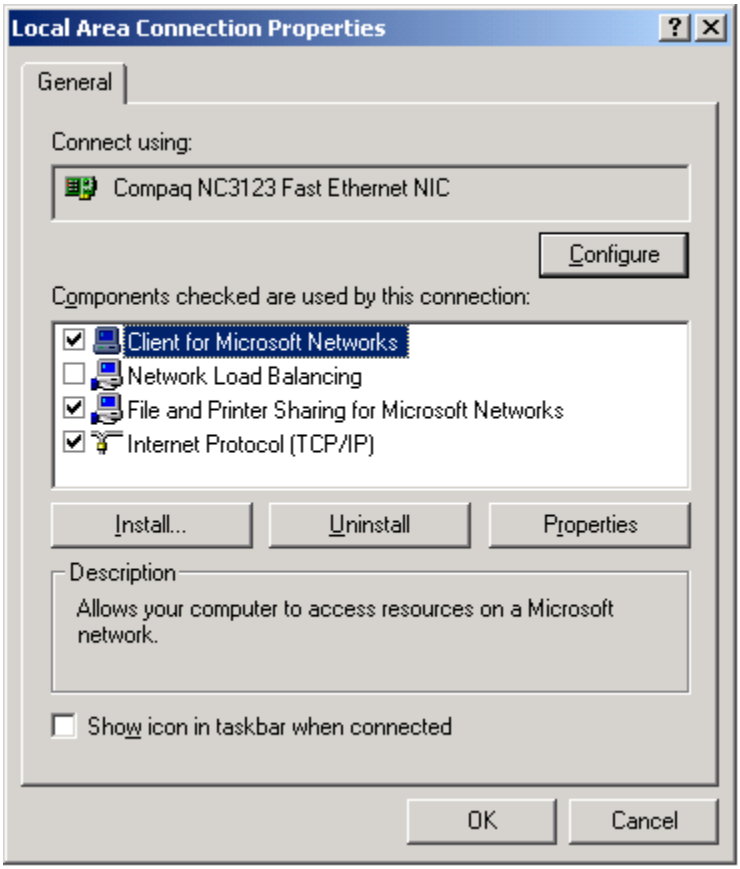

Figure 9. Removing Unused Network Protocols

Configuring the Local Area Connection Network Bindings

Use Control Panel→Network and Dial-Up Connection. To configure the Local Area Connection network bindings path, click on the Advanced tab and select "Advanced Settings". Move the most important protocol to the top of the list to reduce the response time for that protocol; position the remaining protocols in descending order of importance. See Figure 10.

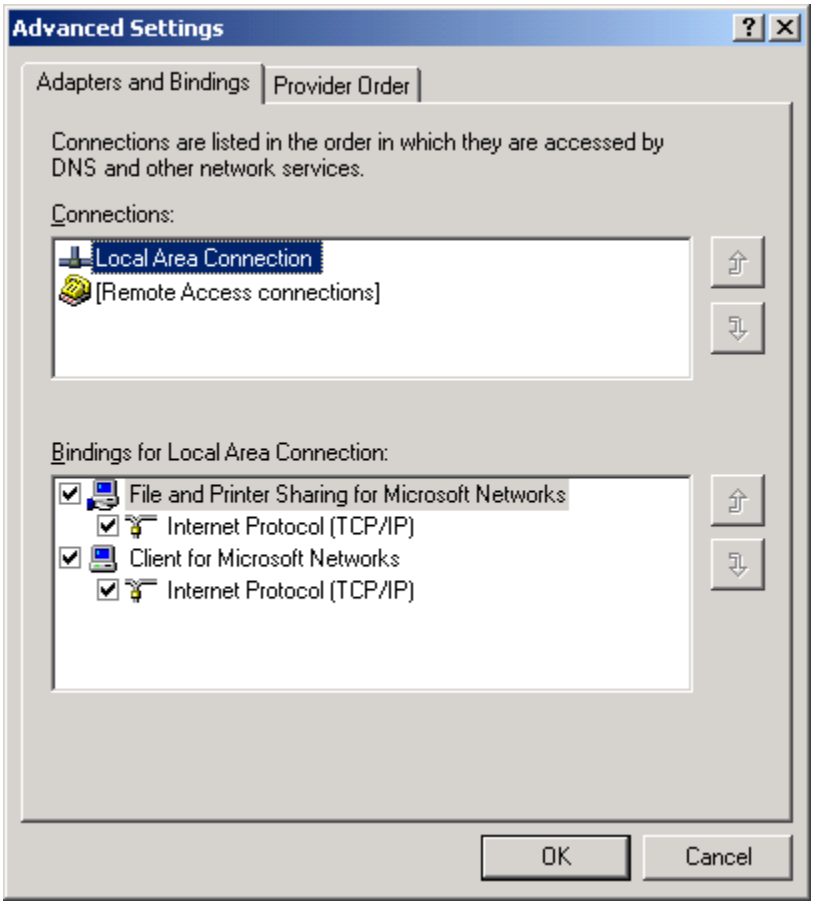

Figure 10. Configuring the Protocol Binding Order

Stopping Services

If the Oracle8*i* Server instance is dedicated to being a database server only, you can stop unnecessary services to reduce overhead on the processor(s). The minimum services required for Oracle8*i* Server are:

- OracleServiceSSSS (where SSSS is the system ID)
- Net8 requires one or more services (depending upon the version of Net8 and the protocols supported) to support user connections such as OracleorantTNSListener.
- Compaq recommends starting the Server, EventLog, and Workstation services, as well as any management services you have installed. You can use the Compaq Software Support Disk to install the management services, which start automatically after installation.

You can turn services off using Programs→Administrative Tools→Computer Management→Services and Applications→Services. If desired, you can change the Startup options for your services to Manual. See Figure 11.

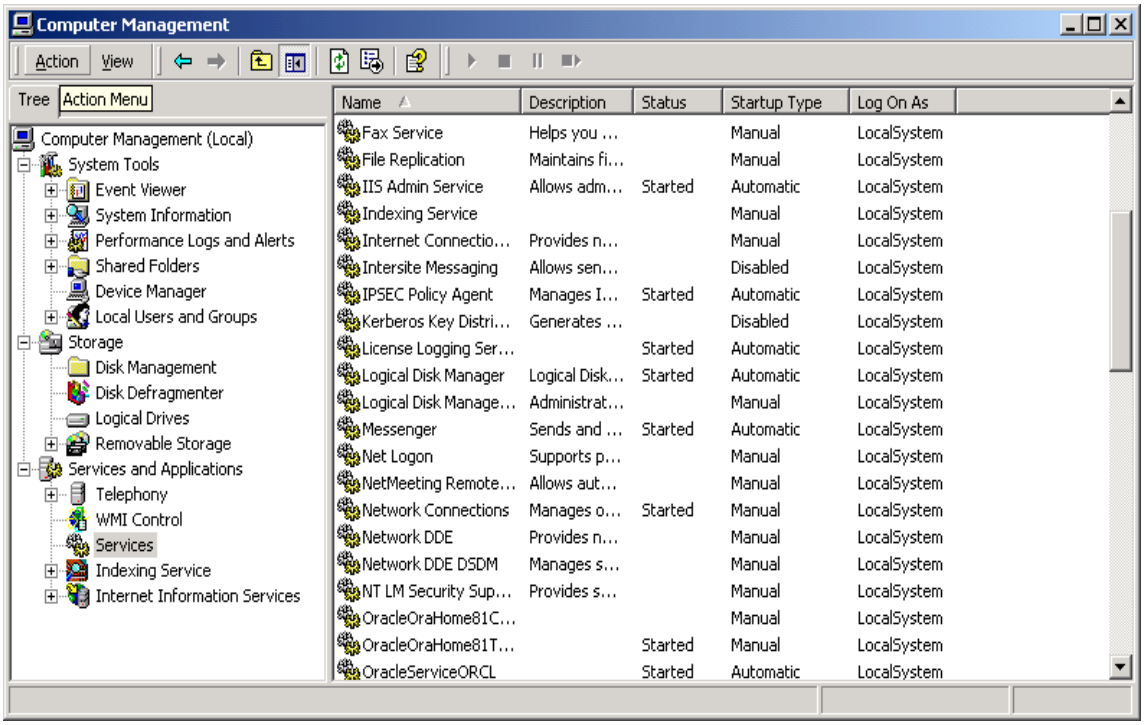

Figure 11. Starting and Stopping Services

Configuring the Windows 2000 File System

Compaq recommends:

- Placing Windows 2000 and Oracle8*i* Server executables on an NTFS partition.
- Placing Oracle8*i* Server system, log, and data files on NTFS or raw file partitions.

Historically, Oracle has achieved better performance when using raw devices over FAT or NTFS file systems. If your system is I/O-bound, you could switch the log and data files to raw partitions to improve I/O performance, but you are limited by the restrictions on using raw devices such as backup and copy. You can minimize the impact of these restrictions by using Oracle8*i* Server Recovery Manager's copy command and products such as MKS Toolkit's **dd** utility.

Each raw partition equates to a single file and can be represented by a single drive letter (such as **\\.\E:**) or a logical physical drive number (such as **\\.\PhysicalDrive0**) or, if you have many partitions per disk, by symbolic links (such as **\\.\ORA_PARTITION_1** where **ORA_PARTITION_1** is defined using Oracle's **setlinks** utility).

I/O Tuning

In most well-tuned Oracle systems, the I/O subsystem is not the performance bottleneck. While adding disk spindles often reduces I/O contention due to their random I/O access, optimizing the OLTP database layout makes a significant impact on performance.

This section focuses on tuning I/O for your OLTP system and provides information on these topics:

- Asynchronous I/O
- Sequential and random I/O
- Disk layout and partitioning
- Physical disk I/O limits
- Array controllers
- Availability versus performance

Asynchronous I/O

Oracle8*i* Server uses the asynchronous I/O capabilities of Windows 2000 by allowing the **DBWn** and **LGWR** threads to issue multiple I/Os simultaneously. This reduces the I/O time and allows the **DBWn** and **LGWR** threads to wake up any waiting Oracle threads sooner, thereby improving throughput and reducing response times.

Using the asynchronous I/O capabilities of Windows 2000 and Oracle8*i* Server means that only one **DBWn** thread is necessary in the system.

Sequential and Random I/O

To achieve maximum performance when Oracle8*i* Server is accessing data files sequentially, the disk(s) must be dedicated to this purpose. Oracle8*i* Server uses sequential disk access for these activities:

- Accessing transaction logs
- Making full table scans
- Making full index scans
- Making direct loads with SQL*Loader (reading from input files and writing to data files)
- Redo log archiving

Of primary importance are the Oracle8*i* Server transaction log files, which are generally accessed in a sequential, write-only fashion; when archiving logs or recovering the database, Oracle8*i* Server accesses transaction logfiles in a sequential, read-only fashion.

In a typical multi-user database system, file access is random. You should spread the data files over as many physical disks as necessary to achieve random I/O rates that do not exceed the recommendations given in Table 7. This is best achieved using the disk striping available with a *Compaq SMART Array* controller, which ensures that the load is balanced equally across all the disks in a volume, and allows a high degree of parallelism to occur on accesses. For more information on optimizing array configurations, refer to *Configuring Compaq RAID Technology for Database Servers*, which is available on the *Compaq ActiveAnswers* web site at http://www.compaq.com/activeanswers.

Disk Layout and Partitioning

Disk layout and partitioning are two of the most important areas of I/O tuning and OLTP performance optimization, and Compaq offers these recommendations based on testing carried out on *ProLiant* servers.

Disk Layout

You should consider these issues when attempting to optimize OLTP performance:

Note: It is helpful to know the partitioning requirements of your system before determining the disk layout.

- **Redo log access**: Redo log access is 100% sequential I/O and should be isolated, if possible. Ensuring fast access is essential; so, Compaq recommends configuring the drives containing the redo logs for hardware fault tolerance using RAID 1 to maximize performance and reliability. Redo logs are required to restore the Oracle database, should data be lost. Therefore, it is important that your redo logs have fault tolerance. For more information, refer to *Configuring Compaq RAID Technology for Database Servers*, which is available on the *Compaq ActiveAnswers* web site at http://www.compaq.com/activeanswers.
- **Redo log archive files**: Redo log archive files are also 100% sequential I/O access. You should isolate these files for maximum performance and configure the drives containing the files for hardware fault tolerance using RAID 1. RAID 5 or Advanced Data Guarding could also be used if cost is an issue.

- **Data files**: Data file access is usually random and should be balanced across all drives allocated to data and indexes. Increasing the number of physical drives allows you to achieve faster I/O rates; using a striped array ensures that I/Os are well distributed.

Disk Partitioning

Once you have determined your disk layout, you can configure the disks and create partitions. Use the *Compaq Array Configuration Utility* to create logical disks, and configure the array controller cache and RAID levels; use the Windows 2000 Disk Administrator to split the logical drives into partitions. See Figure 12.

Note: To create a partition, select Start→Programs→Administrative Tools→Disk Management. Right click on the disk and select "Create Logical Drive".

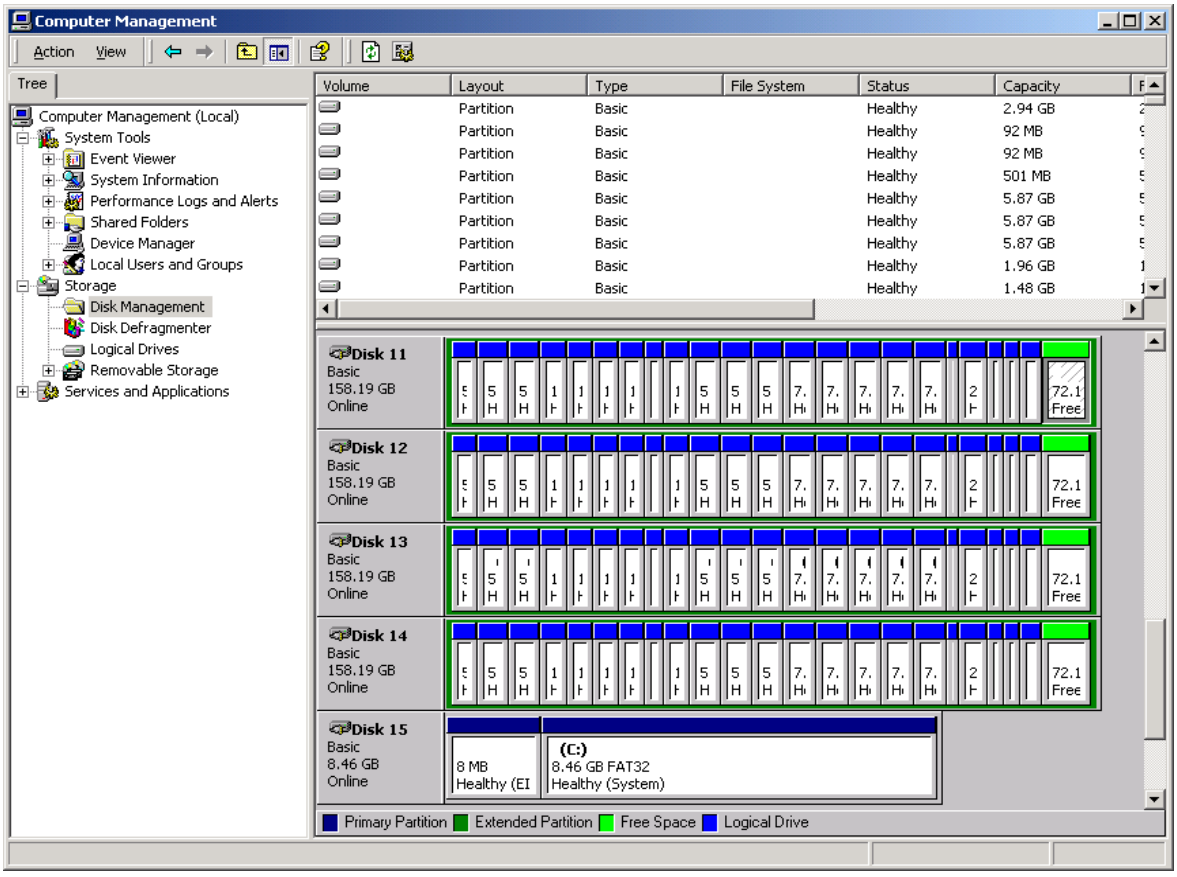

Figure 12. Windows 2000 Disk Administrator

Partitioning tables and indexes in Oracle8*i* Server provides a more granular table and index layout, and allows the better management of table and disk storage. When setting up partitions, you should consider these issues:

- **Table partitioning**: You can partition any table that is not part of a cluster and does not contain Long or Long Raw data types. Oracle8*i* Server supports range, hash, and composite partitioning.

  - **Range partitioning** uses a range of column values (partitioning key) to map rows or index entries to partitions. The partitioning key consists of an ordered list of up to sixteen columns, and is based on the partitioning column values. You should determine which data values you wish to partition; the most common range partitioning is by date.
  - **Hash partitioning** uses a hash function on the partitioning columns to stripe data into partitions. It is beneficial to use hash partitioning if you do not know the amount of data in a range, if the size of range partitions differ substantially, or if partition pruning and partition-wise joins on a partitioning key are important.
  - **Composite partitioning** uses the range method to partition data and, within each partition, uses the hash method to sub-partition the data. Composite partitioning provides both management benefits and data placement and parallelism advantages.

- **Index partitioning**: You can partition an index that is not a cluster index and is not defined on a cluster table. Oracle8*i* Server supports two primary types of index partitioning, local and global:
  - In a **local** partitioned index, all keys in a particular index partition refer only to rows stored in a single underlying table partition[6].

Note: A bitmap index on a partitioned table must be a local index.

  - In a **global** partitioned index, the keys in a particular index partition may refer to rows stored in more than one underlying table partition[7].

  In addition, Oracle8*i* Server supports these index types:

  - Prefixed and non-prefixed local indexes
  - Prefixed global indexes

  Prefixed indexes are partitioned on a left prefix of the index columns, while a non-prefixed index is partitioned on something other than a left prefix. For example: a **SALES** table and its index **SALES_INDEX** are both partitioned on the **MONTH** column, making **SALES_INDEX** a local index. If the **SALES_INDEX** is defined on the columns **MONTH** and **SALES_AMT**, then **SALES_INDEX** is local prefixed because **MONTH** is the left prefix of the index columns. If **SALES_INDEX** is defined on the **QTR** column, then **SALES_INDEX** is a local non-prefixed index. See Table 6.

Table 6. Comparison of Prefixed and Non-Prefixed Indexes

Local Prefixed Index	Sales Table	Local Non-Prefixed Index
SALES_INDEX	**SALES**	**SALES_INDEX**
Partitioned by **MONTH**	Partitioned by **MONTH**	Partitioned by **MONTH**
(**MONTH**, **SALES_AMT**)	(**MONTH**, **QTR**, **SALES_AMT**)	(**QTR**, **SALES_AMT**)

A global index follows the same left prefix rule as a prefixed local index.

[6] All keys in a particular index partition map to all rows in a single underlying table partition.
[7] All indexes in one partition may map to rows in multiple table partitions.

Local indexes facilitate index manageability and provide more availability when there are partition or sub-partition maintenance operations on the table. When partitions in underlying tables are added, dropped, merged, or split, or when hash partitions or sub-partitions are added or coalesced, Oracle8*i* Server maintains the index partitioning for local indexes on that table to ensure that the local index remains equi-partitioned[8] with the table.

In an OLTP environment, global and local prefixed indexes provide better performance than local non-prefixed indexes because they minimize the number of index partition probes. Local non-prefixed indexes are useful for historical databases[9] where the index is defined on one column (such as **SALES_NO**) for faster access but partitioned on another column (such as **MONTH**) to ease partition maintenance.

- **Storage parameters**: All partitions of a table or index have the same logical attributes, but their physical attributes may differ. You can assign each partition to its own tablespace and specify unique physical attributes (such as **PCTFREE, PCTUSED, INITRANS, MAXTRANS**) and storage parameters (**INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, FREELIST, FREELIST GROUPS**).

For more information on partitioning, please refer to the *Oracle8i Server Concepts Manual* or the *Oracle8i Server Application Developer's Guide*.

I/O Limits

Typically, an Oracle8*i* OLTP system can be characterized as many users acting on a small subset of data throughout the database. The resulting I/O profile includes very heavy random reads and writes across the data and index files, while transaction logs receive a heavy stream of sequential write operations of 2KB or less. In this case, it is important to monitor the I/O rate on the disk subsystem and ensure that individual disks are not overloaded with random I/Os.

The (PhysicalDisk:Disk Transfers/sec), (PhysicalDisk:Disk Reads/sec), and (PhysicalDisk:Disk Writes/sec) counters in the Windows 2000 Performance Monitor display the number of I/O operations per second handled by the disk subsystem. If you have a *Compaq SMART Array* controller with several drives allocated to a logical volume, the counters monitor the total number of disk transfers to the entire volume. To calculate the number of transfers per second for each individual disk drive, you must first divide the number of I/O operations per second by the number of disk drives and then factor in the appropriate RAID overhead.

When monitoring and optimizing the drive subsystem, Compaq recommends that you do not exceed the values for I/O requests (disk transfers) per second per disk drive given in Table 7. For more information, refer to *Configuring Compaq RAID Technology for Database Systems* available on the *ActiveAnswers* web site at http://www.compaq.com/activeanswers.

Table 7. Average Disk Seconds Per I/O Operation

Type of 2KB I/O Request	Compaq Hard Drive					
	9.1-GB (10K RPM) Ultra2	9.1 -GB (10K RPM) Ultra3	9.1-GB (15K RPM) Ultra3	18.2-GB (10K RPM) Ultra2	18.2-GB (10K RPM) Ultra3	18.2-GB (15K RPM) Ultra3
Random Read/Write	85	100	120	85	100	120

[8] The index and table are equally partitioned.
[9] Databases that track information over a time interval.

Use this procedure to estimate the average number of I/O requests per second for each disk drive:

1. Using Windows 2000 Performance Monitor, determine the number of (PhysicalDisk:Disk Transfers/sec), (PhysicalDisk:Disk Reads/sec), and (PhysicalDisk:Disk Writes/sec). Average these figures out over an extended time period rather than selecting a few minutes of peak activity (such as a checkpoint).
2. Use the following equations to calculate I/Os per second per disk drive.
   - **RAID 0**: I/Os per second per drive = (PhysicalDisk:Disk Transfers/sec) / Number of drives in the volume
   - **RAID 1**: I/Os per second per drive = ((PhysicalDisk:Disk Reads/sec) + 2*(PhysicalDisk:Disk Writes/sec)) / Number of drives in the volume
   - **RAID 5**: I/Os per second per drive = ((PhysicalDisk:Disk Reads/sec) + 4*(PhysicalDisk:Disk Writes/sec)) / Number of drives in the volume
   - **Advanced Data Guarding**: I/Os per second per drive = ((PhysicalDisk:Disk Reads/sec) + 6*(PhysicalDisk:Disk Writes/sec)) / Number of drives in the volume
3. Repeat these steps for each logical volume. For more information on planning and configuring Compaq RAID technology, please refer to *Configuring Compaq RAID Technology for Database Servers* on the *ActiveAnswers* web site at http://www.compaq.com/activeanswers.

In the example displayed in Figure 13, there is no fault tolerance (RAID 0) on logical NTFS volume T. The average I/O rate for Disk Reads/sec is 111.662 I/Os per second. The average I/O rate for Disk Writes/sec is 192.412 I/Os per second. The total average I/Os per second would be (111.612 + 192.412) or a total of 304.074 I/Os per second. If the drive set consists of two 9.1-GB (10K RPM) Ultra3 drives, then the I/O rate is 304.074 / 2 = 152.037 I/Os per second per drive. The test purposely flooded the logical drive with I/Os, giving a scenario above the recommended 100 I/Os per second per disk.

If the average I/O rate values significantly exceed those suggested in Table 7, increase the speed of your disk subsystem by adding more drives or using faster drives. Going to three 9.1-GB (15K RPM) Ultra3 drives or four 9.1-GB (10K RPM) Ultra3 drives is appropriate to achieve the recommended disk I/O per second rate.

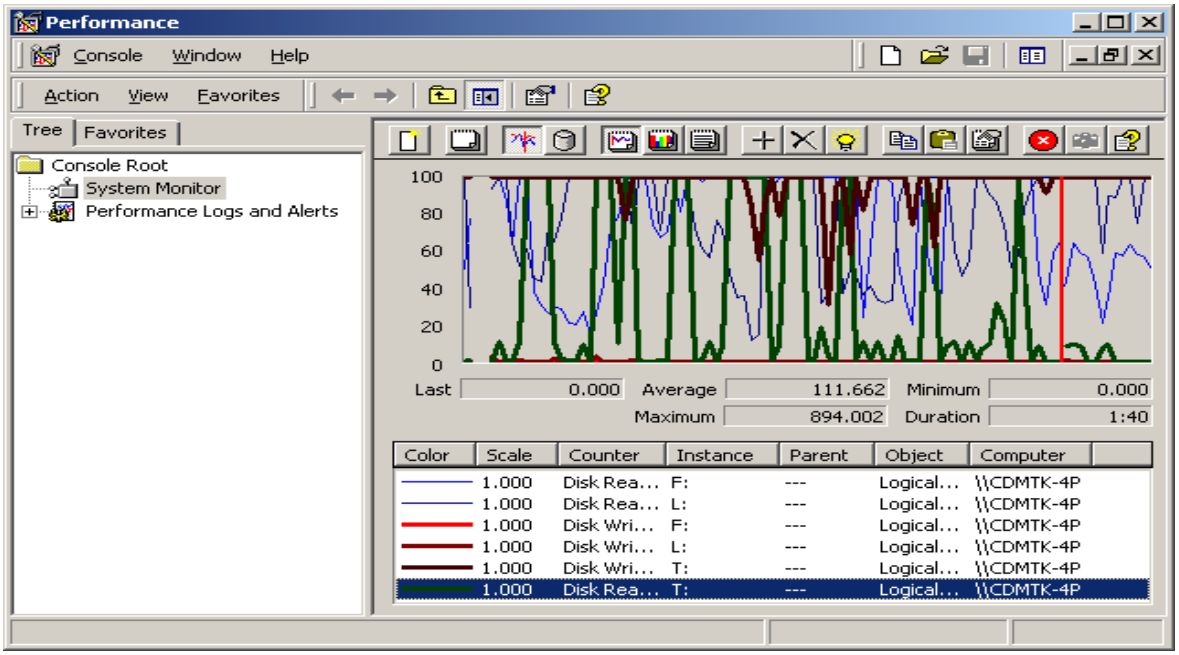

Figure 13. Windows 2000 Performance Monitor Showing Disk I/O

Array Controllers

Compaq recommends using *SMART Array* controller technology rather than generic SCSI controllers. *SMART Array* controller technology is present in Compaq SCSI and Fibre Channel storage solutions and brings these benefits:

- Both Compaq storage solutions provide higher throughput, large drive arrays, and multiple disk striping.
- The Compaq Fiber Channel storage solution provides shared and distributed storage benefits.
- Both Compaq storage solutions provide hardware RAID protection, including multiple RAID levels on a single drive array and hardware RAID protection for your boot drive(s).

Compaq SCSI Storage Solutions

Consider these issues when deploying *Compaq SMART Array* 3200, 4200, and 4250ES[10] solutions:

- You can configure up to 56 disks[11] as one or more logical drives. When multiple disks are collected into a single array, the data on logical drives created on that array are striped across all disks for even distribution. This balances your I/O load more evenly and works well for random I/O loads.
- You can configure multiple logical drives on each disk or array of disks. The maximum number of logical drives for a single controller is 32.
- You can configure the 56MB accelerator cache as either read-ahead cache or write-posting cache or a combination of the two[12] depending on the number of database writes. The cache memory is battery-protected.
- Rather than using software RAID on Windows 2000, you can define RAID levels at the controller level to relieve system processor overhead.

Consider these issues when deploying *Compaq SMART Array* 5300 solutions:

- You can configure up to 56 disks as one or more logical drives. When multiple disks are collected into a single array, the data on logical drives created on that array are striped across all disks for even distribution. This balances your I/O load more evenly and works well for random I/O loads.
- You can configure multiple logical drives on each disk or array of disks. The maximum number of logical drives for a single controller is 32.
- Depending upon the version of the Compaq SMART Array 5300, you can configure up to 128MB accelerator cache as either read-ahead cache or write-posting cache or a combination of the two[13] depending on the number of database writes. The cache memory is battery-protected.
- Rather than using software RAID on Windows 2000, you can define RAID levels at the controller level to relieve system processor overhead. RAID Advanced Data Guarding will be available as of January 2001. Firmware upgrades for Advanced Data Guarding on the SA5304/128 are available from www.compaq.com/smartarray.
- 64-bit memory addressing supports *ProLiant* servers with more than 4GB of RAM.

You can find more information on *Compaq SMART Array* controllers at http://www.compaq.com/products/storageworks/array-and-scsi-controllers/controller_options.html.

[10] This is an internal controller. The number of drive bays in the system limits the number of drives supported.
[11] The *Compaq Smart Array* 3200 is limited to 28 disks per one or more logical drives.
[12] Your choices for allocating the memory are 100/0 (100% read-ahead cache with 0% write-posting cache), 75/25, 50/50, 25/75, or 0/100.
[13] Your choices for allocating the memory are 100/0 (100% read-ahead cache with 0% write-posting cache), 75/25, 50/50, 25/75, or 0/100.

Compaq Fibre Channel Storage Solutions

The Fibre Channel RAID Array 4000 and 4100 storage solutions combine the benefits of a Fibre Channel with the *Compaq SMART Array* controller. Consider these issues when deploying a Fibre Channel solution:

- You can configure up to 12 disks as one or more logical drives. When multiple disks are collected into a single array, the data on logical drives created on that array are striped across all disks for even distribution. This balances your I/O load more evenly and works well for random I/O loads.
- You can configure multiple logical drives on each disk or array of disks.
- You can configure the accelerator cache as either read-ahead cache or write-posting cache or a combination of the two[14] depending on the number of database writes. The accelerator memory is 64MB[15] of which 48MB is battery-powered and is available for read/write; 64MB is available for read-ahead cache.
- Rather than using software RAID on Windows 2000, you can define RAID levels at the controller level to relieve system processor overhead.

You can find more information on the Compaq Fibre Channel RAID Array 4000 and 4100 storage solutions at http://www.compaq.com/products/storageworks/raidstoragesystems.html.

Availability versus Performance

Once you have calculated the disk space required for your OLTP system, it important to determine the appropriate level of protection for your drives. Database availability, performance, and cost are considerations when determining the *Compaq SMART Array* controller RAID level to use.

Note: The RAID level you select directly affects system performance and the physical disk space required for your OLTP system.

Compaq solutions support these RAID levels:

- **RAID 0**: RAID 0 has a 1:1 write ratio (one physical write for each logical write) and a 1:1 read ratio (one physical read for each logical read). All RAID 0 data is stored in one physical location. RAID 0 offers no fault tolerance for your drives, but because additional drives are not required for fault tolerance, this is the lowest cost option.
- **RAID 1**: RAID 1 has a 2:1 write ratio (two physical writes for each logical write) and a 1:1 read ratio. The **data mirroring** feature of RAID 1 provides fault tolerance by storing data in two physical locations and provides the additional benefit of **split reads**.
  - Split reads provide a performance boost, especially when there are not many writes, since the database can read from either copy of the data.
  - Since twice the storage is required, RAID 1 is the highest cost option, but is the most fault tolerant.

[14] Your choices for allocating the memory are 100/0 (100% read-ahead cache with 0% write-posting cache), 90/10, 80/20, 70/30, 60/40, 50/50, 40/60, 30/70, 25/75, and 0/75.
[15] Early versions of Compaq Fibre Channel RAID Array 4000 have only 32 MB of RAM.

- **RAID 5**: RAID 5 has a 4:1 write ratio (two physical reads and two physical writes for each logical write) and a 1:1 read ratio. The distributed **data guarding** feature of RAID 5 provides fault tolerance, but because all RAID 5 data is stored in one physical location, there is no benefit from split reads. Parity is striped across all drives in the RAID 5 array.
  - The number[16] of disk drives required for a RAID 5 configuration is the number of drives required for data files plus one additional drive.
  - There is a substantial performance penalty of 20% to 35% during writes, depending upon your application.
  - With RAID 5, Compaq recommends limiting the number of physical drives to 14 for each logical drive.
- **Advanced Data Guarding**: The *SMART Array* 5300 family utilizes new patent, pending RAID innovation, Advanced Data Guarding (ADG). RAID ADG is an advanced RAID level that increases the number of sets of parity to two, creating multiple sets striped across the disks to provide unparalleled fault tolerance. While ADG does provide improved fault tolerance, keep in mind that there is a substantial performance penalty during writes, depending upon your application. ADG has a 6:1 write ratio, thus its performance penalty for writes is greater than that of RAID 5. More information on ADG and the *SMART Array* 5300 family of controllers is available at www.compaq.com/products/storage.

Compaq SMART Array controllers have the flexibility to use any of these RAID levels either individually or in combination.

Depending upon your database availability, performance, and cost requirements, it may be beneficial to use RAID 1 on critical drives, and RAID 0 or RAID 5 on other drives. Compaq recommends placing log drives on a RAID 1 drive to optimize performance and ensure that no log files are lost in the case of a single disk failure.

For more information on *Compaq SMART Array* controller RAID selection and implementation, refer to *Configuring Compaq RAID Technology for Database Servers*, which is available on *ActiveAnswers* at http://www.compaq.com/activeanswers.

Processor Scaling

While no operating system or hardware provides 100 percent scalability, Windows 2000 and Oracle8*i* Server provide above-average levels of scalability so that, if your system is processor-bound, you can improve system performance by adding processors. Windows 2000 Server supports scalability up to 4 processors, and Windows 2000 Advanced Server supports scalability up to 8 processors.

Because well-tuned Oracle8*i* Server instances are typically system processor-bound, most applications see significant improvements after the addition of more powerful processors, more processors, or more processor cache. However, if the system is memory-bound or I/O-bound, adding system processors will not provide a significant improvement.

Figure 14 provides a graphical representation of system processor scalability on the *Compaq ProLiant* 8500 using tests that are both disk- and memory-intensive.

[16] There must be a minimum of three drives for a RAID 5 array.

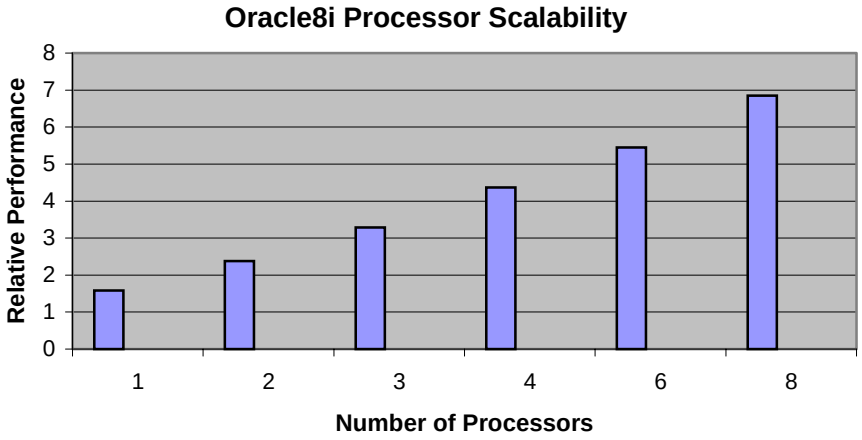

Figure 14. Graphical representation of System Processor Scalability

Memory Tuning

Oracle8*i* Server 8.1.6 for Windows 2000 supports 4GB of RAM on Windows 2000 Server and 8GB of RAM on Windows 2000 Advanced Server. Oracle8*i* Server 8.1.6 for Windows 2000 Advanced Server uses Address Windowing Extensions (AWE) to allow access to memory above 4GB. The memory above 4GB can only be used for database block buffers.

There continues to be a 3GB memory limit for Oracle8*i* on Windows 2000. For Windows 2000 Advanced Server, this limit does not include the Oracle buffer cache, but does include the **`AWE_WINDOW_MEMORY`** size. Therefore, it is important to tune your Oracle8*i* Server so there is enough memory for the user threads (PGA), the non-buffer cache components of the SGA, and **`AWE_WINDOW_MEMORY`**. For Windows 2000 Server, it is important to tune your Oracle8*i* Server so that there is enough memory for the user threads (PGA) and the SGA.

Note: Allocating too much memory to the SGA decreases the memory available for the PGA.

Compaq offers memory tuning recommendations in this section for:

- Database block buffers
- Multiple buffer pools
- Database block size
- SGA size
- User capacity
- PGA size
- Number of threads
- Windows 2000 memory

Database Block Buffers

To reduce excess I/O, tune the Oracle8*i* Server cache buffers to maximize the use of the available resources using the Oracle8*i* Server initialization file (**Init*SID*.ora)** parameter **DB_BLOCK_BUFFERS**.

Since Oracle8*i* Server is a single process, multi-threaded application, all threads share the memory from the single process. To determine how much memory is available for buffers, check the Performance Monitor (Memory: Available Bytes) to identify the amount of memory available in the system. See Figure 15.

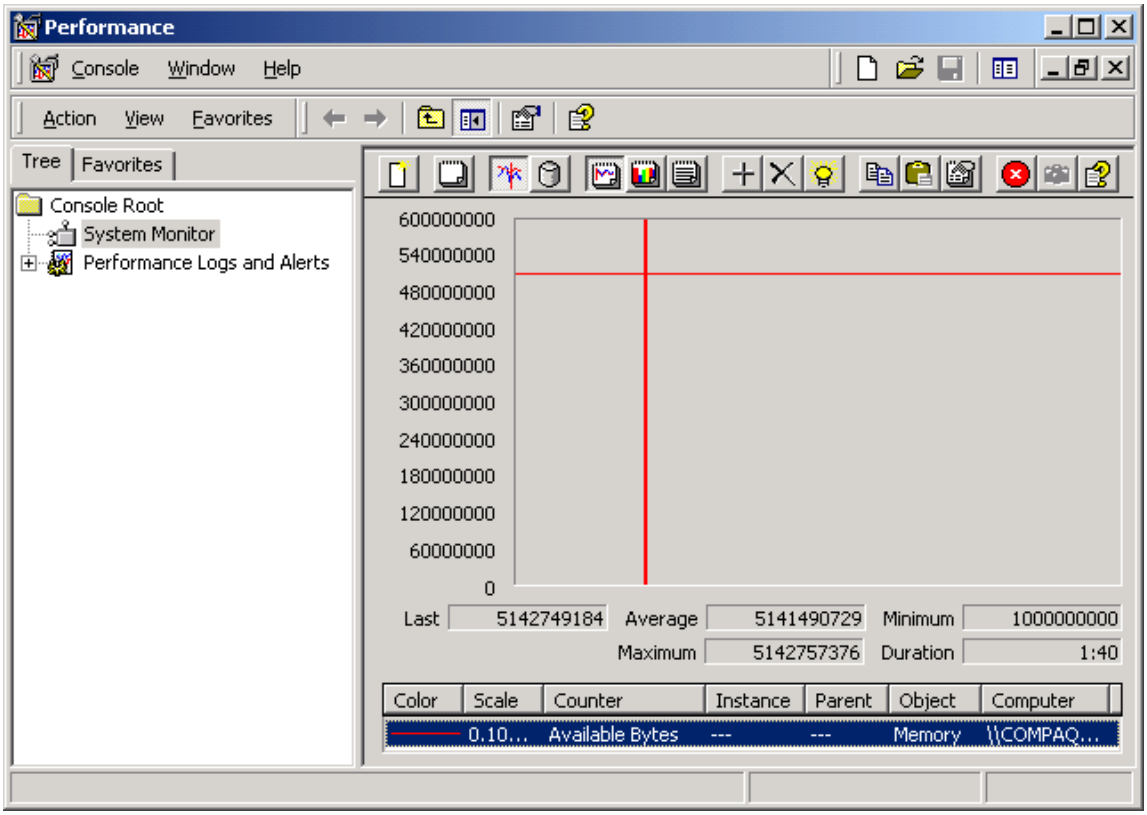

Figure 15. Performance Monitor Showing Available Memory

You should allocate a minimum of 128MB to Windows 2000 and, if your system is not a dedicated database server, you should determine how much memory is required for other applications running on the system. Adding more physical memory to your system[17] allows you to increase the size of **DB_BLOCK_BUFFERS**.

[17] Up to 4GB.

Oracle8*i* version 8.1.6 on Windows 2000 Advanced Server allows up to 8GB RAM in the system. If you plan to use more than 4GB RAM, the memory above 4GB RAM can be only be used for **DB_BLOCK_BUFFERS**. The **AWE_WINDOW_MEMORY** registry parameter specifies how much of Oracle's 3GB address space to reserve for mapping **DB_BLOCK_BUFFERS**. The default value for **AWE_WINDOW_MEMORY** is 1073741824. When the **AWE_WINDOW_MEMORY** value is increased, more **DB_BLOCK_BUFFERS** can be mapped at one time and thus improves performance. However, an increase in **AWE_WINDOW_MEMORY** decreases the available memory for connections into the database. Decreasing **AWE_WINDOW_MEMORY** will increase the available memory for connections, but will decrease performance due to less memory allocation for mapping **DB_BLOCK_BUFFERS**.

- Use the Windows 2000 Registry Editor to change the **AWE_WINDOW_MEMORY** registry parameter under **\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME(OracleID),** where **OracleID** is the ID number found in the **oracle.key** file in the **oracle_home\bin** directory.

Note: If you use all available memory for buffers, **swapping** may occur. The performance degradation caused by swapping is greater than the improvement obtained by tuning the Oracle8*i* Server buffers.

Multiple Buffer Pools

You can allocate schema objects to particular cache buffer pools. Oracle8*i* Server places all blocks read from the object in the appropriate buffer pool.

Oracle8*i* Server supports the following buffer pools:

- **KEEP**: The **KEEP** buffer pool maintains objects in the buffer cache. Small, frequently accessed tables that require a quick response time are good candidates for the **KEEP** pool.
- **RECYCLE**: The **RECYCLE** buffer pool recycles objects to eliminate the consumption of unnecessary cache space. A very large, infrequently accessed table with random I/O is a good candidate for the **RECYCLE** pool.
- **DEFAULT**: Oracle8*i* Server maintains the **DEFAULT** buffer pool for objects that have not been assigned to either the **KEEP** or **RECYCLE** pools.

For more information on enabling multiple buffer pools, refer to the *Oracle8i Tuning Guide*.

Database Block Size

You should select the value for the database block size parameter, **DB_BLOCK_SIZE**, based on your most critical type of transaction load.

Note: The default value for **DB_BLOCK_SIZE** is 8192.

Compaq makes these recommendations for database block size:

- If performance is most critical during applications that primarily access the database in a random fashion (small reads or updates scattered throughout the database), use a block size of 2048.
- If most applications are accessing the database sequentially when performance is most critical, use a block size of 16384.
- If you have a split of random and sequential access during the most critical time, use a block size of either 4096 or 8192 depending upon the ratio of random to sequential access.

System Global Area (SGA) Size

Note: Since the SGA resides in server memory, you must allocate this memory at the startup of the instance.

The SGA is made up of the following components:

- **Fixed size**: The Oracle products that you have installed determine the fixed size.
- **Variable size**: Oracle8*i* Server initialization file (**Init*SID*.ora)** parameters such as **SHARED_POOL_SIZE, DB_BLOCK_BUFFERS**, and **DB_BLOCK_LRU_LATCHES** determine the variable size. In addition, the variable size includes the buffer headers for the database block buffers.
- **Database block buffers**: You can determine the size of the database block buffer component by multiplying **DB_BLOCK_BUFFERS** by **DB_BLOCK_SIZE**.
- **Redo Buffers**: You can determine the size of the redo buffer component by adding approximately 80KB of overhead to the value of the **LOG_BUFFER** parameter.

User Capacity

The number of users you wish to connect has an impact on tuning. A number of parameters are necessary to allow a specified number of connections into the database engine. Also, there may be several areas of operating system tuning required, depending on your configuration.

Oracle8*i* Server initialization parameters that impact user capacity include **PROCESSES**, **SESSIONS**, **CURSORS**, and **TRANSACTIONS**.

If the Oracle available memory (SGA and PGA) has been exceeded, users may not be able to connect to the database. If you exceed your user limit or if you wish to add more concurrent users, you can either decrease the SGA or PGA, increase the Oracle available memory, or reduce the memory required by your Oracle application(s).

Program Global Area (PGA) Size

The PGA is a region of memory that contains data and control information for a single connection. Since this memory must be available at connect time for a particular user, the amount of free server memory limits the number of concurrent connections.

The values of these parameters impact the size of the PGA:

- OPEN_LINKS
- DB_FILES
- LOG_FILES
- **HASH_AREA_SIZE**: Designates the largest size that can be used for hashing. Oracle8i Server will add only what is required during hashing to the PGA. For example, if **HASH_AREA_SIZE** is set to 10MB but only 1MB is needed for your hash area, Oracle8*i* Server adds only 1MB to the PGA.
- **SORT_AREA_SIZE**: Designates the largest size that can be used for sorts. Oracle8i Server will add only what is required during sorting to the PGA. For example, if **SORT_AREA_SIZE** is set to 10MB but only 1MB is needed to sort your data, Oracle8*i* Server adds only 1MB to the PGA.

Number of Threads

You should update the value of the Oracle8*i* Server initialization file (**Init*SID*.ora**) parameter **PROCESSES**, which specifies the number of threads that Oracle8*i* Server can create. This value must include the maximum number of concurrent connections, the two Oracle service threads, and the background threads.

Note: The **V_$PROCESS** and **V_$BGPROCESS** views contain information about the Oracle8*i* Server threads being used.

Windows 2000 Memory

In addition to the memory required for Oracle8*i* and its processes, you must tune the memory allocated to Windows 2000.

Note: You can monitor memory usage via the Windows 2000 Task Manager→Performance window.

The primary goals in tuning Windows 2000 memory are:

- **Reducing the amount of paging and swapping**: Windows 2000 uses paging or swapping to move information from one storage location (such as real memory, virtual memory, or disk) to another. Excessive paging or swapping can reduce performance.

  To monitor paging, use the Windows 2000 Performance Monitor→(Memory:Pages Input/sec), where (Memory:Pages Input/sec) records the number of pages read in from disk to resolve memory references to pages that were not present in memory when referenced. If paging-in[18] occurs, either increase the total memory on your system or decrease the amount of memory you have allocated to the SGA.

- **Fitting the SGA and PGA into main memory**: Since the purpose of the SGA is to store data for fast access, the SGA should always be part of main memory. You can cause Oracle8*i* Server to read the entire SGA into memory when you start your instance by setting the value of the **Init*SID*.ora** parameter **PRE_PAGE_SGA** to **YES**. This may increase the amount of instance startup time but typically decreases the time taken for Oracle8*i* Server to reach its full performance capacity after startup.

  Note: Setting **PRE_PAGE_SGA** to **YES** does not prevent Windows 2000 from paging or swapping the SGA after it is initially read into memory.

  As stated earlier in the "Memory Tuning" section of this guide, it is important for your entire SGA and PGA to fit into Oracle available memory to prevent paging or swapping.

Updating Init*SID*.ora

Earlier sections of this guide have discussed the tuning of certain Oracle8*i* Server initialization file (**Init*SID*.ora**) parameters; this section discusses additional **Init*SID*.ora** parameters and provides further information on the previously discussed parameters.

Table 8 provides information on tuning **Init*SID*.ora** parameters.

[18] (Memory:Pages Input/sec) is greater than zero.

Table 8. Init*SID*.ora Parameters

Parameter	Description
BUFFER_POOL_KEEP	The **BUFFER_POOL_KEEP** parameter defines the **KEEP** buffer pool. The number of buffers in the **KEEP** buffer pool must be at least 50 times the number of **DB_BLOCK_LRU_LATCHES** allocated to this pool.
BUFFER_POOL_RECYCLE	The **BUFFER_POOL_RECYCLE** parameter defines the **RECYCLE** buffer pool. The number of buffers in the **RECYCLE** buffer pool must be at least 50 times the number of **DB_BLOCK_LRU_LATCHES** allocated to this pool.
CURSOR_SPACE_FOR_TIME	The **CURSOR_SPACE_FOR_TIME** parameter specifies whether or not Oracle8*i* Server can de-allocate a shared SQL area from the library cache to make room for a new SQL statement. If there are no library cache misses in your OLTP system, setting this value to **TRUE** may slightly improve the performance of execution calls while preventing the de-allocation of private SQL areas until associated application cursors are closed.
DB_BLOCK_BUFFERS	The size of the buffer cache is equal to the value of the **DB_BLOCK_BUFFERS** parameter multiplied by the value of the **DB_BLOCK_SIZE** number. You can calculate the cache hit ratio from: Cache hit ratio = (1 – (physical reads) / (db block gets + consistent gets)) You can obtain the values for "physical reads," "db block gets," and "consistent gets" from the **V$SYSSTAT** view. If the cache hit ratio is less than 0.9, you should increase the **DB_BLOCK_BUFFERS** value, if possible. If an increase to the value of **DB_BLOCK_BUFFERS** improves the performance of your OLTP system, try another increase.
DB_BLOCK_SIZE	You must specify the **DB_BLOCK_SIZE** parameter before creating the database because a change in **DB_BLOCK_SIZE** requires the database to be rebuilt. Depending on the application, the value of **DB_BLOCK_SIZE** should be between 2KB and 16KB.
DB_FILE_MULTIBLOCK_ READ_COUNT	The value of the **DB_FILE_MULTIBLOCK_READ_COUNT** parameter determines how many database blocks are read with a single operating system read. Since the maximum I/O size of the *Compaq Smart Array* controller is 64KB, you should set the **DB_FILE_MULTIBLOCK_READ_COUNT** to 64KB divided by the value of the **DB_BLOCK_SIZE** parameter (or a multiple thereof). For a **DB_BLOCK_SIZE** of 4KB, Compaq recommends a **DB_FILE_MULTIBLOCK_READ_COUNT** of 16.
DB_BLOCK_LRU_LATCHES	The maximum number of **DB_BLOCK_LRU_LATCHES** is the lower of (# CPUs *2 * 3) or (**DB_BLOCK_BUFFERS** / 50). If you are using multiple buffer pools, you should set **DB_BLOCK_LRU_LATCHES** to a multiple of the number of processors in the system.
LOG_BUFFER	The value of the **LOG_BUFFER** parameter reserves a fixed space for the redo log buffer. If the ratio of redo log space requests to redo entries is greater than 1:5000, you should increase the size of **LOG_BUFFER** until the space request ratio stops falling.
SHARED_POOL_SIZE	The value of the **SHARED_POOL_SIZE** parameter should be large enough to hold all data dictionary, shared SQL, and compiled objects.

For more information on **Init*SID*.ora** parameters, refer to the *Oracle8i Tuning Guide*.

Network Tuning

Oracle8*i* Server uses Net8 to support connections using LU6.2, Named Pipes, SPX/IPX, and TCP/IP protocols. For more information on the setup and configuration of Net8, see the *Oracle Net8 Administrator's Guide*.

System Management

Since managing the operating system, the database, and the hardware is of great importance, Compaq recommends using *Compaq Insight Manager*.

Note: If the system is used in a mission-critical role, it is especially important that users notify the administrator of any fault in the system so that corrective action can be taken quickly.

Compaq Insight Manager

Insight Manager is a Windows-based utility[19], which uses SNMP in conjunction with operating system and driver agents on the server to monitor system hardware and operating system components and report alarms (hardware failures, imminent failures, or system degradation caused by a hardware problem). You can configure *Insight Manager* to page the system administrator if an alarm occurs.

Using the *Insight Manager* pre-failure warranty allows you to replace a failing hardware component under warranty before it can impact your system.

Note: If you installed the Compaq Non-Miniport driver on your system, you cannot use *Insight Manager*.

Conclusion

The information in this paper is not a complete tuning guide but supplements other tuning information provided by Oracle and Microsoft.

To achieve an optimal configuration, there are several areas that you should carefully plan and tune: the application, the hardware, the operating system, and the network.

The tuning process is iterative and should be performed several times to achieve the best performance possible. Compaq expects the information provided in this Technical Guide to facilitate the tuning process. Although all the guidelines provided here have been tested extensively, you should not assume that tuning a particular parameter always gives the desired result.

Compaq welcomes feedback on your configurations and experiences to help improve future information products. Please send any comments or suggestions to Compaq at ESE_AA_Solutions@compaq.com.

[19] The utility is available on the *Compaq SmartStart* CD-ROM.