Configuration and Tuning of Sybase SQL Server 11 for SCO UnixWare 2.1 on Compaq Servers

White Paper

Prepared By
Database Engineering
Compaq Computer Corporation

June 1997

NOTICE

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state or local requirements. Compaq does not warrant products other than its own strictly as stated in COMPAQ product warranties.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Configuration and Tuning of Sybase SQL Server 11

for SCO UnixWare 2.1

on Compaq Servers

Second Edition (June 1997)

Document Number 541A/0697

Compaq Computer Corporation

Table of Contents

Configuration and Tuning of Sybase SQL Server 11 for SCO UnixWare 2.1 on Compaq Servers

Introduction

The purpose of this document is to share the knowledge acquired by Compaq Systems Engineers in the area of configuration and performance tuning of Sybase SQL Server 11 on UnixWare 2.1 on the Compaq Proliant family of servers. It is our desire to deliver the best technical information possible on a specific topic in a timely manner and in a highly useable format. Any comments, suggestions and feedback are always appreciated.

The information presented in this document is based on Sybase SQL Server 11 for UnixWare 2.1 and is a result of numerous performance tests executed within the context of an industry-standard TPC-C benchmark, internal benchmarking for hardware development and optimization, and analyzing customer-reported expectations, performance trends, and solutions.

Compaq is an active member of the Transaction Processing Performance Council, and publishes a number of benchmarks every year proving the superior performance and price-performance ratios of Compaq servers. The results of these benchmarks can be obtained directly from Compaq Computer Corporation or from the Transaction Processing Performance Council. The contact information is listed below.

Compaq Computer Corporation
Database Performance Engineering
P.O. Box 692000
Houston, TX 77269-2000
http://www.compaq.com

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311
http://www.tpc.org

Even though most of the testing that provided basis for this document was done in the area of online transaction processing, much of the information presented does apply to other environments, such as decision support and batch processing. We recommend that you always experiment before applying any changes to your production server.

Other publications covering these and related topics are listed below:

- *Configuring Compaq RAID Technology for Database Servers*, Compaq TechNote, P/N 184206-001
- Sybase SQL Server System Administration Guide
- Sybase SQL Server Performance and Tuning Guide
- Compaq Tech Communiqué 'COMPAQ INSIGHT Server Management'

Tuning Goals

In order to achieve the best performing system possible there are several factors which must be reviewed. These include optimization of the hardware, the Sybase SQL Server, the operating system and the application software. This paper will focus on the hardware, Sybase SQL Server and the OS. It is also important to tune the Sybase application to take advantage of the system. Due to the diversity of database applications, they are beyond the scope of this paper. This paper will discuss CPU scalability and tuning, disk controller optimization and I/O tuning, memory tuning, and network tuning. Also, specific Sybase and UnixWare configuration and tuning issues will be presented throughout the paper.

System Processor Planning

This section is provided to demonstrate the scalability of various processor configurations on the Compaq Proliant family of servers and to provide you with some performance information necessary to determine the best configuration for your environment.

CPU Scalability

The performance information presented in this section was gathered on three Compaq Proliant models, Proliant 4500 and Proliant 5000 and Proliant 6000, with different system processor configurations and different amounts of memory. All configurations had one aspect in common: all were CPU bound.

The tests run were all disk-intensive tests. Disk-intensive tests are designed to access the entire range of a database that is many times larger than the data cache of the server. The net effect is that only a very small portion of the database can fit into the data cache at any point in time, and a large amount of physical I/O is generated in addition to the transaction log I/O. Such an environment tests heavily the CPU, memory, and the disk subsystem.

The tests generated a large number of update-intensive transactions and utilized heavily the transaction log. No hardware or software options were disabled to sacrifice data integrity of the system.

Figure 1 shows scalability on the Proliant 4500 with Pentium®/100MHz and Pentium®/133MHz processors, using 1GB RAM. Figure 2 shows scalability on the Proliant 5000 with PentiumPro®/166MHz processors, using 1GB RAM and 2GB RAM in the server. The PentiumPro®/166MHz has a 512KB level 2 cache. Figure 3 shows the scalability on the Proliant 6000 with PentiumPro® 200MHz processors with 512K level 2 cache and 4GB RAM. Figure 4 shows memory scalability on the Proliant 6000 with 4 PentiumPro® 200MHz processors with 512K level 2 cache.

Figure 1: CPU Scalability of the Proliant 4500

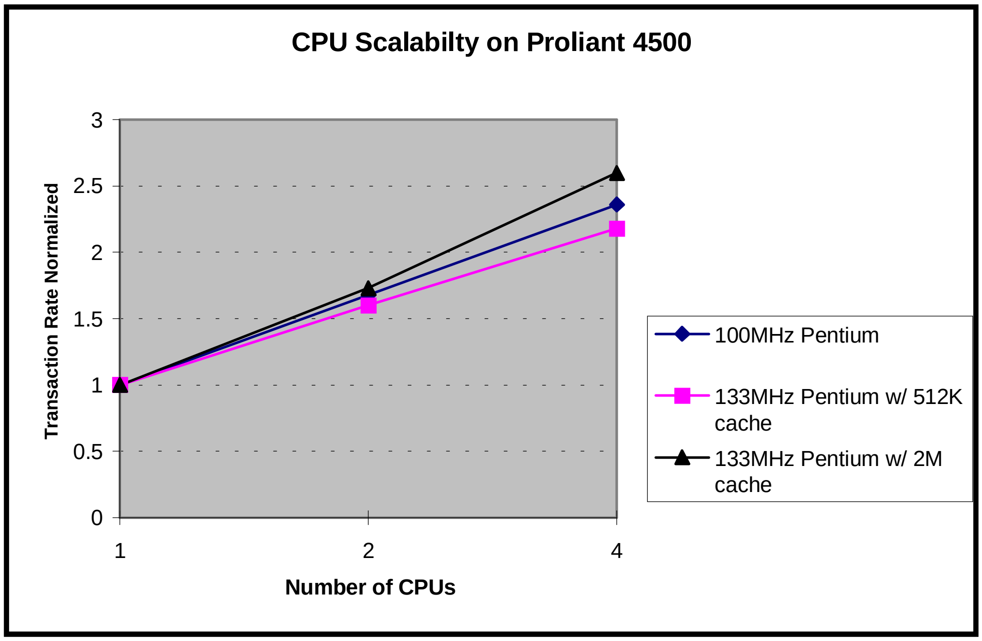


Figure 2: CPU Scalability for Proliant 5000

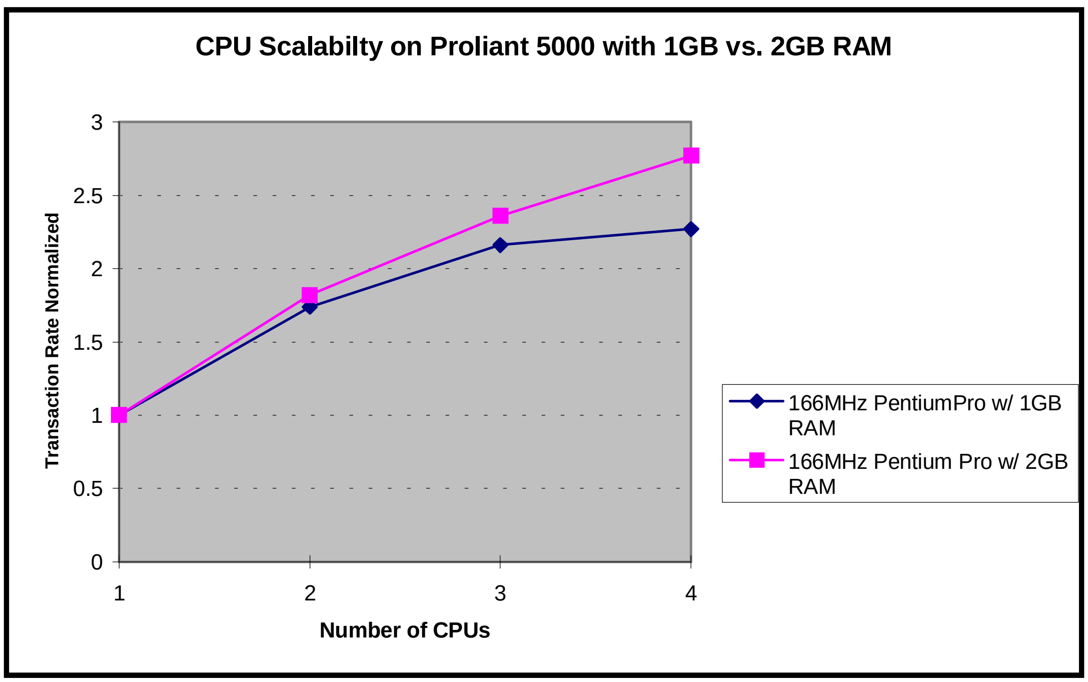


Figure 3: CPU Scalability for Proliant 6000

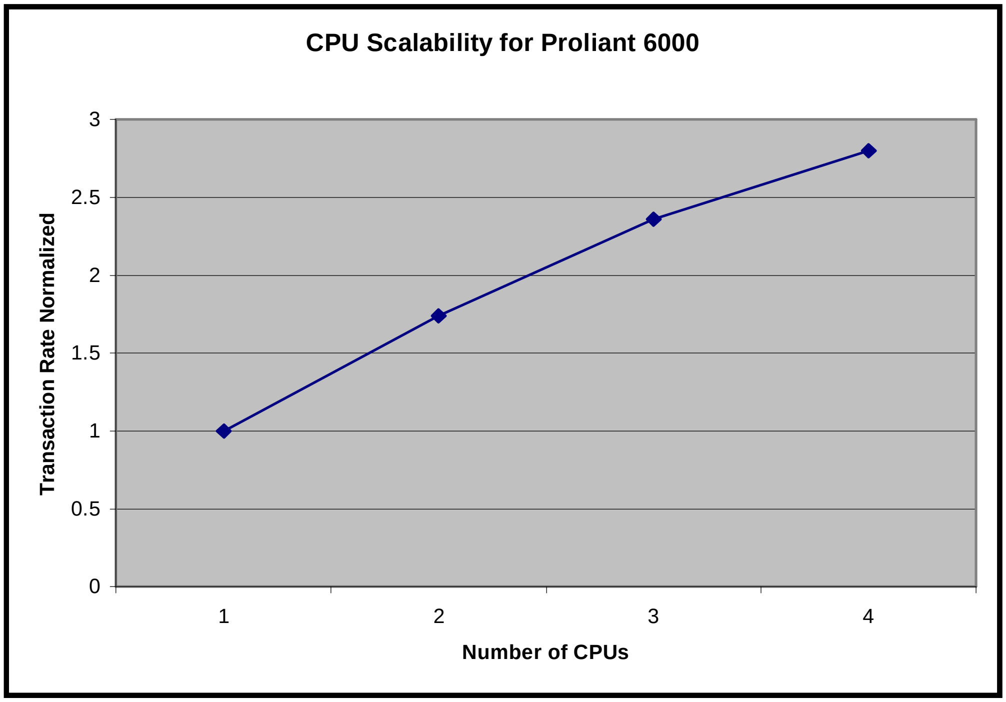
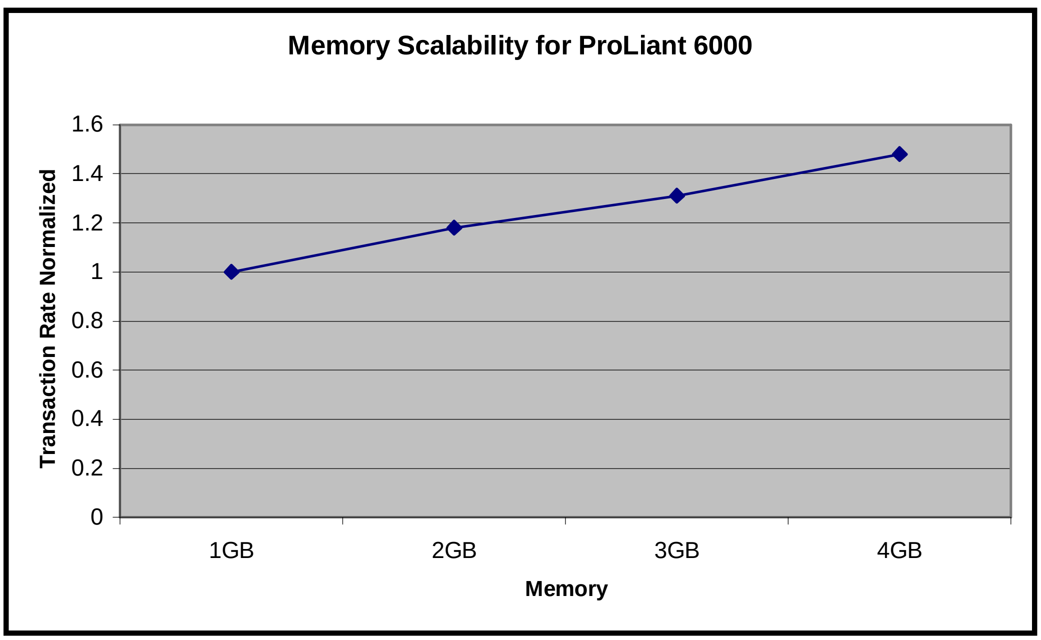


Figure 4: Memory Scalability for Proliant 6000



Initial CPU Recommendations

The choice of the right system processor depends on your environment. As technology evolves, more powerful processors are becoming available, pushing performance to new levels at very competitive costs per processing unit. PentiumPro®-based systems have become the standard for application servers, especially the PentiumPro® 166MHz and 200MHz processors.

Our recommendation is to carefully evaluate your environment, and experiment with various processor configurations, if possible. Always have future growth and expansion in mind. You may want to start with Pentium® processors for smaller departmental systems, and PentiumPro® processors for higher demand systems. If your environment has many concurrent users, you should evaluate benefits of multiprocessing.

Before upgrading the system processor(s), you should closely monitor performance of the system and tune it from the software perspective. If the performance bottleneck is in software, hardware upgrades can only partially improve performance. It may be more cost effective to tune the software rather than to purchase hardware upgrades. If the performance bottleneck is clearly at the system processor(s), upgrading to a higher speed processor, or adding another processor can dramatically improve performance.

CPU Tuning

To monitor CPU performance and determine whether your system is CPU bound or not, you can run the UnixWare performance monitor, **rtpm**. This should be run during a normal workload to get an accurate reading. You also may use the **sar** utility to save the information to a file. A system that is tuned well will have the following CPU characteristics:

- Most of the CPU utilization is in user mode (**%usr**). Again, this is verified by running *sar* or *rtpm* and looking at the percentage of CPU time spent in system and in user time. System time (**%sys**) can be thought of as operating system overhead such as time spent in the I/O subsystem or in system calls. The higher the percentage of user to system time that you have, the better. For a system performing mostly Sybase processes, 83 **%usr** and 17 **%sys** is a very good split, assuming there is no CPU idle time. The key is that Sybase should be getting most of the CPU time. Here is some sample output from *sar -u*:

```
> sar -u 5 5
```

09:00:10	**%usr**	**%sys**	%wio	%idle
09:00:15	**82**	**17**	0	1
09:00:20	**83**	**17**	0	0
09:00:25	**80**	**18**	0	2
09:00:30	**82**	**17**	0	1
09:00:35	**81**	**18**	0	0
Average	**82**	**17**	0	1

The above example shows a system that is CPU bound if all other areas had already been tuned and optimized for performance. CPU bound means that in spite of your efforts to tune the system, you cannot get more performance from it because there is no more processing power left on the CPU(s). If your database and application are well tuned, and if there is no idle time and no waiting on I/O on the CPU(s), then you may be CPU bound. In that case, an additional system processor or higher speed processors(s) could greatly improve response times.

Users should see good response times. A system that appears to be tuned well and is experiencing poor response times could have any or all of the following problems:

- An inefficient database design. This could include poor indexing schemes or inefficient layout of the data on the drives when the database was created.
- Inefficient statements in the database application, such as poor SQL statements in stored procedures.
- Excess latencies in the I/O subsystem or network.

Adding engines to Sybase

If you add a processor(s) to a server, you may need to reconfigure the number of Sybase engines. The parameter that sets the maximum number of Sybase engines that can be started when SQL Server boots is called **max online engines**. If this parameter is set to a number greater than the number of processors in the machine, then Sybase automatically starts with one engine per processor. The Sybase server *errorlog* will have a message indicating that the number of engines configured could not all be started. This is because a maximum of one Sybase engine per processor is allowed.

For a one processor environment, **max online engines** should be set to one. For SMP environments, one engine per processor is common. For example, if you have 4 processors in the server and most or all processes being performed on the system are Sybase processes, then you should set **max online engines** to 4. If there is a need to execute non-Sybase applications on the system, then it may be appropriate to set the parameter to 3, using only 3 of the 4 processors for Sybase engines. This will leave one processor available at all times to perform the non-Sybase application processes.

Disk Subsystem Planning

The objective of this section is to provide information on the benefits of the Compaq SMART-2 SCSI Array Controller Array Accelerator and the pros and cons of various fault tolerance methods.

Additional information on disk subsystem configuration and Compaq drive array technology can be found in the following manuals:

- *Configuring Compaq RAID Technology for Database Servers,* Compaq TechNote, P/N 184206-001
- Compaq SMART-2 Array Controller User Guide, P/N 184482-001
- Sybase SQL Server System Administration Guide, Doc. ID 32500-01-1100-02

Array Accelerator: Its Function and Benefit in a Sybase SQL Server Environment

The Array Accelerator is a feature of the SMART-2 SCSI Array Controllers. The main function of the Array Accelerator is to boost performance of disk operations by storing data in the cache memory on the controller. The SMART-2 Array Controller has 4MB of ECC memory. The Array Accelerator is shared among all logical drive volumes configured on the controller, and can be enabled/disabled on a per-logical-volume basis.

SMART-2 SCSI Array Controller characteristics

The SMART-2 controller allows the Array Accelerator to function as write cache, read-ahead cache, or a combination of both. You can, for example, configure the Array Accelerator on the particular controller to function as 50% read-ahead and 50% write cache. Then every logical volume on this controller is supported by the Array Accelerator in the 50/50 mode. The only exception is when the Array Accelerator is disabled for the volume.

Write Cache

When the Array Accelerator performs write caching, the drive controller writes data to the cache memory on the Array Accelerator rather than directly to the drives. The system can access this cache memory more than 100 times faster than accessing disk storage. The controller writes the data in the Array Accelerator to the drive at a later time, when the controller is otherwise idle.

Without the Array Accelerator's write cache, the application must wait until each write request is written out to the disk. Writing to a disk device is slower than posting the write request in the Array Accelerator, thus possibly resulting in decreased performance.

Read Cache

The SMART-2 controller uses the Array Accelerator to increase performance in some cases by anticipating possible future read requests. The Array Accelerator uses a multi-threaded algorithm to predict the next likely read operation for the drive array. That data is pre-read into the cache on the Array Accelerator and therefore is ready before you access it. When the SMART-2 controller receives a read request for the cached data, it can be burst read immediately into system memory, thus avoiding a disk access after the read request.

The read-ahead option of the SMART-2 SCSI Array controller can boost performance in environments that utilize sequential scans of data; for example, range lookups, data loads, table scans, decision support environments, etc. Environments with a very random I/O profile, such as on-line transaction processing, typically do not take advantage of the read-ahead capabilities of the controller, and in most cases it is beneficial to configure 100% of the Array Accelerator for write caching.

Housekeeper, Checkpoints, and Transaction Log Writes

There are three main write-intensive operations Sybase SQL Server performs: housekeeper, checkpoints, and transaction log writes.

- During idle time on the SQL Server, the **housekeeper** writes dirty pages from the data cache to the disk at a lower priority than the checkpoint process. Unlike a checkpoint process which must write all dirty pages from the data cache to disk before terminating, the housekeeper writes only what it can during idle times of the system. If the system is idle for a long enough period of time the housekeeper may actually write all dirty pages from the data cache to disk. When this occurs the housekeeper notifies the checkpoint process and requests that a checkpoint be performed on the database so that the transaction log will have a record that all dirty pages were written to disk at that time. (For details on configuring the housekeeper, see **Housekeeper Free Write Percent**)
- During **checkpoints**, Sybase SQL Server generates a large number of write requests in a short time interval. The main objective of the checkpoint is to write **all** dirty pages from the data cache to the disk. The time it takes to write the dirty pages depends on several factors, such as the configuration of the housekeeper and the **recovery interval** of the SQL Server. (See the explanation of **recovery interval**)

  In some environments, the amount of write activity that the checkpoint generates can saturate the Array Accelerator, thus interfering with read requests pending at the controller. Proper tuning of the housekeeper can help alleviate this problem.

- The **transaction log** activity is composed exclusively of sequential writes and does not saturate the Array Accelerator. However, the benefits of caching the transaction log writes at the SMART or SMART-2 SCSI Array Controller level with the Array Accelerator can have a significant beneficial impact on performance. For optimal performance the Array Accelerator should be enabled. It is very important to make sure you follow the guidelines below for data integrity if you choose to enable the Array Accelerator on the transaction log.

Integrity of cached data in the Array Accelerator

The Array Accelerator contains batteries that maintain any data in the cache if a system power failure occurs. Make sure you maintain the batteries in a good condition and fully charged (they are automatically recharged while system power is present). At a fully charged state, the batteries can preserve data in the Array Accelerator for four days. When power is restored to your system, an initialization feature writes the preserved data to the disk drives.

Another step to insure data integrity in case of system failure is to install an Uninterruptible Power Supply (UPS). Installation of a UPS will allow the controller to flush all data out to disk in the event of a power failure. The UPS does not, however, insure data integrity in the case of a controller failure, when valid data exists in the Array Accelerator. In that case, the Array Accelerator may be removed as a complete unit from one SMART-2 controller and installed on another, while preserving any data cached. The data will be written to disk upon power up.

Array Accelerator: Its Impact on Performance

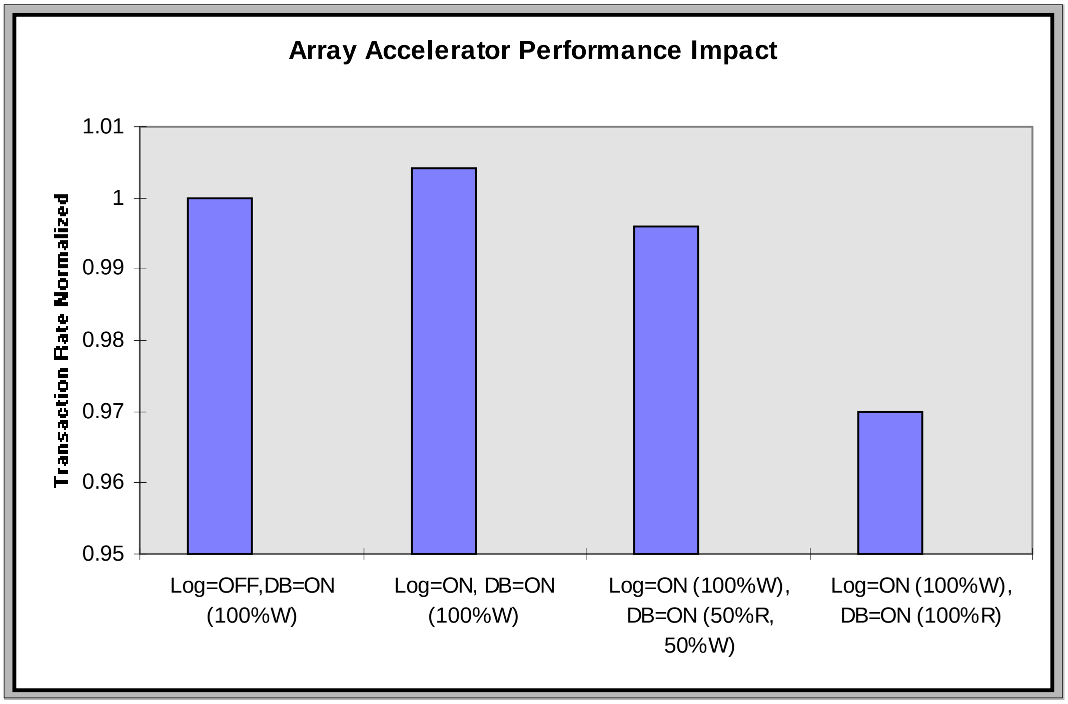
The following figure shows performance impact of various Array Accelerator configurations in an on-line transaction processing environment. For our baseline we chose to disable the Array Accelerator for the transaction log volume(s) and enable the Array Accelerator with 100% write cache for all data volumes.

When the Array Accelerator is disabled on the database volume(s), the performance greatly decreases since write requests cannot be cached in the Array Accelerator. On the other hand, allowing the controller to cache the transaction log writes can improve performance in some cases.

As already mentioned, partially configuring the Array Accelerator for read-ahead when the I/O profile is purely random does not improve the performance. On the contrary, the attempts of the controller to perform read-ahead when little-to-none is possible can hurt performance, as well as the effect of decreasing the write cache to allocate some read-ahead cache. As shown in Figure 3 below, the I/O profile for that test could not take advantage of read-ahead, so configuring 50% of the Array Accelerator cache for read-ahead actually decreased performance.

In our environment, the highest performance was achieved when the Array Accelerator was enabled for both the transaction log volume(s) and the database volume(s) and both were configured for 100% write cache. In Figure 3 below, each bar shows whether the Array Accelerator was enabled (ON) or disabled (OFF) for the log and for the data volumes. Also “W” shows the percent of the Array Accelerator cache dedicated to caching writes, and “R” shows the percent dedicated to caching reads.

Figure5: Array Accelerator Performance Impact



Configuring the Array Accelerator

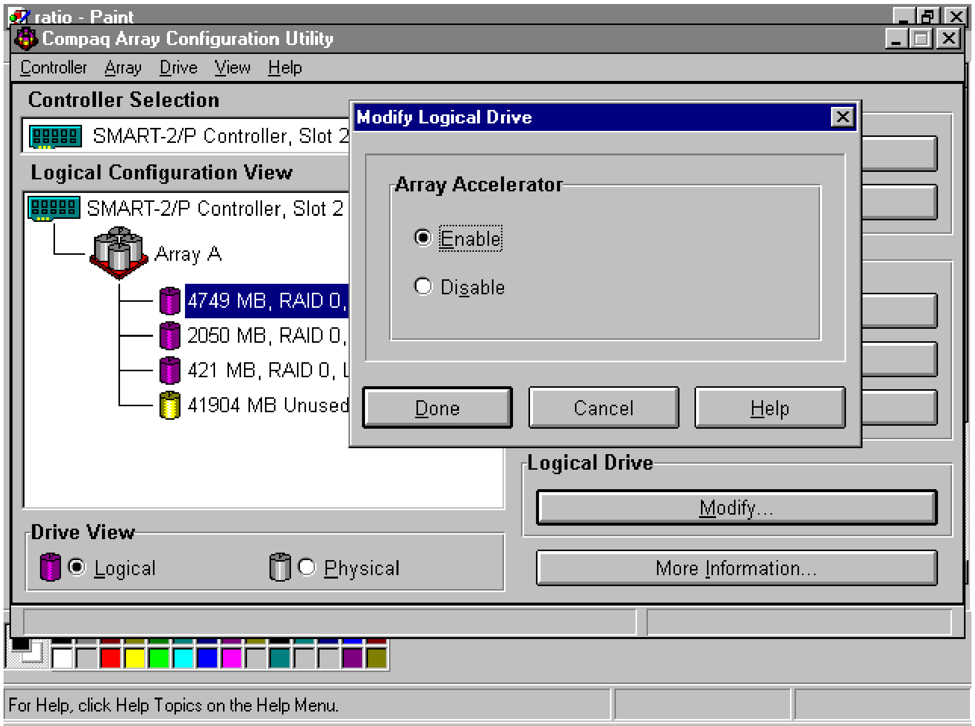
The Array Accelerator of the SMART-2 SCSI Array controller is configured via the **Compaq Array Configuration Utility**. Always obtain the latest copy of the appropriate utility for your Compaq SCSI Array controllers.

With the SMART-2 Array controller, you can select the ratio of read-ahead cache to write cache for every controller. Once selected, this Array Accelerator ratio will apply to all logical volumes on this controller that have the Array Accelerator enabled. You can enable or disable the Array Accelerator on a per-logical-volume basis.

To select the Array Accelerator ratio for a SMART-2 Array controller, you must install and run the Compaq Array Configuration Utility from diskettes or from the System Partition. Using the Array Configuration Utility, highlight the appropriate controller and use the "*Controller/Settings...*" option.

Figure 4 below shows how to enable/disable the Array Accelerator for a particular logical drive. Using the Array Configuration Utility, highlight the appropriate logical volume and use the "*Logical Drive/Modify...*" option.

Figure 6: SMART-2 Array Controller - Logical Drive Array Accelerator Settings



Fault Tolerance Considerations

Your transaction log needs to always be protected against a disk failure. Most mission critical sites protect both the transaction log and the database devices and they usually choose hardware-based fault tolerance. Non-mission critical sites are often satisfied with the protection of the transaction log only and performing frequent backups/dumps.

You have two choices of protecting your data:

- Use a RAID-based disk controller such as the Compaq SMART-2 SCSI Array Controller
- Use Sybase SQL Server-based mirroring/duplexing.

Below are some key points to be aware of when choosing the appropriate fault tolerant method. The performance differences between fault tolerance implementations can vary from insignificant to very significant, depending on your configuration and environment. Therefore, we omitted supplying performance differences for the purpose of not misleading our readers. We recommend that you evaluate the various fault tolerance methods using your own applications.

1. RAID-based Disk Controllers

- Hardware RAID is configurable on a logical volume basis. Therefore, the whole logical volume is protected by the appropriate fault tolerance. The capacity reduction depends on the size of the logical volume and the type of fault tolerance configured.
- Hardware RAID offers the best reliability and automatic recovery. When a drive fails, the system continues operating. Once the failed drive is replaced, the controller automatically rebuilds the new drive and restores the system to its full fault tolerant capabilities.
- Since the data protection occurs at the hardware (controller) level, there is no overhead on the system processor. This type of RAID is totally transparent to the operating system and

the applications. With a CPU-intensive application/environment, such as Sybase SQL Server, hardware-based fault tolerance can provide the best performance. Please refer to *Configuring Compaq RAID Technology for Database Servers* TechNote, published by Compaq (P/N 184206-001) for a more complete discussion.

2. Sybase SQL Server Mirroring/Duplexing

- Sybase SQL Server mirroring is based on Sybase SQL Server devices. This offers you the finest granularity and lowest capacity reduction due to duplicated data. Refer to *Sybase SQL Server, System Administration Guide* for guidelines on choosing which devices to mirror.
- You have an option of controller duplexing if you place the mirrored device on a different disk controller.
- Sybase SQL Server supports both serial and nonserial write mode of mirroring. When utilizing the default, serial write mode, writes to the first device must finish before writes to the second device begin. Changing from serial to nonserial write mode allows the writes to both devices to be queued immediately, one to each mirrored device. The nonserial write mode will incur less overhead than serial writes that results in a slight performance improvement.
- Mirroring through Sybase SQL Server induces an additional processing overhead on the system, resulting in a lower performance.

I/O Tuning

In well tuned Sybase systems, I/O is not a limiting factor. In order to assure that this is not a problem, the following factors need to be verified.

- Sequential I/O's are isolated to a controller volume, separate from volumes with random I/O's. This means a sequential I/O volume should be alone on a controller or on one port of the controller.
- Random I/O's are balanced across all drives allocated to data and indexes.
- Physical disk I/O limits are not exceeded. (See Table 1 below).
- There will be little or no waiting on I/O (**%wio**) by the CPU(s). This is verified by running the UnixWare tools *sar* or *rtpm*. No waiting on I/O indicates that the CPUs always have some work to do while there are outstanding I/Os. If asynchronous I/O is enabled for Sybase, then UnixWare will never show waiting on I/O. The exception to this is if some non-Sybase processes issue I/O requests that are not asynchronous, then you may see some waiting on I/O. (See section on AIO).

Separate Sequential and Random I/O's

In order to achieve maximum performance on data files being accessed sequentially, the disk(s) need to be dedicated to this purpose. Of primary importance are the Sybase transaction log files, which are accessed in a sequential, write-only fashion. Other partitions with little I/O activity can share the disk(s) with the transaction logs, such as the OS partition and swap.

In typical multi-user database systems, data access is random. This data should be spread out over as many physical disks as necessary to achieve random I/O rates that do not exceed recommendations (see Table 1 below). This is best achieved by using the disk striping available with the SMART-2 SCSI Array Controller. Spreading out the disk requests among many disks allows a high degree of parallelism to occur on accesses. Using the Compaq SMART-2 SCSI Array Controller ensures that the load will be balanced equally across the disks. For more information on optimizing array configurations refer to the Compaq TechNote, *Configuring Compaq RAID Technology for Database Servers*.

Layout of Tables and Files

In order to improve performance where disk I/O is a problem, keep in mind the following:

- Transaction log access is 100% sequential I/O and needs to be isolated if possible. Speed of the log is essential to the performance of the system. If possible, these drives should be fault tolerant, either mirrored or distributed data guarding. Hardware fault tolerance provides the maximum performance and reliability. See the Compaq Database Engineering TechNote *Configuring Compaq RAID Technology for Database Servers.*
- Data file access is usually random and needs to be spread across as many drives as necessary. By increasing the number of physical drives, greater I/O rates can be achieved. Using a striped array will assure that the I/O's are well distributed.

Use the following guidelines when monitoring and optimizing the drive subsystem. You should not have more I/O requests (disk transfers) per second per disk drive than the values in the table on the next page.

Table 1: Maximum I/O operations per Second per Disk Drive

	1.0GB drives (Max I/Os per Second per Drive)	2.1GB drives (Max I/Os per Second per Drive)	4.3GB drives (Max I/Os per Second per Drive)
Sequential Writes (Transaction Log)	≈150	≈160	≈180
Random Reads/Writes (Database Access)	≈30-40	≈50	≈55-60

NOTE: With the Array Accelerator enabled, you may actually see substantially higher I/O per second per drive rates than suggested above, especially during checkpoint. This increase is due to the Array Accelerator write posting some of these I/Os. In the Compaq Database Performance labs we have actually measured rates of up to 90 random I/Os per second per drive, some of which were being temporarily cached by the Array Accelerator.

Checking Disk I/O Rate

Try not to overload any individual disk with random I/Os. To determine the I/O rate per drive, first determine the number of I/O's per second to each logical volume. This can be done with *sar -d*, *rtpm* or 3rd party tools. Take the number of I/O's per second to each logical volume and divide by the number of physical disks in that logical volume. This will provide the number of I/O's per second per disk. If this number exceeds recommended I/O's per second rating, adding more physical disks should improve average system performance. The following *sar* command provides an example for determining the reads and writes per second, **r+w/s**, for each logical controller volume on a system.

> sar -d 10 20

10:51:47	device	%busy	avque	**r+w/s**	blks/s	avwait	avserv
…							
Average	sd011	77	1.5	125	1001	9.3	6.1
	sd012	94	3.2	**301**	1504	10.0	3.1
	sd013	88	2.7	257	1031	9.1	3.4
	sd014	95	3.5	317	1271	10.5	3.0

It is best to use values from the *sar* average over a period of time to calculate I/O rates. Using the above output as an example, if volume sd012 contained 7-2.1GB disk drives that were used for random data access, then the calculation to find how many I/O's per second per drive were done on that volume follows:

301 r+w/s divided by 7 drives in volume = 43 r+w/s per drive (< 50)

Therefore, the sd012 logical volume is within Compaq's recommended I/O limit of 50 I/O's per second for the 2.1GB drive.

Enabling Asynchronous I/O

Under virtually all circumstances disk I/O runs faster asynchronously that synchronously. This is because when SQL Server issues an asynchronous I/O, it does not have to wait for a response before issuing further I/O's.

Asynchronous I/O (AIO) is available on the UnixWare 2.1 operating system. AIO can be used with databases built using RAW devices or file systems and will achieve the best performance. Using AIO to RAW devices is preferable because it is more efficient than AIO to file systems. To enable AIO through UnixWare you need to do two things:

1. Do **chmod 666 /dev/async** to give permissions on that file to the user (possibly the system administrator) who starts up the SQL Server.
2. Edit the file **/etc/conf/node.d/async**, and change the mode number there to **666** also.

The Sybase parameter **allow SQL server async i/o,** by default of 1, enables AIO for Sybase. So you do not have to make any changes to Sybase to use AIO. If you choose **not** to use AIO, you must set **allow SQL server async i/o** = 0. See page 11-26 of <u>Sybase SQL Server System Administration Guide</u> for more details.

The Sybase parameter **o/s async i/o enabled** indicates whether or not AIO is enabled through the OS (1=enabled and 0=disabled). You cannot configure this parameter as it is read-only. It is an easy way to double check that AIO is enabled through the OS. See page 11-60 of <u>Sybase SQL Server System Administration Guide</u> for more details.

<u>Asynchronous I/O UnixWare Parameters</u>

There are two UnixWare parameters that may need tuned for AIO, **NUM_AIO** and **AIO_LISTIO_MAX**.

NUM_AIO sets the number of AIO control blocks available for kernel use. One control block is needed each time an asynchronous request is initiated. If no control blocks are available, then an asynchronous call will fail, therefore this tunable determines the maximum number of outstanding AIO's. The default value is 256, which may be sufficient.

AIO_LISTIO_MAX sets the maximum number of I/O's that can be submitted in a single request to the kernel. The default value is 128, which may not need tuning.

For I/O-intensive applications, you may need to increase these two parameters. Doubling each of the default values should be sufficient.

Memory Tuning

Initial Memory Recommendations for Sybase

Sybase recommends a minimum of 16 MB memory for SQL Server, plus 70K of RAM per additional user. These initial memory requirements may need to be increased based upon the number of users, complexity of queries, number of disk controllers, amount of total disk storage, number of network cards, and intensity of the workload in your environment. To estimate the minimum memory requirements for a 512 user system simply multiply 512 * 70KB for a total of 35,840KB (i.e. 35 MB) and add this to the Sybase minimum requirement of 16 MB for a total of 51 MB.

Do not tune Sybase memory up at the expense of swapping. Swapping will degrade system performance. (See next page to learn how to check for swapping.)

UnixWare Patches for 4GB Support and Sybase Support

UnixWare 2.1 currently supports up to 4GB of RAM. You will need to install two SCO patches to get this support, ptf3190 and ptf3191. Another patch you need in order to start Sybase on UnixWare 2.1 is ptf3142. These three patches can be downloaded from the SCO web site at www.sco.com. Look under "Support", then "Supported Patches and Supplements", "UnixWare 2.1.x". Read the ptf3190.txt, ptf3191.txt, and ptf3142.txt files for an explanation of the patches. To install the patches on your system login as root and do for both:

1. uncompress ptf3190.Z
2. pkgadd –d /*directory/…/*ptf3190

UnixWare Shared Memory Parameters

The amount of shared memory allowed in UnixWare needs to equal or exceed the amount of shared memory required for Sybase. The OS tunable **SHMMAX** sets the maximum size of a single shared memory segment. The maximum amount of shared memory that can be used by a single process is equal to the OS tunables **SHMMAX*SHMSEG** (maximum shared memory segment size * maximum number of shared memory segments). Setting **SHMMAX** equal to the total amount of RAM available in your system for Sybase use will ensure that only one shared memory segment will be allocated for Sybase. Having more than one shared memory segment allocated to Sybase is less efficient than having one large shared memory segment. The number of shared memory segments that have been allocated on your system can be verified with the OS command *ipcs* (see example on next page).


```
> ipcs -b
IPC status from /dev/kmem as of Sun Jan 29 10:33:19 1995
T    ID    KEY         MODE        OWNER    GROUP QBYTES
Message Queues:
T    ID    KEY         MODE        OWNER    GROUP  SEGSZ
Shared Memory:
m    900 0x10043232 --rw-r----- sybase    dba       50745344
T    ID    KEY         MODE        OWNER    GROUP NSEMS
Semaphores:
```

Note**:** Only one shared memory segment has been allocated with a size of 50745344 Bytes.

By increasing the amount of shared memory allocated, you are reducing the amount of memory available to non-Sybase processes. Be careful not to increase Sybase's memory size to a point where swapping may occur. Swapping can be detected by noting swapping activity with *sar -w* or with *rtpm*. To see how much memory is free, use *sar -r*. If there is always a significant number of MB of free memory, then you may want to allocate a greater amount of memory to the Sybase server to use for data and procedure cache. If you do, check again for swapping. Below are sample outputs from the *sar -w* and s*ar -r* commands.

> sar -w

10:09:28	**swpin/s**	bswin/s	**swpot/s**	bswot/s	pswch/s
10:09:33	**0.00**	0.0	**0.00**	0.0	150
10:09:38	**0.00**	0.0	**0.00**	0.0	123
10:09:43	**0.00**	0.0	**0.00**	0.0	134
10:09:48	**0.00**	0.0	**0.00**	0.0	152
10:09:54	**0.00**	0.0	**0.00**	0.0	144
Average	**0.00**	0.0	**0.00**	0.0	140

The blocks swapped in and out per second (**bswin/s** & **bswot/s**) are given in 512 byte blocks. The number of **swpin/s** and **swpot/s** per second is zero. This indicates that no swapping is occurring.

If you do see that swapping is occurring (**swpot/s** or **swpin/s** > 0), reduce the memory size given to SQL Server (run the **sp_configure total memory** command in isql, or edit the Sybase configuration file), restart the SQL Server, and check the system monitors again. In the case of a system with non-Sybase applications running as well, try reducing the memory used by those applications also. Continue this cycle until you see no swapping. In some cases, adding more memory to the machine may be the best solution.

>sar -r 5 5

10:09:54	**freemem**	freeswp
10:09:59	**28635**	196608
10:10:04	**28627**	196608
10:10:09	**28637**	196608
10:10:15	**28637**	196608
10:10:20	**28637**	196608
Average	**28636**	196608

Freemem is given in number of 4K pages and **freeswp** is in 512-Byte disk blocks. The output above shows that about 111MB of memory is free. In this case, it may be helpful to allocate more memory to Sybase to use for data and procedure cache.

UnixWare PSE Feature

To make the most efficient use of system memory on an Intel Pentium or PentiumPro® processor you should take advantage of the Page Size Extension feature (PSE). The default page size allocated in memory is 4Kbytes. Enabling the PSE feature allows memory to be allocated in contiguous **4 Mbyte pages** that can be used for shared memory. The larger page size allows more efficient page lookup on Intel Pentium® processors. To configure and enable PSE, do the following:

1) Add the kernel tuning parameter called **PSE_PHYSMEM** to the **stune** file. The size you give specifies how much memory will be allocated using PSE, 4Mbyte pages. The number is in bytes and is rounded up to the nearest 4 Mbyte. To allow Sybase to use this shared memory area, both **PSE_PHYSMEM** and **SHMMAX** must be tuned to a size equal to or greater than the memory size given to SQL Server. Setting **PSE_PHYSMEM** equal to the **total memory** size given to Sybase is recommended when running mostly Sybase user processes.

2) If you need to increase the amount of memory to use for PSE beyond the default, you must edit the file **/etc/conf/mtune.d/pse**. (If you have configured beyond the default, you will see an error message when trying to rebuild the kernel). Change the last number entry in the line for **PSE_PHYSMEM** to be the same number of bytes that you specified in the **stune** file for **PSE_PHYSMEM**. Otherwise, you never need to edit this file.

3) Now that you have the memory size specified, you must enable PSE. To do this, edit the file **/etc/conf/sdevice.d/pse**. Change the “N” to a “Y”.

4) Now you must relink the UnixWare kernel by running, as root, **/etc/conf/bin/idbuild -B** and **reboot** the system.

UnixWare User Process Capacity Parameters

There are two UnixWare parameters relating to user processes that may need tuning, **MAXUP** and **NPROC**.

The OS parameter **MAXUP** specifies the maximum number of processes allowed on the system on a per user basis. The OS parameter **NPROC** specifies the maximum number of processes allowed on the system. **NPROC** must be at least 50 greater than **MAXUP** to allow for other OS and user processes to run. Both **MAXUP** and **NPROC** are tuned automatically based on the amount of memory in your system. To check on the current value, use the OS command *sysdef*. You may need to increase these values if they are not sufficient. Just add them into the **stune** file, relink the UnixWare kernel, and reboot the machine.

For descriptions of these and more of the OS system parameters, refer to the System Tuner program on your UnixWare desktop or to the UnixWare on-line documentation.

Sybase Installation Issues

Installation with CD-ROM

If you are installing Sybase from a CD-ROM, you must mount the CD-ROM to a directory such as */mnt* to run the *sybload* program. Here is an example:

Log in as root:

> mount -F -r cdfs /dev/cdrom1 /mnt

Log out and log back in as your System Administrator and do:

> cd $SYBASE

> /mnt/sybload -D

Now you should be able to go to the *$SYBASE/install* directory and run *sybinit*.

Sybase Devices

Managing disk partitions

After creating the disk partitions through UnixWare, it is helpful to link each partition, or disk slice, to a more readable logical name for Sybase use. You could create a directory, for example /dev/sybase, and link each disk partition to a filename, in that directory, that would specify which table or index is on that partition. In the case of a table called customer, which has its data on disk slice /dev/rdsk/c1b0t0d0s1 and its index on slice /dev/rdk/c2b0t0d0s1, the link commands could be:

ln -s /dev/rdsk/c1b0t0d0s1 /dev/sybase/cust_data

ln -s /dev/rdsk/c2b0t0d0s1 /dev/sybase/cust_index

Now, the device can be referred to by the linked name, /dev/sybase/cust_data, rather than the UnixWare device name, /dev/rdsk/c1b0t0d0s1. It is a good idea to put all the link commands into a shell script so they can be easily executed in the future.

Device Permissions

You must give Sybase permission to access the devices you've created before you can install the master and sybsystemprocs databases in *sybinit*. The user that is going to install and start the Sybase server needs permissions on all devices that will be used by Sybase. Simply use the

UnixWare commands *chown*, *chgrp*, and *chmod*. For example, if the Sybase system administrator (who will start up SQL Server), uses the login name *sybase*, (and his home directory, /home/sybase, is the directory where SQL Server will be installed), and the group is called *sybgrp*, then the commands would be:

chown sybase /dev/sybase/cust_data

chgrp sybgrp /dev/sybase/cust_data

chmod 666 /dev/sybase/cust_data

Do the same for all Sybase devices.

SQL Server 11 Configuration and Tuning Parameters

sp_configure and Configuration file (*servername.cfg*)

With Sybase SQL Server 11 there are two ways to configure your server. One way is by using the **sp_configure** command that was used for SQL Server 10. With **sp_configure** you can set a parameter while the server is running, and you may or may not have to restart the server for the new value to take effect, depending on whether the parameter is a static or dynamic one. You can check in the Sybase SQL Server System Administrator Guide, chapter 11, under the section "Details on Configuration Parameters" to find whether a parameter is static or dynamic.

The second way to configure your server is new for SQL Server 11. A server configuration file, named *servername.cfg*, where *servername* is the name given to your SQL Server, is created when you install SQL Server. All of the configuration parameters are found in this file and their values can be edited. To make a changed parameter value take effect, simply shutdown and restart SQL Server. See page 11-9 of Sybase SQL Server System Administration Guide for more details.

Note: The configuration file eliminates the need for the SQL Server 10 *buildmaster* command, which was used to change a Sybase parameter while the server was down.

Recovery Interval in Minutes

Recovery interval in minutes sets the maximum number of minutes per database that SQL Server should use to complete its recovery procedures in case of a system failure. The recovery procedure recovers transactions that occurred after the last checkpoint. SQL Server uses the **recovery interval in minutes** to decide when to checkpoint each database. (See Sybase SQL Server System Administration Guide, section "Details on Configuration Parameters" for more detail on this parameter.)

During a checkpoint, the changed or 'dirty' pages in the data cache are written to disk, leaving the data cache buffers 'clean'. During the checkpoint, user tasks continue to run, but their response times may become longer because the high number of disk writes being done by the checkpoint take up CPU time. Immediately after a checkpoint, user response times will be slightly faster than normal because SQL Server is more likely to find a clean buffer when it needs a new one, and because buffers that move into the wash area are more likely to be clean and will not need to be written to disk. Once the data cache is filled, user response times will slow down to 'normal' levels due to necessary disk accesses and memory management.

This **recovery interval in minutes** should be left at its default value of 5 minutes unless you are willing to take the risk of setting it to a higher value. If the recovery interval is set too long, the user response times will deteriorate and become intolerable when a checkpoint does occur. Setting the recovery interval too short will waste valuable CPU cycles and generate excessive disk I/O. See page 11-17 of Sybase SQL Server System Administration Guide for more details.

Number of User Connections

This parameter sets the maximum number of users that can be connected to SQL Server at the same time. It should be set as low as possible to leave more memory for the data cache. Setting this parameter too high wastes memory and increases the size of the table that Sybase needs to scan when looking for new user logins or existing users logging out. The default is 25.

SQL Server allocates approximately 70KB of memory as overhead per user connection. You may have to readjust the **total memory** value depending upon the number of user connections your environment requires. If you increase the **default network packet size** or **stack size** configuration parameters, the amount of memory per user connection increases also. See page 11-100 of Sybase SQL Server System Administration Guide for more details.

Total Memory

This setting can be initially left at its default value. If you have a large number of active user connections or users are unable to connect to the database you will need to increase this value. The value expressed by **sp_configure** and in the *servername.cfg* file is in 2KB pages. This memory area is used to store the data and procedure caches. Refer to the sections in this document on "Initial Memory Recommendations" and "Number of User Connections" for more details on determining the proper value for this parameter. Further tuning can be done by adjusting this value up or down for best user response times. In general, more memory allocated to Sybase gives better performance, up to the point where swapping starts to occur.

One method to improve performance would be to set **total memory** large enough for heavily used tables and indexes to fit into the data cache area. Be careful in doing this because setting this value higher than necessary may cause a LOSS of performance not a gain. SQL Server may end up spending too much time attempting to manage the data cache memory area instead of utilizing it. If you set the value for memory higher than the amount of memory available to the server, you will be unable to start SQL Server. If this occurs, edit the parameter **total memory** in the *servername.cfg* file and try to restart. See page 11-64 of Sybase SQL Server System Administration Guide for more details.

Procedure Cache Percent

This setting is a percentage of the memory allocated to SQL Server that is reserved for caching of stored procedures. The initial default setting of 20 percent should be sufficient for most database environments. You may want to experiment running with a lower percentage for procedure cache, to leave more memory for data cache. On the other hand, if you run a lot of different procedures or ad hoc queries, you may want to increase this value. The procedure cache is not only used to store the compiled stored procedures. The space is also used during the creation of stored procedures and to compile queries. See page 11-24 of Sybase SQL Server System Administration Guide for more details.

To find the number of 2K pages that a stored procedure requires in memory, you can run this isql statement:

select (count(*)/8) +1 from sysprocedures where id=object_id("*procedure_name*")

Or you can use the isql command dbcc memusage which gives information on the twenty largest stored procedures.

SQL Server Clock Tick Length

The **sql server clock tick length** parameter specifies the duration of the server's clock tick, in microseconds. The default value for the UnixWare platform is 100,000. CPU-bound tasks will benefit from increasing the clock tick length (up to 1,000,000) because it allows the tasks to execute longer between context switches. For mixed applications with lots of CPU-bound tasks, decreasing the clock tick length will help I/O-bound tasks. A recommended value to try is 20,000. Shortening the clock tick length will cause CPU-bound tasks to context switch more frequently and allow other tasks greater access to the CPU. See page 11-95 of Sybase SQL Server System Administration Guide for more details.

HouseKeeper Free Write Percent

Housekeeper free write percent specifies the maximum percentage by which the housekeeper task can increase database writes. Valid values range from 0 to 100. The default is 1. A value of 0 disables the housekeeper.

For example, to allow the housekeeper to increase disk writes up to 25 percent above normal, set the **housekeeper free write percent** to 25. See page 11-75 of Sybase SQL Server System Administration Guide for more details.

Named Data Caches

Dedicated named data caches provide a method of reserving a portion of the Sybase data buffer area for a specific database, index or table. For more information on using dedicated named caches refer to Chapter 9 "Configuring Data Caches" in the Sybase SQL Server System Administration Guide. For more information on performance tuning of the data caches refer to Chatper 15 "Memory Use and Performance", Section "Named Data Caches and Performance" in the Sybase SQL Server Performance and Tuning Guide.

Large I/O

By default, Sybase utilizes an I/O block size of 2K. By using dedicated named caches you can enable I/O block sizes larger than 2K. If a dedicated named cache is configured to use a block size of 16K, a request for 16K of data will be filled in a single read request to the operating system. Without large I/O the request would generate 8 separate reads of 2K each. For more information on performance tuning of large I/O refer to Chatper 15 "Memory Use and Performance", Section "Large I/Os and Performance" in the Sybase SQL Server Performance and Tuning Guide.

Network Characteristics of a SQL Server Environment

Typically a client workstation assembles a group of SQL commands and submits them for execution by the database server. The server processes the commands and returns the resultant data. Rather than having the client workstation send a huge grouping of SQL commands, profile the queries. Determine if any of the queries are candidates for conversion to stored procedures. A stored procedure is a grouping of 'standardized' SQL query commands that are pre-compiled and placed into the procedure area of the database by the System Administrator. The stored procedure can then be referenced by name for execution. Utilizing stored procedures for most of the standard DBMS activities will reduce the amount of network traffic and will use less server CPU resources to process the query.

Compaq Insight Manager (CIM)

CIM is a Windows based utility that uses SNMP in conjunction with OS and Driver Agents on the server to report hardware failures and system degradation due to a hardware problem. CIM can be configured to page the System Administrator if a component is failing. Using CIM pre-failure warranty will allow a hardware component to be replaced under warranty before it fails. CIM monitors system hardware and a few OS components.

Conclusion

The information in this paper is not a complete tuning guide but a supplement to other tuning information provided by Sybase and SCO. To achieve an optimal configuration, there are several factors to include. The application, the hardware, and the OS are all areas that must be carefully planned and tuned. The tuning process is iterative and will be done several times in order to achieve the most optimal performance possible. We hope that the information provided in this paper will help in this process. The information given is based on experience in tuning Sybase on UnixWare 2.1, however, each configuration is unique. Although all of the hints given here have been tested extensively, do not assume that tuning a specific parameter will always give the desired result. Do not be afraid to experiment.

We welcome feedback on your configurations and experiences to improve our information products in the future. Please send us any comments or suggestions on the attached form, attaching addition sheets if necessary. This will help us tailor future information products to your needs, and will enable us to make future revisions of this document and related new information products available to you.

References

Sybase SQL Server System Administrator Guide, Document ID: 32500-01-1100-02

Sybase SQL Server Performance and Tuning Guide, Document ID: 32645-01-1100-02

Compaq SMART-2 Array Controller User Guide, P/N 184482-001

User Registration/Evaluation Form

Please fill out and return to us this registration/evaluation form to help us keep you up to date with future revisions of this document and related new information products. Your effort will help us improve the quality of the future information products.

Name: ____________________
Title: ____________________
Company: ____________________
Address: ____________________
____
____
Phone Number: ____________________

Please evaluate the quality of this document:

	Excellent				Poor
Technical Accuracy	1	2	3	4	5
Organization	1	2	3	4	5
Clarity	1	2	3	4	5
Completeness	1	2	3	4	5
Timeliness	1	2	3	4	5

Please indicate the type of environment you have at your site:

Operating Systems	RDBMS	Processing Type
❑ SCO Unix Open Server	❑ Microsoft SQL Server	❑ On-Line Transaction Processing
❑ Microsoft Windows NT	❑ Sybase SQL Server 11	❑ Decision Support
❑ IBM OS/2	❑ Oracle 7	❑ Batch Processing
❑ Novell NetWare	❑ Other:	❑ Other:
❑ SCO UnixWare		
❑ Other:		

Please indicate the type of information you would like us to provide in the future:

Topic	Operating Systems	RDBMS
❑ Configuration and Tuning	❑ Microsoft Windows NT	❑ Microsoft SQL Server
❑ Capacity Planning	❑ Novell NetWare	❑ Sybase SQL Server 11
❑ Integration Information	❑ IBM OS/2	❑ Oracle 7
❑ Competitive Analysis	❑ SCO UnixWare	❑ Other:
❑ Systems Management	❑ SCO Unix Open Server	
❑ Other:	❑ Other:	

Additional Comments:

Return to:
Database Engineering
Compaq Computer Corporation
MailCode 090803
20555 SH 249
Houston, Texas 77070