# Configuring HP SIM for optimal performance in a large environment

<table>
<thead>
<tr>
<th>Introduction</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP Systems Insight Manager</td>
<td>3</td>
</tr>
<tr>
<td>Customizing HP SIM</td>
<td>3</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>3</td>
</tr>
<tr>
<td>System resources</td>
<td>3</td>
</tr>
<tr>
<td>Job cleanup for large number of managed systems</td>
<td>4</td>
</tr>
<tr>
<td>Remote Microsoft SQL Server (recommended)</td>
<td>5</td>
</tr>
<tr>
<td>HP SIM Sizer tool</td>
<td>5</td>
</tr>
<tr>
<td>Customizing HP SIM to your environment</td>
<td>6</td>
</tr>
<tr>
<td>Creating a hierarchy: One master CMS with several regions</td>
<td>6</td>
</tr>
<tr>
<td>Master CMS:</td>
<td>6</td>
</tr>
<tr>
<td>Regionalize CMS for:</td>
<td>6</td>
</tr>
<tr>
<td>Discovery related</td>
<td>6</td>
</tr>
<tr>
<td>IP range</td>
<td>6</td>
</tr>
<tr>
<td>Changing Timeouts and Retries</td>
<td>7</td>
</tr>
<tr>
<td>System and event management</td>
<td>8</td>
</tr>
<tr>
<td>Customizing the query system</td>
<td>8</td>
</tr>
<tr>
<td>Customizing the task system</td>
<td>11</td>
</tr>
<tr>
<td>Tuning MS Windows Server 2003</td>
<td>12</td>
</tr>
<tr>
<td>Using Performance Monitor to optimize Microsoft Windows 2003</td>
<td>12</td>
</tr>
<tr>
<td>Configuring the logging feature</td>
<td>13</td>
</tr>
<tr>
<td>Examining the log data</td>
<td>13</td>
</tr>
<tr>
<td>Performance Analysis of Logs (PAL) tool</td>
<td>13</td>
</tr>
<tr>
<td>File System Cache Settings</td>
<td>14</td>
</tr>
<tr>
<td>Virtual memory paging</td>
<td>15</td>
</tr>
<tr>
<td>Tuning disk input/output</td>
<td>16</td>
</tr>
<tr>
<td>Disk tuning—RAID 5 performance</td>
<td>16</td>
</tr>
<tr>
<td>Tuning a single disk</td>
<td>16</td>
</tr>
<tr>
<td>Configuring the network</td>
<td>16</td>
</tr>
<tr>
<td>Microsoft Windows Server 2003 performance tuning</td>
<td>17</td>
</tr>
<tr>
<td>CPU configuration</td>
<td>17</td>
</tr>
<tr>
<td>Memory</td>
<td>17</td>
</tr>
</tbody>
</table>
Introduction

HP Systems Insight Manager

HP Systems Insight Manager (HP SIM) is a Web-based management application that integrates existing management technologies with the latest advances in Web technology. It provides a proactive, easy-to-use, automated, cost-effective solution for managing distributed systems.

HP SIM allows for the management of standards-based, distributed computing environments. By enabling browser access to its components, HP SIM provides management of systems and groups of systems, anywhere at any time.

HP recognizes the challenges faced in managing distributed enterprise systems. The growing mix of networks, computer platforms, applications, and databases makes this task complex, especially in an era of shrinking budgets. Ideally, you want to manage the availability of all of these distributed resources with a minimal impact to your network and to system resources.

There are two types of data collected by HP SIM. They consist of asynchronous events (or traps) and polled data. Polled data is further broken down into status polling and data collection polling. Each imposes consequences on the system and the network. In general, you want to enable asynchronous events (or traps) to carry the burden of problem detection because SNMP traps, WMI events, and DMI indications are short messages that only impact network utilization periodically and are initiated when devices detect a problem.

Polling, on the other hand, causes regular, periodic traffic consisting of a request for status from the HP SIM management server, followed by responses from the polled systems. The more systems that are polled and the more frequently they are polled, the more network resources are consumed to manage the environment. This white paper provides data to enable the reader to balance these network consumables against optimum health of the devices in the network. For example, it is critical to monitor servers that perform important functions, while it is not as important to monitor each desktop or notebook as closely.

Customizing HP SIM

Because every network is unique, you might need to customize HP SIM so that it gives you the most productivity while using the least amount of network and system resources. HP SIM can be customized to better fit your needs in several ways (setting timeouts, altering tasks and queries, and adjusting Cluster Monitor, to name a few).

Microsoft SQL Server

While MS SQL Server (2005 and later) is self-tuning, there are several actions you can take to improve performance as it relates to HP SIM. MS SQL Server can be adjusted and tuned in many of the same ways. Use the information in this document to determine how to tune MS SQL Server around your system, network, and application needs.

System resources

The HP SIM system resources are divided into the following categories:

- **CPU**—The type, number, and speed of processors needed to support a given workload
- **Disks**—The number of drives needed to support the input/output requirements or the amount of storage space (in MB) required to hold the data
- **Memory**—The amount of memory needed to support a given workload
• Network—The number and type of network interface cards (NICs) needed to support workload demands.

Ideally, the CPU should be set up as the limiting resource, because processing power provides the most scalable solution. However, this requires an established balance between the remaining resources (disk, memory, and network). After this balance is achieved, adding more memory or additional processors does not necessarily improve system performance. The best way to improve performance is to increase CPU speed. It is also possible that the limiting resource might not be hardware related at all, but is instead related to the application itself. The following sections address what can be modified to enhance the daily performance of HP SIM.

**Job cleanup for large number of managed systems**

When running HP SIM in an environment which contains a large number of managed systems, the completed job output might exceed several thousand files. This can result in HP SIM consuming large amounts of memory, task page graphical user interface slowdown, or out of memory errors, requiring a restart of HP SIM.

If such behavior is noted, the following can be implemented to alleviate the consuming of memory and disk space. Users may adjust the retention values for completed tasks, which can be altered as desired to reduce resource consumption.

The thresholds for task results data retention can be found on the Task Results Settings page (Figure 1). The Settings page can be displayed by selecting Tasks and Logs->View Task Results->Customize from the HP SIM menus. Historical task retention can be fine tuned by adjusting the values in order to achieve the desired balance of interface performance versus and the need to retain task results.
When managing up to 5000 systems, the following default values are recommended. Please note that these settings may not be ideal for every situation; rather, they serve as a starting point at which further adjustment might be necessary depending on desired performance characteristics.

### Table 1 - Recommended Thresholds for Large Number of Managed Systems

<table>
<thead>
<tr>
<th>Setting</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduled Tasks: Number of Results</td>
<td>3</td>
</tr>
<tr>
<td>Scheduled Tasks: Number of Days</td>
<td>7</td>
</tr>
<tr>
<td>Run Now Tasks: Number of Results</td>
<td>100</td>
</tr>
<tr>
<td>Run Now Tasks: Number of Hours</td>
<td>8</td>
</tr>
</tbody>
</table>

Remote Microsoft SQL Server (recommended)

When the HP SIM database is co-located with Microsoft SQL Server on the same system, performance degradation occurs (due to input/output). It takes approximately thirty-five percent longer to process the same number of systems when the database is located on a local server versus a properly configured Microsoft SQL Server running remotely. Throughout the rest of this document, assume that Microsoft SQL Server and HP SIM are located on remote servers, denoted as the management server, unless otherwise specified.

HP SIM Sizer tool

The HP SIM Sizer assists systems administrators in building highly available, high performance HP SIM deployments running on Microsoft Windows, Linux, or HP-UX operating systems. Based on input from the results of HP’s quality assurance and performance testing, the tool recommends server configurations based on projected management workloads, installed management applications, and number of console users. The tool also provides performance and configuration data designed to help systems administrators optimize the performance of existing HP SIM deployments. HP recommends that the Sizer tool be consulted when considering CMS hardware and database configurations.
Customizing HP SIM to your environment

Creating a hierarchy: One master CMS with several regions

Creating a hierarchy in an environment where systems can be broken down into regions makes sense to create several CMS systems: one master CMS and several regional systems. First, create the master CMS to collect the entire event that is important to the business: critical, major and all Proliant events. Second, don’t install any plug-in to the master CMS. Each plug-in must be installed based on the use to the specific region. For example, if Region A has a virtual environment and Region B does not, only Region A would have VMM plug-in installed. The following are examples of a master CMS and regionalized CMS:

Master CMS:
- List of all server and important systems
- For inventory, contract, and warranty
- Long poll cycles (daily)
- Daily data collections (off production hours, by region)
- No event handling except for it regional CMS’s

Regionalize CMS for:
- Better event handling - less events
- Use for active Plug-in – VMM, VPM, PMP, VCRM

Discovery related

IP range
By default, HP SIM pre-populates discovery’s IP range with the network subnets used by the management server. You must modify this to represent the correct range of systems that you need to manage in your group. The Inclusion and Exclusion ranges give you the ability to fine-tune the systems that you want to discover using IP range pinging.

After HP SIM is installed on a system and the Discovery IP range is established, the discovery process must run to discover the systems on your network. For an empty database, the maximum time needed to complete an IP range scan for initial system detection is \((\text{Number Of IPAddresses} \times (\text{Retries} + 1) \times \text{Timeouts})/16\) in seconds (hence, a fully populated network could take a much shorter time period). \(\text{Number Of IP Addresses}\) can be calculated from the range you indicate in the Inclusion Range, less the IP addresses in the Exclusion Range (if any). By default, \(\text{Number Of IP Addresses}\) is the 254 IP addresses for each subnet the management server is connect to, using a class “C” network mask scheme. The Automatic Discovery window shows the progress of discovery. System status polling and Data Collection are invoked when new systems are discovered. Polling and data collection, like discovery, have retries and timeouts associated with them after they are invoked. HP recommends that the default tasks in HP SIM be customized to represent the network of systems being managed. For servers that require constant vigilance (like Microsoft Exchange servers), create a Hardware Status Polling task (based on a collection containing the Microsoft Exchange server systems) to run every five minutes. For file and print servers, create a separate Hardware Status Polling task to run once every hour. For desktops and portables, either change the polling frequency to once a day or week, or turn off Hardware Status Polling for these types of systems. This can be edited and scheduled by going to Task & Logs->View All Schedule Task as shown below in Figure 2.
Suggested polling intervals for the two default Hardware Status Polling task intervals should be configured as follows:

**Table 2 - Suggested Polling Intervals for Hardware Status Polling**

<table>
<thead>
<tr>
<th>Number of systems managed</th>
<th>Hardware Status Polling for Servers</th>
<th>Hardware Status Polling for Non Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 500</td>
<td>Use default of 5 minutes</td>
<td>Use default of 30 minutes</td>
</tr>
<tr>
<td>501 to 2000</td>
<td>Change to 30 minutes</td>
<td>Change to 2 hours</td>
</tr>
<tr>
<td>2001 to 5000</td>
<td>Change to 1 hour</td>
<td>Change to 4 hours or greater</td>
</tr>
</tbody>
</table>

**Note:** Do not let status polling tasks overlap. Time the Server Polling task completion time by making sure the interval is longer than the completion time.

**Changing Timeouts and Retries**

HP SIM utilizes Ping, TCP, and HTTP, WBEM and SSH protocols to determine if a system is on the network. During the communication phase of discovery, several items can be tailored to conform to your network environment. The first of these is Ping and SNMP Retries and Timeouts.
HP SIM is tailored, out-of-the-box, for a LAN configuration. In many companies, the network has both LAN and wide area network (WAN) components. To communicate over a WAN, network latency must be taken into account.

You must first customize Network Timeouts. If systems are not being discovered through pinging, or information is not being collected for a specific system over a WAN using SNMP, you must increase the Ping Timeouts setting.

To customize HP SIM to your environment, begin by investigating how other applications are tailored for communicating over your network. You can try increasing the Timeouts by a factor of 20 percent. For example, by default SNMP Timeouts are set to five seconds. First increase the value to \((5 \times 1.2)\) six seconds. If this still does not enable you to discover a particular system, increase it to \((6 \times 1.2)\) eight seconds.

**Note:** Setting timeouts greater than 10 to 20 seconds can cause the various polling tasks in HP SIM to take much longer to complete. In a mixed system latency environment, it is best to adjust timeouts and retries for the slower-responding systems using groups (such as LAN/WAN) and not through the default settings on the discovery page.

After a system is discovered, you can view a collection that contains that system and drill down to the **System Page**. From this window, you can further customize SNMP timeouts for that specific system. After the value is changed for that system, all tasks utilize the new timeout (or retry) when communicating with that particular system. Should your network have latency (WAN) problems, the best communication item to fine-tune is timeouts. If you are experiencing packet loss with low latency networks, the ideal item to customize is timeout retries (keep the current timeout values). HP suggests the minimum number of retries be set to one, using zero retries can result in packet loss and false notification of unreachable systems.

HP SIM is designed so that if a request is sent out and the reply is received after the timeout period, then the reply is ignored. Therefore, you must tailor timeouts before retries. Retries should only be incremented in single values. For example, if the default ping retries value of 1 is not adequate for your system, the next attempt should be 2.

**System and event management**

**Customizing the query system**

The collections feature of HP SIM enables you to generate information about a subset of systems, or a group. You can then perform tasks on all of the systems in that group with one collection. Several default collections enable you to see collected information from the database.

Two default queries, **All Systems** and **All Events**, produce a listing of all systems that have been discovered by HP SIM, and a list of all events that have been received. Separating events by what is important to your business needs is essential to maintain the performance of HP SIM. When managing a large number of systems (greater than 1,000), you can segment the systems according to importance, location, type, operating system, and more. Leverage the customize collections or Advanced Search features to create different collections to suit your needs. For example, fine tuning the events to collect selected ProLiant events instead of all events:
1. Select **Advance Search** and search for specific event types with a critical and major severity type.

![Figure 4: Collection for Critical and Major events for specific ProLiant events](image)

When viewing collections of up to 5000 items, HP SIM can seem slow depending on the system resources that are available on both the host and the client. This is because of the amount of data being returned from the database. Segmenting collections by responsibility or area of expertise (like e-mail servers, database servers, file servers, print servers, task servers, and Line of Business servers) provides greater functionality for viewing specific information while improving responsiveness. This also enables you to set up tasks specific to those particular systems. In a normal network environment, you can potentially receive a large number of SNMP events (Traps). Creating a number of smaller collections allows the query system to function more responsively and allows greater segmentation for executing a specific task when events occur.

**Note:** The Discovered System event is generated so that actions can be taken when new systems are added or removed from the network.

2. For every system that is discovered in HP SIM, an event is created that cannot be deleted (the Discovered System event) until the system is deleted. If you are managing 1,500 systems, this means that 1,500 events are automatically generated before any SNMP traps are received. These Discovered System events can be cleared so that they can be easily filtered out. This can be performed automatically by creating a collection to filter all events older than 14 days, then creating a task to delete events from that collection in a week. The example below filters all Informational Events older than 14 days.
3. After you have created and saved your collection, apply the collection to the delete event task by selecting **Options->Delete Events**. We saved the previous collection as **Information Events Older Than 14 days** under **Events by Time**.

4. To schedule the task, click **Schedule**.
5. Name the task in the **Task name** field. For instance, the task below is named **Delete Informational Events Older than 14 days**. We scheduled the task to run periodically and refined the schedule to run every 14 days at 6:00 am. All of these options are configurable to your business needs.

**Customizing the task system**

The task system of HP SIM enables you to execute a specific task on a group of objects (whether they are systems, events, or a combination of both). When collections are segmented in large network environments, you have greater control and an improved ability to customize HP SIM according to your needs.
Initially, HP SIM is configured to execute all of the various polling tasks according to a defined default schedule. These tasks are tied to queries in the System Default area. Most of these collections are configured for the **All Systems** collection criteria. In a large network, this arrangement might not be practical, nor would it be likely that you would want to poll all of your desktops or all of your servers on the same schedule.

By segmenting the collections by areas of responsibility or specific functions of systems (Exchange servers, clusters, and so on), you can configure HP SIM to poll those systems appropriately. For example, you can poll your most important Exchange or Microsoft SQL servers once a minute, and all other servers in your environment once every thirty minutes. You might only want to poll desktops and portables once per day or even once per week (utilizing the DMI Status Polling task).

To segment your tasks, change the collection that the default tasks are tied. You can edit these tasks and select a smaller collection set (like **Systems By Name** or **All Systems Within X Range**). As a starting point, you can edit the collection criteria for the system default collections so that you are polling a reduced set of systems instead of being tied to the **All Systems** collection criteria. At this point, you must create new tasks (Hardware Status Polling, Data Collection, Software Status Polling, and so on) and tie them to other collections that cover a different set of systems.

By creating several like tasks (Hardware Status Polling or Data Collection) against several different collections, you can manage your network exactly the way you need to without inducing any unnecessary traffic.

**Note:** The task system is a serialized system for like tasks. This means that if you create two data collection tasks, the first task executes and run to completion. The second task, if it was scheduled to execute in the same time period, remains queued until the first task has completed. This means that two data collection tasks do not run at the same time. If you have two dissimilar tasks, such as Hardware Status Polling and Data collection, then they execute simultaneously.

---

**Tuning Microsoft Windows Server 2003**

**Using Performance Monitor to optimize Microsoft Windows 2003**

The Microsoft Windows Performance Monitor that was used for most tuning tests can be found under **Start->Programs->Administrative Tools->Performance**. Use caution with the Performance Monitor because there are many counters and all of a Microsoft Windows 2003 server’s resources are interrelated.

HP recommends that you turn on Performance Monitor and log all of the objects and counters so that you are not missing one you want later. When setting up the sampling times, remember that performance logs grow large over time. To avoid being overwhelmed with data, look at a few key counters, then when they show problems, dig a bit deeper.

At a minimum, monitor the following objects and counters:

- **Memory Object**
  - Memory: Available bytes
  - Memory: Pages per second
- **Logical Disk**
  - Percent Disk Time
  - Dedicated disk or disk drive that houses the `pagefile.sys`
- **Logical Disk Object**
  - Percent Disk Time
  - Disk Queue Length
  - Disk sec/transfer
• Network Interface Object
  – Output Queue Length
  – Bytes Total per second
• System Interface Object (for CPU monitoring)
  – Percent Total Processor Time
  – Queue Length

In general, if an application is running at over 70 percent CPU utilization and a queue is forming, dig deeper to determine which system resource is holding back the performance of your Microsoft Windows 2003 system.

Configuring the logging feature
1. Start the Performance Monitor (in the Administrative Tools program group).
2. Click the logging toolbar button (the drum).
3. Click +, or select Edit and add to the Chart menu item.
4. Enter the name of the computer you want to monitor.
5. Select the object (HTTP Server, Gopher Server, and so on) from the Object list-box. Readings from all counters in selected objects are recorded.
6. Click OK.
7. Select Chart->Options menu and enter the fully qualified path for the log file you want to create.
8. Click Start Log.

Examining the log data
To examine the log data:
1. Start the Performance Monitor (in the Administrative Tools program group).
2. Select Options->Data from the menu.
3. Select the Data values displayed from the Log File option.
4. Enter the fully qualified path for the log file.
5. Click OK.
6. Click Edit, Add, or + to select the values to replay.
7. Select the object (HTTP Server, Gopher Server, and so on) from the Object list-box.
8. Select the counter to monitor from the Counter list-box.
9. Click Add.
10. Repeat as necessary, and then click Done.
11. Click the Edit->Time Window menu option to select when you want to view the specified counter values.

Performance Analysis of Logs (PAL) tool
When performing performance analysis and tuning activities, it might be helpful to employ the PAL tool. PAL can analyze Windows Performance Monitor log files utilizing common threshold values, and present an HTML report, which charts the performance counters and displays alerts when those thresholds are exceeded. PAL is not itself a replacement for methods of performance analysis, but it can provide meaningful information quickly and potentially save time when focusing on performance optimization. The tool can be found at the following URL: http://www.codeplex.com/PAL. It is open source and is released under the Microsoft® Public License (MS-PL).
File System Cache Settings

Windows Server 2003 is configured to perform as a file server by default. This may not be appropriate for memory or network intensive programs. In order to adjust this default behavior, it is necessary to adjust the file system cache settings. To reach the settings dialog, click Start->Control Panel->Network Connections, then right-click Local Area Connection and click Properties. On the dialog that comes up, double-click File and Printer Sharing for Microsoft Networks to bring up the Server Optimization window. This dialog box offers five options:

- Maximize data throughput for file sharing (default)

For most multi-user environments, two common options for tuning the memory strategy are: **Maximize Throughput for File Sharing** and **Maximize Throughput for Network Applications**. When you select Maximize Throughput for File Sharing, the operating system uses all available memory for the file system cache (dynamic disk buffer allocation). Use this option when you use the server as a file server only. Allocating all memory for file system buffers generally enhances disk and network input/output performance. By providing more RAM for disk buffers, you increase the probability that the server will complete input/output requests in the faster RAM cache instead of the slower file system on the disk. However, this relationship is not good if you are running any other applications on the server. When you start a client or server application, such as Microsoft SQL Server or another memory-intensive application, the management server might begin to page (swap information between RAM and the disk) excessively.

- Maximize data throughput for network applications (suggested)

The second strategy for tuning the memory of Microsoft Windows Server 2003 is to maximize throughput for network applications. When you select this option, the server allocates less RAM for the file system cache so that running applications can have access to more RAM. With this option, you can tune applications, such as Microsoft SQL Server and Microsoft Exchange, to use specified amounts of RAM for areas such as buffers for disk input/output and database cache.

Knowing what is running on your system is particularly important when maximizing throughput for network applications. If you allocate too much memory for each application in a multi-application environment, excessive paging can turn into thrashing, which leads to very slow system response. Thrashing is the state when all active processes and file system cache requests become so great that they overwhelm the system’s RAM. When this condition occurs, requests for RAM create hard page faults at a large rate and the virtual memory manager begins stealing pages from one process to fulfill the request of another process. On a busy system, the increased workload of the virtual memory manager consumes more disk resources as the use of the Pagefile.sys increases. The management server then wastes CPU cycles on memory management functions instead of servicing productive processes. Thrashing can quickly consume an inordinate amount of system resources and typically increases user response times considerably.

You can accept some paging or even continuous paging if the response times to the end user are reasonable. You can use tools such as third-party remote terminal emulation and third-party remote workstation emulation to test system loading in conditions that emulate your environment. This testing measures factors such as user response times and overall system throughput.
Virtual memory paging

A process’s working set is where its current code and data are physically located in RAM. When sufficient RAM is available and a process requests information that is not currently in RAM, the virtual memory manager in Microsoft Windows Server 2003 leaves the process’ current working set in RAM and retrieves the additional information from disk. This action is a soft page fault.

When RAM resources become scarce (either from the working sets of other processes or because of the RAM that the file system cache is using), the virtual memory manager moves older pages from RAM to the paging file system on the disk drive. This process is a hard page fault.

When a hard page fault occurs, the virtual memory manager has, in essence, stolen some RAM. In other words, it has trimmed the working set of another process to fulfill a request from the process that is currently running. Occasionally, paging which results from hard page faults is acceptable, but if this type of paging occurs excessively, system resources become unbalanced and a memory bottleneck forms.

To determine whether your system is paging, use the Performance Monitor to observe the relationship of the metrics. If the pages per second counter are high (greater than 50) and growing compared with your baseline, then memory is getting bottlenecked. If the available bytes counter is also decreasing and the disk drive(s) that house the Pagefile.sys files are busy (marked by an increase in Percent Disk Time, Disk bytes per second, and Average Disk Queue Length), you have a memory bottleneck.

You can improve the system’s memory and paging performance when loads are heavy by spreading the paging file across two or more disks. This modification improves the overall paging file read/write rates because more disks are available to process the paging file workload.

During installation, the operating system creates one Pagefile.sys file on the root (C:) drive. To spread the load, review the disk metrics you gathered with Performance Monitor and select two disks that are under the lightest load. Select Control Panel->System, and then click the Advanced tab, and the Performance button. Click Advanced->Virtual Memory and create two new paging files, one on each disk. After the new paging file systems are in place, remove the default Pagefile.sys on the root disk.

As a guide to determining the Pagefile.sys sizes on the new disks, use Performance Monitor to monitor the percent usage and percent peak usage counters of the paging file. Usually, you create pagefiles that are the same size on both disks, with initial sizes of at least the value shown for percent usage and a maximum of at least the value for percent peak usage.

Sizing the Pagefile.sys correctly ensures that the server operating system does not waste cycles creating larger Pagefile.sys files. If possible, dedicate two disk drives to the task of containing the paging file systems. This approach guarantees that no other application or process contend with the server operating system when the system needs the paging file system.

If the system begins to page at an unacceptable degree, use Performance Monitor and Task Manager to isolate the applications or processes that are using excessive amounts of memory and reduce the amount of memory allocated to them (if possible). If the application source code is available, you can work with the application developers to improve overall memory performance. When all tuning efforts fail to improve user response times related to a lack of memory, install more memory into the system or distribute memory-intensive applications to the appropriate number of additional management servers.
Tuning disk input/output

Disk tuning—RAID 5 performance

Ensure that there are no other bottlenecks (CPU, NIC, or memory) before adding more disks. Typically the SCSI or fibre channel is not saturated, but rather there are not enough disks configured. Ensure that you place the log files, indexes, and database files on different array sets.

The RAID stripe size is written to each system in the array, or to the group of systems in the RAID set. In other words, if you have a stripe size of 128 KB and four disks in a RAID 5 array, the stripe size is divided across all the disks in the array. Each chunk is written to each drive in the array. So if you have three systems in a RAID 5 array, three chunks are written. Logically though, only two of the three drives are receiving your data. The other drive is receiving the XOR parity data, which is rotated among the disks.

For RAID 5, set the low-level raid format chunk size so that, when multiplied by the stripe width (number of drives in the array minus one), the product is equal to the typical input/output size you use from the server operating system’s perspective.

For example, if you determine your input/output writes are occurring 8KB at a time using Performance Monitor and you have a three-disk RAID 5 set, chunk size must equal to 8KB per two drives, or a 4KB chunk. If this is not possible, set the chunk size to the next larger size available. Now, each 8KB write fills an entire stripe set (user data) and the parity data is written on the same stripe set on the last disk. Keep in mind that this parity information rotates around the disks in the array.

Set the relationship between the RAID stripe size and the formatted sector/cluster size so that the operating system input/output writes to the cluster size with the proper chunk size. Set the allocation unit size of the NTFS partition to 8192.

Tuning a single disk

To optimize the performance of single, large hard disks in heavy throughput server environments, RAID is the solution of choice, with throughput being distributed over multiple drives and, more particularly, by the application of large amounts (32 MB or greater) of write back cache on the hardware RAID controller. However, many people are moving to more fault-tolerant configurations with full management server mirroring and full fail-over for a more dynamic, high availability solution. This solution deals with any type of management server failure, not merely a hard disk failure. In this environment it is unnecessary and costly to mirror complete RAID systems.

Additional tips

- Use only one disk partition per physical disk (lowers head movement).
- Defragment the disk on a regular basis.
- Use the proper allocation size (match it to your workload).

Configuring the network

Generally, no specific tuning is required to get a NIC in Microsoft Windows Server 2003 to run at full speed, but there are some items to consider:

- Do not use auto detect. Set your network system, clients, and management server NICs to the exact settings you want. In many cases auto detect provides less than optimal performance. Foregoing auto detect also ensures that your network systems are working properly at the speed you desire. If they are not, connectivity problems are likely to occur.
• Ensure that nothing else is a bottleneck on the management server, client, or network. If your network is not the limiting factor in your setup, then a fast NIC will probably not help. Ensure that the application, CPUs, memory, and disks are not limiting your network performance.
• Try a test case of your management server and clients on their own isolated LAN to see what kind of throughput can be achieved.

Microsoft Windows Server 2003 performance tuning

Even though most of the above performances tuning options are equally applicable to Microsoft Windows 2003, there are additional performance aspects to consider when optimizing your Microsoft Windows server.

CPU configuration
Changing the CPU speed, the CPU cache size, or the number of CPUs in a Web server can improve performance significantly. Using a faster CPU or one with a larger cache always improves the performance of a Web server with a CPU bottleneck. For example, on Web servers that respond to a significant number of dynamic requests and use encryption, increasing the number of CPUs, CPU speed, or CPU cache size can be a very effective way to increase performance. Even though HP SIM is typically not CPU bound, other system components, such as the operating system, Microsoft SQL Server, or other applications, all benefit from the addition of another processor, improving overall system performance. However, adding CPU resources to a CPU-limited Web server sometimes does not improve performance. If you do not see much performance improvement for a highly dynamic site when adding CPU resources, the problem might be with the design of the dynamic content or Web application. For static workloads, the CPU is unlikely to be the bottleneck; the usual culprit is the network.

Memory
Web server performance is very sensitive to the amount of memory in a server. For example, Microsoft Windows 2003 is able to cache high demand files in physical memory. By caching static files in memory, the server is able to process requests more efficiently since disk input/output is eliminated (except for logging). For the best performance, a Web server should have enough memory to hold all static files. If this is not possible, the disk subsystem becomes more critical.

HP SIM memory requirements
There are three main processes associated with HP SIM:

mxdomainmgr

mxdomainmgr is the main process that runs HP SIM. It is started by the primary HP SIM service and is responsible for initializing and managing the core HP SIM web application, partner code, and so on. This service runs the web server as well as the bulk of the code that is HP SIM. It is not unusual to observe mxdomainmgr using 750MB or more of memory.

Note that mxdomainmgr runs within a Java Virtual Machine process, which is restricted to the maximum amount of memory available to any single process in Windows Server 2003, namely, approximately 1.6GB. This restriction is because 32-bit Windows does not allow any single process to allocate more than 2GB of memory. The JVM heap must be allocated in a single contiguous block of memory, and since Windows breaks up contiguous memory with additional libraries and so on, the JVM never utilizes more than 2GB of memory. Keep this in mind when considering system RAM upgrades; though SQL Server benefits (as will other concurrent processes), the JVM does not benefit.
mxdtf

**mxdtf** is launched by the service. The main purpose of this process is to handle the audit logging and SSH command line execution for MSA, SSA, and command launch tools. Memory for this process is not nearly as high, usually less than 100 MB.

mxinventory

**mxinventory** is started as needed for WBEM and DMI data collection. When idle for longer periods it shuts itself down for a period of time. Note that for there might be two instances of this process started, one for WBEM the other for DMI.

System Page file

If your configuration has two or more hard disk drives, HP recommends that you move the system page file to a different drive than the one where the Microsoft Windows 2003 operating system is installed. Splitting the page file across multiple disks also improves paging performance.

Disk subsystem

The disk subsystem has very little effect on Web server performance for static workloads that can be cached in memory. For Web servers that have less memory or that use disk input/output to generate dynamic content, a RAID subsystem with at least four disks can improve performance substantially over a configuration that uses single disks connected to a SCSI controller.

Network subsystem

The total bandwidth available to the Web server spread across all of the network adapters in the server sets the limit for the number of bits that a server can send or receive. For Web servers on the Internet, this is probably the most common bottleneck. Correspondingly, the network bandwidth limits the request rate a server can handle. To determine the performance capabilities of a Web server, you must make sure that there is enough network bandwidth so that the tests reach the server’s peak performance. For instance, if a powerful server only has 100 megabits of bandwidth, this could be the factor that prevents the server from performing better. If the CPU is the bottleneck and there is still some network bandwidth available, using a network adapter that supports the offloading capabilities in Microsoft Windows 2003 can free up CPU cycles to process more requests.

Tuning Microsoft SQL Server

The performance of Microsoft SQL Server depends on the interaction of many factors, ranging from the hardware that the system is running to the application coding techniques used. Microsoft SQL Server provides methods and tools to tune Microsoft SQL Server for optimum performance. Keep these principles in mind when you are tuning Microsoft SQL Server:

- **Plan for disk input/output subsystem performance.**
  The placement of the paging file, transaction logs, and system databases can have a big impact on system performance and recoverability. Place log files and database files on different physical drives. Microsoft SQL Server performance can be enhanced by locating log files and database files on separate physical drives. HP SIM places these types of files on the same drive by default. During installation of HP SIM, designate either the log files or the database files to be installed on a separate physical drive. The physical disk subsystem must provide a database server with sufficient input/output processing power for the database server to run without disk queuing, which indicates poor performance.

- **RAM is a limited resource**
  An integral feature of the database server environment is the management of RAM buffer cache. Access to data in RAM cache is much faster than access to the same information from disk, but
RAM is a limited resource. If database input/output can be reduced to the minimum required set of data and index pages, these pages stay in RAM longer. Too much unnecessary data and index information flowing into buffer cache quickly pushes out valuable pages. The focus of performance tuning is to reduce input/output so that buffer cache is best utilized. Memory is a cheap resource always in demand. No amount of tuning can make up for a lack of RAM.

- Let Microsoft SQL Server do most of the tuning.
  Microsoft SQL Server has been dramatically enhanced to create an auto-configuring and self-tuning database server. Take advantage of the auto-tuning settings available. These settings help Microsoft SQL Server run at peak performance even as user load and queries change over time.

- Use the Create Index Wizard feature to create indexes for queries you add.
  A key factor in maintaining minimum input/output for all database queries is to ensure that good indexes are created and maintained. This wizard guides you through a series of steps to create indexes, prompting you for input along the way.

- Take advantage of Microsoft SQL Server Profiler and Index Tuning Wizard.
  Microsoft SQL Server Profiler can be used to monitor and log a Microsoft SQL Server workload, which can then be submitted to the Index Tuning Wizard to tune indexes for better performance. Regular use of Microsoft SQL Server Profiler and the Index Tuning Wizard helps you optimize the indexes, allowing Microsoft SQL Server to perform well with changing query workloads.

- Monitor the management server with Windows Performance Monitor
  Regularly monitoring the management server, especially its disk utilization, reveals important clues for troubleshooting performance problems. Use Microsoft SQL Performance Monitor before you have a problem to establish a performance baseline for your management server. Microsoft SQL Server provides counters that the Microsoft Performance Monitor can track. Counters that can help you monitor disk activity include PhysicalDisk (Percentage Disk Time), PhysicalDisk (Average Disk Queue Length), Microsoft SQL Server (Buffer Manager Page Reads/second), and Microsoft SQL Server (Buffer Manager Page Writes per second). These disk input/output counters are turned on by default in Windows Server 2003 and are available in the performance monitor tool.

Optimizing Microsoft SQL Server performance with Microsoft SQL Server components

Worker threads
Microsoft SQL Server maintains a pool of Microsoft Windows operating system threads to service batches of Microsoft SQL Server commands submitted to the database server. The total of these threads (called worker threads) available to service all incoming command batches is dictated by the setting for the `sp_configure` option max worker threads. If the number of connections actively submitting batches is greater than the number specified for maximum worker threads, the worker threads are shared among connections actively submitting batches. The default setting (255) works well for many installations.

Worker threads write out most of the dirty 8-KB pages from the Microsoft SQL Server buffer cache. Input/output operations are scheduled by worker threads asynchronously for maximum performance.

Lazy Writer
Microsoft SQL Server Lazy Writer helps produce free buffers, which are 8-KB data cache pages that contain no data. As Lazy Writer flushes each 8-KB cache buffer out to disk, it initializes the cache page identity so that other data can be written into the free buffer. Lazy Writer produces free buffers during periods of low disk input/output, so that disk input/output resources are readily available for use and so that there is a minimal impact on other Microsoft SQL Server operations.
Microsoft SQL Server automatically configures and manages the level of free buffers. Monitor the Microsoft SQL Server: Buffer Manager, Free Buffers object to ensure that the free buffer level remains steady. Lazy Writer maintains the level of free buffers to keep up with the user demand for free buffers. The Microsoft SQL Server: Buffer Manager, Free Buffers object should not drop to zero as this indicates that there were times the user load demanded a higher level of free buffers than the Microsoft SQL Server Lazy Writer was able to provide.

If the Lazy Writer cannot keep the free buffer steady, or at least above zero, it might mean the disk subsystem cannot provide Lazy Writer with the disk input/output performance that it needs to maintain the free buffer level. You can compare drops in free buffer level to any disk queuing to confirm that there is a disk subsystem problem.

One solution to the disk queuing problem is to add more physical disk drives to the database server disk subsystem to provide more disk input/output processing power. The Microsoft SQL Server: Buffer Manager, Lazy Writes/sec object indicates the number of 8-KB pages written to disk by lazy writer.

Monitor the current level of disk queuing in Performance Monitor by looking at the counters for (logical or physical) Disk: Average Disk Queue or Current Disk Queue. The disk queue needs to be at a level less than 2 for each physical drive associated with any Microsoft SQL Server activity. For database servers that employ hardware RAID controllers and disk arrays, divide the number reported by disk counters (logical or physical) by the number of actual hard disk drives associated with that logical drive letter or physical hard disk drive number reported by the Disk Administrator program. Microsoft Windows and Microsoft SQL Server are unaware of the actual number of physical hard disk drives attached to a RAID controller. You should know the number of drives associated with RAID array controller to interpret the Performance Monitor reports on disk queue numbers.

You should monitor disk queuing on hard disk drives associated with Microsoft SQL Server data files to see if Microsoft SQL Server is sending down more disk input/output requests than the disk(s) can handle. If this is the case, then more disk input/output capacity must be added to the disk subsystem to handle the load.

Log manager
Like other major Relational Database Management System (RDBMS) products, Microsoft SQL Server ensures that all write activity (inserts, updates, and deletes) performed on the database is not lost if something interrupts the Microsoft SQL Server online status (power failure, disk drive failure, fire in the data center, and so on). The Microsoft SQL Server logging process helps guarantee recoverability. Before any implicit (single Transact-SQL query) or explicit (transaction that issues Begin Transaction, Commit, or Rollback statements) transactions can be completed, the Microsoft SQL Server log manager must receive a signal from the disk subsystem telling it that all associated data changes have been written successfully to the associated log file. This rule guarantees the transaction log can be read and reapplied in Microsoft SQL Server when the management server is restarted after an abrupt shut down during which the transactions written into the data cache are not yet flushed to the data files. Flushing data buffers are Checkpoint or Lazy Writer responsibility. Reading the transaction log and applying the transactions to Microsoft SQL Server after management server stoppage is referred to as recovery.

Disks containing Microsoft SQL Server log files must have sufficient disk input/output handling capacity for the anticipated transaction load because Microsoft SQL Server must wait for the disk subsystem to complete input/output to Microsoft SQL Server log files as each transaction is completed. The method for monitoring disk queuing is different for Microsoft SQL Server log files than it is for Microsoft SQL Server database files. You can use the Performance Monitor counters Microsoft SQL Server: Databases database instance: Log Flush Waits Times and Microsoft SQL Server: Databases database instance: Log Flush Waits/sec to view log writer requests waiting on the disk subsystem for completion.
For highest performance, you can use a caching controller for Microsoft SQL Server log files if the controller guarantees that data entrusted to it is written to disk eventually, even if the power fails.

**Moving tempdb location**

If more physical drives can be added to the system, HP recommends that tempdb locations be moved to different physical drives to avoid read or write delays associated with operating system and SQL Server background activity.

By default, tempdb is placed in the same physical directory as the SQL Server binaries. To move tempdb to another physical location, enter the following commands in SQL Server Management Studio or an SQL Server prompt (note that <new location> represents the desired location of tempdb on the new physical drive):

```sql
use master
go
Alter database tempdb modify file (name = tempdev, filename = '<new location>\tempdb.mdf')
go
alter database tempdb modify file (name = templog, filename = '<new location>\templog.ldf')
go
```

**SQL Server minimum and maximum memory**

When tuning SQL Server, it might be important to adjust the minimum and maximum amount of memory that can be consumed. If HP SIM and SQL Server are running on the same machine, it becomes even more critical as the operating system, HP SIM, SQL Server, and other applications compete for RAM.

It might be difficult to know how much memory to allocate to SQL Server, especially as system configuration becomes more complicated as noted above. On a server with 4GB of RAM that is also running HP SIM, HP recommends that 50% of memory be allocated to SQL Server maximum value. As the system runs, performance statistics should be gathered by perfmon over an extended period of time. These statistics can be analyzed to determine if it is necessary to adjust the minimum and maximum values further.

The settings can be configured as follows (note that values are in megabytes):

```sql
sp_configure 'min server memory', 1024
RECONFIGURE
GO
sp_configure 'max server memory', 2048
RECONFIGURE
GO
```

**PAE and AWE**

Physical Address Extension (PAE) refers to the feature of the X86 and X86-64 processors that allows them to address more than 4GB of memory. Previously, the X86 32-bit processors were only able to address 232 (4GB) of memory, however the addition of more address lines increased that amount to 236 (64GB). All of this additional address space is still unusable without operating system support, and since the 32-bit virtual addressing space is unchanged, must use page tables to map the 4GB virtual space to the 64GB physical space. Applications are still limited to the traditional 4GB virtual space as well.
Advanced Windowing Extensions (AWE) is a method that the Windows operating system can make more than 4GB available to applications through system calls. To take advantage of AWE memory, you must configure both Microsoft Windows 2003 and Microsoft SQL Server.

**PAE**

PAE is the added ability of the IA32 processor to address more than 4 GB of physical memory. To enable PAE, add the `/PAE` switch in the `boot.ini` file. In most circumstances, the PAE flag is enabled by default if you install Windows Server 2003 on a system with 4GB of RAM or more. When upgrading system RAM, please note if you are exceeding 4GB and verify that the `boot.ini` PAE flag is set.

**Note:** This applies specifically to Windows Server 2003 Enterprise Edition. Windows Server 2003 Standard Edition cannot address above 4GB of RAM.

In Windows Server 2003 Enterprise, PAE is automatically enabled only if the server is using hot-add memory devices. In this case, you do not have to use the `/PAE` switch on a system that is configured to use hot-add memory systems. In all other cases, you must use the `/PAE` switch in the `boot.ini` file to take advantage of memory over 4GB.

The following table details the maximum physical memory that a Windows version can recognize, with the PAE switch enabled:

<table>
<thead>
<tr>
<th>Version</th>
<th>Maximum Physical Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 2000 Advanced Server</td>
<td>8 GB</td>
</tr>
<tr>
<td>Windows 2000 Datacenter Server</td>
<td>32 GB</td>
</tr>
<tr>
<td>Windows XP</td>
<td>4 GB</td>
</tr>
<tr>
<td>Windows Server 2003 Enterprise Edition</td>
<td>32 GB</td>
</tr>
<tr>
<td>Windows Server 2003 R2 (or SP1) Enterprise Edition</td>
<td>64 GB</td>
</tr>
<tr>
<td>Windows Server 2003 Datacenter Edition</td>
<td>64 GB</td>
</tr>
<tr>
<td>Windows Vista</td>
<td>4 GB</td>
</tr>
<tr>
<td>Windows Server 2008 Enterprise or Datacenter Edition</td>
<td>64 GB</td>
</tr>
<tr>
<td>Windows Server 2008 other editions</td>
<td>4 GB</td>
</tr>
</tbody>
</table>

Typically, a process running under Windows Server 2003 can access up to 2 GB of memory address space (assuming the `/3GB` switch was not used) with some of the memory being physical memory and some being virtual memory. The more programs (and, therefore, more processes) that run, the more memory you commit up to the full 2 GB of address space.

When this situation occurs, the paging process increases dramatically and performance might be negatively impacted. The Windows Server 2003 memory managers use PAE to provide more physical memory to a program. This reduces the need to swap the memory of the page file and results in increased performance. The program itself is not aware of the actual memory size. All the memory management and allocation of the PAE memory is handled by the memory manager independently of the programs that run.

The preceding information is valid for programs that run when the `/3GB` switch is used. A program that requests 3GB of memory is more likely to have more of its memory remain in physical memory rather than be paged out. This increases the performance of programs that are capable of using the `/3GB` switch. The exception is when the `/3GB` switch is used in conjunction with the `/PAE` switch. In this case, the operating system does not use any memory in excess of 16GB. This behavior is caused by kernel virtual memory space considerations. Thus, if the system restarts with the `/3GB` entry in the
The boot.ini file, and the system has more than 16GB of physical memory, the additional physical random access memory (RAM) is not used by the operating system. Restarting the computer without the /3GB switch enables the use of all the physical memory. For example, if you are running a system that has 4 GB of RAM installed and you then add 4 GB of additional RAM, Windows may recognize only 4 GB of physical memory or possibly 6 GB instead of the full 8 GB.

Systems with the PAE feature can support up to 64 GB of RAM (2^36).

### Configuring Microsoft Windows Server 2003 to use PAE

How you configure AWE memory support depends on how much RAM your server has. To configure Microsoft Windows, you must enter one of the following switches in the boot line of the boot.ini file, and reboot the server:

<table>
<thead>
<tr>
<th>System Memory Size</th>
<th>Boot.ini switch settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 GB</td>
<td>/3GB (AWE support is not used)</td>
</tr>
<tr>
<td>8 GB</td>
<td>/3GB /PAE</td>
</tr>
<tr>
<td>16 GB</td>
<td>/3GB /PAE</td>
</tr>
<tr>
<td>More than 16 GB</td>
<td>/PAE</td>
</tr>
</tbody>
</table>

The /3GB switch is used to tell Microsoft Windows 2003 to make 3 GB out of the base 4 GB of RAM that Microsoft Windows 2003 supports natively available to user programs, reserving only 1 GB to the operating system. If you don’t specify this option, then Microsoft SQL Server can only use 2 GB of the first 4 GB of RAM in the server, essentially wasting 1 GB of RAM.

AWE memory technology is used only for the RAM that exceeds the base 4 GB of RAM, that’s why the /3GB switch is needed to use as much of the RAM in your server as possible. If your server has 16 GB or less of RAM, then using the /3GB switch is important. But if your server has more than 16 GB of RAM, then you must not use the /3GB switch because the 1 GB of additional RAM provided by adding the /3GB switch is needed by the operating system in order to take advantage of all of the extra AWE memory. In other words, the operating system needs 2 GB or RAM itself to manage the AWE memory if your server has more than 16 GB of RAM. If 16 GB or less of RAM is in a server, then the operating system only needs 1 GB of RAM, allowing the other 1 GB of RAM for use by Microsoft SQL Server.

The following is an example of a boot.ini file where the PAE switch has been added:

```plaintext
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINDOWS="Windows Server 2003, Enterprise" /fastdetect /PAE
```

If there is more than 16 GB of physical memory available on a computer, the operating system requires 2 GB of virtual address space for system purposes and therefore can support only a 2 GB user mode virtual address space. For the operating system to use the memory range above 16 GB, be sure that the /3gb parameter is not in the Boot.ini file. If it is, the operating system cannot use any physical memory above 16 GB.

### Configuring Microsoft Windows Server 2008 to use PAE

Microsoft Windows Server 2008 does not support setting boot options in the boot.ini file. Instead, the BCDEdit utility must be employed to configure PAE options. (Please note that the PAE option only applies to 32-bit Windows versions.) If the system is configured for hot-add memory devices with address ranges beyond 4 GB, Windows Server 2008 automatically configures itself with the PAE option enabled.
To enable PAE, enter the following command:

```
> bcdedit /set {current} pae default (or ForceEnable)
```


**Using AWE memory in Microsoft SQL Server 2005 to access >2GB memory**

Note that AWE is not needed and cannot be configured on 64-bit operating systems. Though the Use AWE option is available in Microsoft SQL Server 2005 64-bit edition, it is ignored. However, it is still recommended that this option be selected.

Because many Microsoft SQL Server databases are huge, greatly exceeding 2GB, more than 2GB of RAM is often needed in order to attain a 90% or greater buffer cache ratio. Because of this problem, Microsoft Windows Server 2003 and Microsoft SQL Server 2005 support a feature called Address Windowing Extensions (AWE). AWE is a way to allow Microsoft SQL Server 2003 to access more than 2GB of memory.

In most Microsoft SQL Servers, the biggest performance bottleneck is input/output. First, the purpose of a database is to store data, and data is stored on disk arrays, requiring input/output to retrieve the data from the disk array before it can be put into RAM for use. Second, disk access is the slowest component of any Microsoft SQL Server. Combined, these two factors contribute to this common bottleneck.

To help reduce input/output bottlenecks, Microsoft SQL Server includes a buffer cache, located in RAM, used to store the most recently accessed Microsoft SQL Server data. By caching data, Microsoft SQL Server does not need to access disk input/output as often, helping to boost overall performance of Microsoft SQL Server.

The greater the size of the buffer cache, the more data Microsoft SQL Server can store in RAM, and the less input/output access is required. In an ideal world, all of the data in a database would be stored in the buffer cache, greatly reducing disk input/output, and boosting Microsoft SQL Server’s performance. In some cases this is possible, but in most cases, it is not.

For ideal performance, the buffer cache hit ratio (the amount of data in RAM that is being reused) should exceed 90%. If it does, excessive disk input/output is generally avoided, helping performance. But if the buffer cache hit ratio is less than 90%, then disk input/output might become a serious bottleneck.

The easiest way to boost the buffer cache hit ratio in a Microsoft SQL Server is to allocate additional physical RAM to the Microsoft SQL Server. Assuming you have not changed the default memory configuration settings in Microsoft SQL Server it then automatically uses as much RAM as you add to your server.

Microsoft Windows 2003 is a 32-bit operating system. This means that by default, it can only support up to 4GB of memory (2GB for the operating system and 2GB for user applications). In a practical sense, this generally means that Microsoft SQL Server is limited to 2GB of RAM. In many cases, this is more than enough RAM for Microsoft SQL Server. Please refer to Configuring Microsoft Windows Server 2008 to use PAE.

**Microsoft SQL Server Express**

When installing HP SIM, you are given the option to install and use Microsoft SQL Server Express. There are important considerations to keep in mind when choosing this option, especially when planning performance tuning. SQL Server Express is limited to 1GB of RAM, one processor, a 4GB database size, and runs in Windows on Windows (WOW) mode on a 64-bit platform. Therefore, there might not be a significant benefit in upgrading system memory, CPU, or storage size when using SQL Server Express. In addition, if HP SIM is managing many systems, events, or performing
historical data collection, HP recommends that you do not use SQL Server Express, and instead elect to utilize a more capable version of SQL Server.

Configuring Microsoft SQL Server 2005 to use AWE memory
Even though Microsoft Windows 2005 knows about the AWE memory, Microsoft SQL Server 2005 cannot use it until it has been configured appropriately. This is a two-step process:

AWE Support must be enabled: To enable AWE memory on Microsoft SQL Server, enter the following commands:

```sql
sp_configure 'show advanced options', 1
RECONFIGURE
GO

sp_configure 'awe enabled', 1
RECONFIGURE
GO
```

After SQL Server restarts, the following message is written to the SQL Server error log: Address Windowing Extensions enabled.

Because the **awe enabled** option is an advanced option, you must first permit advanced options to be set by running the **show advanced options** option, as shown above.

**IMPORTANT**: To use AWE memory, you must run the Microsoft SQL Server 2005 mssqlserver service under a Microsoft Windows 2005 account that has been assigned the Microsoft Windows 2005 **lock pages in memory** correct. Microsoft SQL Server Setup automatically grants the mssqlserver service account permission to use the **Lock Page in Memory** option. If you start an instance of Microsoft SQL Server 2005 from the command prompt using sqlservr.exe, you must manually assign this right to the interactive user’s account using the Microsoft Windows Group Policy utility, or Microsoft SQL Server will be unable to use AWE memory when run from the command prompt.

**Note**: If you are clustering Microsoft SQL Servers that use AWE memory, it is important that all the servers in the cluster have the same amount of RAM and have their memory settings configured identically, otherwise a failover operation may not be successful.

If you have successfully enabled AWE memory, the message **Address Windowing Extension enabled** appears in the Microsoft SQL Server error log when Microsoft SQL Server is started.

Using AWE memory
After both Microsoft Windows 2003 and MS SQL Server 2005 are properly configured, you are ready to take advantage of the benefits of your added RAM. If you have just added RAM to a new Microsoft SQL Server, you must watch the buffer cache hit ratio to see the effect of the addition of more RAM. If you still do not get a 90% or more buffer cache hit ratio after adding the extra RAM, you might need to consider adding even more RAM, or redesigning your database so that it is more efficient.

Carefully monitor the paging on the server after first configuring AWE memory. As mentioned before, Microsoft SQL Server does not dynamically manage AWE memory. If you see that paging starts to become a problem, you might want to reduce the amount of RAM Microsoft SQL Server can access by reducing the Microsoft SQL Server **max server memory** setting. It might take some experimentation before you find the ideal setting for the amount of maximum memory Microsoft SQL Server should have. Ideally, paging should be virtually nonexistent on a well-tuned Microsoft SQL Server.
For more information