Configuration and Tuning of
Sybase System 10 for Novell UnixWare 2.0
on Compaq Servers

White Paper

Prepared By

Database Engineering

Compaq Computer Corporation

May 1995

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.

Configuration and Tuning of Sybase System 10

for Novell UnixWare 2.0

on Compaq Servers

First Edition (May 1995)

Document Number 181A/0595

Compaq Computer Corporation

Table of Contents

Configuration and Tuning of Sybase System 10 for Novell UnixWare 2.0 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 System 10 on UnixWare 2.0 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 System 10 for UnixWare 2.0. Since there is an abundance of information available concerning general tuning of System 10 on the UNIX platform, much of the discussions will be focused on Compaq specific tuning. Wherever possible, references will be made to other useful tuning documentation.

Other publications covering these and related topics listed below:

- “Configuring Compaq RAID Technology for Database Servers”, Document Number 267A/0294
- **SYBASE SQL Server System Administration Guide**, Document ID:32500-01-1000-02
- **SYBASE SQL Server Troubleshooting Guide**, Document ID:39998-01-1000-0
- 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 important to tune the application to take advantage of the system. Due to the diversity of applications they are beyond the scope of this paper.

The tuning process and parameters in this paper should be used as a starting point. Tuning is an iterative process that evolves as user and work loads change on your system.

An optimally tuned Sybase System 10 on UnixWare 2.0 system should have the following characteristics:

- There will be little or no waiting on I/O. This is verified by running *sar* or *rtpm*. This indicates that the CPUs always have some work to do while there are outstanding I/Os.
- Most of the CPU utilization is in user mode. Again, this is verified by running *sar* or *rtpm* and looking at the percentage of CPU time spent in kernel and in user time. System time 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.

```
> sar -u 5 5


09:00:10    %usr    %sys    %wio    %idle
09:00:15    69      23      6       1
09:00:20    69      24      8       0
09:00:25    68      23      7       2
09:00:30    67      24      8       1
09:00:35    69      24      7       0

Average     68      24      7       1
```

- Users should see good response times. A system that appears to be tuned well and is experiencing poor response times could have inefficient statements in the application or could have excess latencies in the I/O subsystem or network. Since well tuned database applications are CPU bound, an additional system processor could improve response times.

I/O Tuning

In most 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 their own controller volume.
- Random I/O’s are balanced across all drives allocated to data and indexes.
- Physical disk I/O limits are not exceeded.

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 (unless your machine is memory limited).

In typical, multi-user database systems, file access is random. These files should be spread out over as many physical disks as necessary to achieve random I/O rates that do not exceed recommendations. This is best achieved by using the disk striping available with the Compaq SMART Array controller. Spreading out the disk requests among many disks allows a high degree of parallelism to occur on accesses. Using the Compaq SMART Array controller ensures that the load will be balanced equally across the disks. For more information on optimizing array configurations refer to the Compaq White-Paper *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. Hardware fault tolerance provides the maximum performance and reliability. See the Compaq Database Engineering White Paper *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.

Checking Disk I/O Rate

Try not to overload any individual disk with random I/Os. Compaq recommends random I/Os not exceed 50 I/Os per second per drive for 2GB drives and not exceed 40 I/Os per second per drive for 1GB and 500MB drives. To determine the I/O rate per driver 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 I/O, r+w/s, for each logical controller volume on a system.

```
> sar -d 25 1


10:51:47   device   %busy    avque    r+w/s    blks/s    avwait    avserv

10:52:12   sd011    26.13    218.81   156.95   712.22    362.65    1.66

           sd012    100.00   56.07    295.67   1178.64   681.90    12.38

           sd013    100.00   89.28    331.34   1319.40   1140.51   12.92

           sd014    100.00   109.97   260.40   1029.66   1605.28   14.73
```

Enabling Asynchronous I/O

Asynchronous I/O is available on the UnixWare 2.0 operating system. AIO can be used with databases built using RAW devices or file systems and will achieve the best performance. To enable AIO 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. Open the file */etc/conf/node.d/async*, and change the mode number there to 666 also.

Sybase will automatically use AIO. No parameters need to be set in the SQL Server.

Sybase Installation Issues

Pre-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:

```
> cd $SYBASE

> /mnt/sybload -D
```

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

Sybase Installation Default Parameters

Sybase SQL Server will boot at the end of *sybinit.* It starts up with one data engine and the default configuration values. You will probably want to change some or all of the following parameters. (See the section ‘System 10 Server Configuration and Tuning Parameters’ for more detail on these. Also, refer to the ‘Sybase SQL Server System Administration Guide’.)

memory- Sets the memory size, in 2K units, that SQL Server allocates from the OS.

devices- Sets the number of database devices the SQL Server can use. Include master and log devices in this count. The default is 10. Devices needs to be set before doing **disk init** to initialize your devices so the server will restart successfully.

user connections- Sets the maximum number of connections that can connect to the SQL Server at the same time.

max online engines- Sets the number of data server engines that will start up when SQL Server boots. For a one processor environment, this should be set to one. For SMP environments, one engine per processor is common. If there are a lot of non-server processes being done by the users, then the number of data engines configured should be one less than the number of CPUs.

Memory Tuning

Initial Memory Recommendations for Sybase

Sybase recommends a minimum of 48 MB memory to install and run SQL Server. 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 NICs and intensity of the workload in your environment. For each user, SQL Server allocates approximately 40KB for the users' stack area and an additional 5-10KB for the users' procedure cache. Therefore to estimate the minimum memory requirements for a 512 user system simply multiply 512 * 50KB for a total of 25,600KB (i.e. 25 MB) and add this to the Sybase minimum requirement of 48 MB for a total of 63 MB.

Do not tune Sybase memory up at the expense of swapping. Swapping will degrade system performance more than the advantages acquired by giving more memory to Sybase. (See below on how to check for swapping.)

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 maximum amount of shared memory allowed in a system is determined by several tunable parameters. The OS tunables **SVMMLIM** and **HVMMLIM**, sets the maximum amount of memory available to a user on the system. 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). Check to see that only one shared memory segment is being allocated. If more than one segment is being allocated, it is less efficient than having one large shared memory segment and you should increase **SHMMAX**. This can be verified with the OS command *ipcs.*

```
>  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.

By increasing the amount of shared memory allocated, you are reducing the amount of memory available to non-server user processes. Be careful not to reduce this memory to a point where swapping may occur. Swapping can be detected by noting the available space on the swap file with the UNIX command */etc/swap* -l or by noting swapping activity with *sar -r*, *sar -w*, or with *rtpm*.

```
sar -r 5 5
```

```
10:09:54      freemem      freeswp
10:09:59      28640        196608
10:10:04      28627        196608
10:10:09      28637        196608
10:10:15      28637        196608
10:10:20      28637        196608

Average       28636        196608
```

These values are given in number of 4k pages. As you can see there is plenty of free memory. If the amount of freemem dips below 100 and freeswp decreases, then swapping is occurring.

```
> 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        20
10:09:38    0.00       0.0        0.00       0.0        23
10:09:43    0.00       0.0        0.00       0.0        24
10:09:48    0.00       0.0        0.00       0.0        22
10:09:54    0.00       0.0        0.00       0.0        23

Average     0.00       0.0        0.00       0.0        22
```

Note: The blocks swapped in and out per second (bswin/s & bswot/s) are given in 512 byte blocks. The number of swap ins and swap outs per second is zero. This indicates that no swapping is occurring.

If you do see that swapping is occurring, reduce the memory size given to SQL Server (run the **sp_configure memory** command in isql), and check again until you get no swapping.

UnixWare PSE Pages

To make the most efficient use of system memory on an Intel Pentium processor you should take advantage of PSE memory. PSE memory allocates contiguous memory with a 4 Mbyte page size that can be used for shared memory. The larger page size allows more efficient page lookup on Intel Pentium processors. PSE pages are allocated in 4 Mbyte pages and are configured with the kernel parameter **PSE_PHYSMEM**. **PSE_PHYSMEM** is specified in bytes but 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.

UnixWare User Capacity Parameters

There are several areas of tuning necessary based on the number of users that you want to connect. The Sybase parameter, **user connections**, is explained in the section ‘System 10 Server Configuration and Tuning Parameters’. Here we describe the two OS parameters that deal with user connections, **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*.

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.

Data Striping with Compaq Fast SCSI-2 Controller vs. Compaq SMART Array Controller

The Compaq SMART Array Controller has a great advantage over the Compaq Fast SCSI-2 in that it can do hardware striping of data in 16KB blocks on your drives. The best way to illustrate this advantage is with an example. Let’s assume you have a system with 5-2GB drives. The first two drives are mirrored and hold the OS and log files only. The other 3 drives hold data only and are configured for ‘No Fault Tolerance’. Let’s look at how each controller handles data striping in this scenario.

Fast SCSI-2 Controller Example

The SCSI-2 Controller requires that each of the drives be configured as a separate EISA logical volume. For the example, that means you have 5 logical volumes total. In order to stripe data on the 3 data drives, you must use software striping. This is done during Sybase’s **create database** command. You will first need to use Sybase’s **disk init** to initialize a device for each of the 3 logical data volumes. Then use those devices in **create database** to stripe the data by using fragments. The following gives an example:

Log. 1 Log.2 Log. 3 Log.4 Log.5 (Logical Volumes 1-5)

Log & OS Data: dev1 dev2 dev3

```
ISQL

/* Initialize the three devices you have made partitions for through UnixWare utilities */

1> disk init name = “dev1”,

2>        physname = “/dev/syblink/dev1”,

3>        vdevno = 5,

4>        size = 921600                    /* 1 million-2KB pages is max size for disk init.

                                           That is 1953.125 MB which is < 2GB */

5> go


1> disk init name = “dev2”,

2>        physname = “/dev/syblink/dev2”,

3>        vdevno = 6,

4>        size = 921600

5> go

1> disk init name = “dev3”,

2>        physname = “/dev/syblink/dev3”,
3>        vdevno = 7,
4>        size = 921600
5> go

/* Create the database with fragments, striping across  the three devices */
1> create database mydb on
2>        dev1 = 180, dev2 = 180, dev3 = 180,
3>        dev1 = 180, dev2 = 180, dev3 = 180,
4>        dev1 = 180, dev2 = 180, dev3 = 180,
5>        dev1 = 180, dev2 = 180, dev3 = 180,
6>        dev1 = 180, dev2 = 180, dev3 = 180,
7>        dev1 = 180, dev2 = 180, dev3 = 180,
8>        dev1 = 180, dev2 = 180, dev3 = 180,
9>        dev1 = 180, dev2 = 180, dev3 = 180,
10>       dev1 = 180, dev2 = 180, dev3 = 180,
11>       dev1 = 180, dev2 = 180, dev3 = 180
...
> go

/* Create one segment spanning all three devices in order to load table data onto that segment.
That way the data will span across all three devices. */
1> exec sp_addsegment Seg1, dev1
2> go
1> exec sp_extendsegment Seg1, dev2
2> go
1> exec sp_extendsegment Seg1, dev3
2> go
```

This method works, but is more tedious and complicated than the next example with the SMART Array Controller.

Compaq SMART Array Controller Example

The SMART Controller allows you to create a logical volume that spans up to 14 drives. The UnixWare size limit for a raw device partition is 2GB. That means you would need to have 14 raw partitions defined if you want to use all the drive capacity of 14-2GB drive system. A good way to divide the raw devices then is to put 7 partitions on each logical volume.

In our example with 3 data drives, we can make one logical volume. Now we can use the data striping capability of the SMART Controller to do the work for us.

Logical vol. 1 Logical vol. 2

Log & OS Data

```
ISQL
/*  Initialize three devices you have made partitions for through UnixWare utilities.
If your data size will be less than 2GB, you only need one 2GB partition and one device. */
1> disk init name = “dev1”,
2>          physname = “/dev/syblink/dev1”,
3>          vdevno = 5,
4>          size = 921600
5> go
1> disk init name = “dev2”,
2>          physname = “/dev/syblink/dev2”,
3>          vdevno = 6,
4>          size = 921600
5> go
1> disk init name = “dev3”,
2>          physname = “/dev/syblink/dev3”,
3>          vdevno = 7,
4>          size = 921600
5> go

/* Now the create database command is simple.  The SMART Controller will stripe your data
across the devices in 16KB fragments.  This is much better than the previous example. */
1> create database mydb on
2>          dev1 = 1800, dev2 = 1800, dev3 = 1800
...
> go

/* Now create one segment spanning all three devices in order to load table data onto that
segment. That way the data will span across all three devices. */
1> exec sp_addsegment Seg1, dev1
2> go
```

```
1> exec sp_extendsegment Seg1, dev2

2> go

1> exec sp_extendsegment Seg1, dev3

2> go
```

Using the SMART Array Controller's hardware striping capabilities increases performance and makes it easier for the System Administrator to maintain the database.

If you have more than one logical volume and you need to spread a table across volumes to get faster I/O performance, you will have to use Sybase software striping, through the **create database** command, as shown above. Then you will need to create and extend a segment to span a device on each of the logical volumes on which you want to spread the data. That way the table data will be loaded sequentially across volumes, according to your fragment sizes. The SMART Controller still uses its hardware striping capabilities within each logical volume during creation of the database and loading of the table data. Each drive in a logical volume will get 16KB of data alternately and sequentially, until the total fragment size that you defined in **create database** is reached, or during data load, until the all data for the table is loaded. Therefore, by using a combination of SMART Controller hardware striping and Sybase software striping, the data is striped evenly across drives in a volume, as well as across volumes.

System 10 Server Configuration and Tuning Parameters

sp_configure

Usually the default settings for the **sp_configure** parameters are sufficient for running SQL Server. If you do need to alter these parameters do so with care. Some of the **sp_configure** values take effect dynamically as you change them and others will require you to stop and restart SQL Server to take effect. If SQL Server has determined that you are setting a value outside of its pre-determined normal guidelines, you may have to issue a *reconfigure with override* followed by a *checkpoint* for the new value to be set. In all cases, check via ISQL to insure that the **sp_configure** run_value and config_value of the items you have changed match before letting additional users onto the system.

It is possible to configure these parameters such that you will be unable to start SQL Server. If this does happen you can reset ALL of the **sp_configure** parameters to their default values via the **buildmaster** routine found in the Sybase's *bin* directory. The -d option is followed by the full path name of your master device. For example:

```
buildmaster  -d /dev/master  -r
```

You may not be able to start your database if the number of devices your database requires exceeds the default number of 10 devices. If this is the case, use the command line switch -ycnvdisks=nn, where nn is the number of database devices you need. For example if your database utilizes 25 devices the reset command line will look like:

```
buildmaster -d /dev/master  -ycnvdisks=25
```

To get a listing of all the **buildmaster** parameters and their current values, do:

```
buildmaster -d /dev/master -yall
```

sp_configure Recovery Interval

This setting should be left at its default value 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. During a checkpoint, SQL Servers' data cache area is forcibly written to disk, during which time all other database activity is suspended. Immediately after a checkpoint, user response times will be slightly faster than normal until the data cache area becomes filled. Once this area is filled, user response times will slow down to 'normal' levels due to necessary disk access and memory management.

sp_configure User Connections

This setting should be set at the minimum you can run with. 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.

Note: For each user connection SQL Server allocates approximately 40KB for the users' stack area and an additional 5-10KB for the users' procedure cache. You may have to readjust the **sp_configure memory** value depending upon the number of user connections your environment requires.

sp_configure 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** 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 "sp_configure 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 **sp_configure 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, use the **buildmaster** routine with the -r option as described above to reset the value and restart SQL Server.

sp_configure Procedure Cache

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 to compile queries and during the creation of stored procedures.

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 Novell. To achieve an optimal configuration, there are several factors to include. Tuning the application, tuning the hardware, tuning the OS, and tuning the network 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 2.0, 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: 32500-01-1000-02

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

- □ SCO Unix
- □ Microsoft Windows NT
- □ IBM OS/2
- □ Novell NetWare
- □ Novell UnixWare
- □ Other:

RDBMS

- □ Microsoft SQL Server
- □ Sybase System 10
- □ Oracle 7
- □ Other:

Processing Type

- □ On-Line Transaction Processing
- □ Decision Support
- □ Batch Processing
- □ Other:

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

Topic

- □ Configuration and Tuning
- □ Capacity Planning
- □ Integration Information
- □ Competitive Analysis
- □ Systems Management
- □ Other:

Operating Systems

- □ Microsoft Windows NT
- □ Novell NetWare
- □ IBM OS/2
- □ Novell UnixWare
- □ SCO Unix
- □ Other:

RDBMS

- □ Microsoft SQL Server
- □ Sybase System 10
- □ Oracle 7
- □ Other:

Additional Comments:

Return to:

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