Paper 277-26

Windows NT Server Configuration and Tuning for Optimal Server Performance

Susan E. Davis, Compaq Computer Corp., Cary, NC
Carl E. Ralston, Compaq Computer Corp., Cary, NC

ABSTRACT

This paper focuses on how to configure the CPU, memory and input/output subsystems of your server for optimal SAS® software performance. Configuration of the I/O subsystem, RAID tradeoffs, and various Windows NT and SAS options will be discussed that effect SAS performance and throughput. Proper configuration of your Windows NT server can be the difference between acceptable and unacceptable performance for your SAS applications.

INTRODUCTION

When you install a new server or upgrading an existing one, your initial hardware configuration is someone's best guess at what you need based on the information available. While experience can make that "guess" more accurate; it is based on some assumptions about how many jobs, how much data, and the type of processing being done. It is important to monitor a system to see how those assumptions measure up against reality.

A discussion of the Windows NT Performance Monitor tool is beyond the scope of this paper, but it is an essential tool for monitoring and tuning your system. The general strategy is to use Performance Monitor to identify a bottleneck and then work to relieve the bottleneck. The book "Optimizing Windows NT" is an excellent guide on how to use Performance Monitor and tune an NT system.

The focus of this paper is not in identifying the bottlenecks, but how to configure your system to minimize them. The adage "Jack of all trades and master of none" can hold true for computers as well as people. Different applications place different demands on a system and optimizing the performance of one application may degrade performance of the other. Tuning such a system is difficult and neither application performs as well as it might be able to. It is strongly recommended that SAS applications be placed on their own server without any other major applications.

The major performance characteristics are determined by your system's hardware configuration. The decisions you make when you purchase your system are the most important in determining how your system performs. System tuning can help identify problems and alleviate bottlenecks, but cannot compensate for poorly configured system.

SELECTING CPUS

SAS is often a CPU intensive application. CPU technology is constantly evolving with CPU speeds increasing frequently. It is recommended that you get the fastest CPU available to you. The Compaq ProLiant server family has 700MHz to 1GHz Pentium III and Intel Xeon processors available in configurations from one to thirty-two CPUs.

SAS V8.x is a single threaded application, so a single SAS job will use the resources of only one CPU. While SAS version 8 introduced (MP) CONNECT to allow a single job to exploit multiple CPUs on a server, this must be done programmatically. Unless an application is designed specifically to use the features of (MP) CONNECT, adding multiple CPUs will not improve an application's performance. Multiple CPUs will however improve performance when multiple SAS jobs are running simultaneously. The number of simultaneous SAS jobs is one of the major determining factors in how many CPUs should be configured on a server.

A CPU bottleneck can mask both memory and I/O bottlenecks. Analyze and tune both the memory and I/O subsystems to verify that the CPU is really the bottleneck. There are three basic strategies for dealing with a CPU bound system.

1. Make use of NT scheduling tools to move work to hours when the server is under utilized.
2. Offload work onto other servers.
3. Upgrade your server with faster and/or additional CPUs.

USING MEMORY EFFICIENTLY

One of the adages of performance tuning for computers is that "the fastest I/O is the one you don't do." No matter how well you tune your I/O subsystem, writing to disk is always much slower than working in memory. Ensuring that you have enough memory, and that you use that memory efficiently can greatly improve performance of you SAS server.

When discussing memory it is important to distinguish between virtual address space and physical memory. Windows NT's 32-bit architecture allows for 4GB of virtual address space. By default, 2GB is reserved for the system and each process is allowed to use up to 2GB of private virtual address space. (Discussion of the /3GB boot switch which allows up to 3GB of virtual address space to be visible to user processes is beyond the scope of this paper.) With 2GB of virtual address space each process acts as if it has 2GB of physical memory available for its use. Behind the scenes Windows NT manages this virtual address space with a combination of physical memory and disk space using the page file (pagefile.sys). Refer to the I/O section of this paper for further discussion on configuring the page file.

When the virtual memory requirements of all of the processes on the server exceed the amount of physical memory, NT makes use of the page file, reading from it and writing to it in order to keep track of all the virtual address space. This process is called paging, and it is less efficient than if the pages were already available in physical memory. Minimizing system paging is key to optimizing server performance.

How much memory should a SAS server have? That of course, depends on which SAS procedures you will be using and how much data you have. A good rule of thumb is to have at least 128MB of RAM for each concurrent SAS job. In addition to the memory for SAS, you should add an additional 512MB to your server for Windows NT overhead like the file system cache. Remember that these are starting numbers, you will need to round up or down depending on your SAS procedures, server capabilities and budget. Refer to the SAS options section of this paper for discussion on SAS system options which impact memory usage.

Once you have decided how much memory your system needs, you need to be aware of the memory capacities of your server. Most Compaq servers have between four and sixteen memory slots, arranged into memory banks. All slots in a single memory bank must be populated at the same time with the same size and speed of memory. For example if you were to order 1GB of RAM for a Proliant 8000 it would come as two 512MB DIMMs, for a Proliant ML570 it would be four 256MB DIMMs, and for a Proliant DL380 it would be one 1024MB DIMM.

Memory is often one of the more expensive components in your server. Plan your purchases to preserve your memory investment. For example, if you want to purchase a Proliant DL380 with 1GB of RAM, there are several ways to put 1GB of RAM in the system. The Proliant DL380 has four memory slots arranged as four memory banks and comes with 128MB RAM standard. The least expensive way to purchase 1GB of RAM, would be to buy one 128MB DIMM, one 256MB DIMM, and one 512MB DIMM. The problem is that this configuration will fill all of the memory slots on the machine. If you determine you need more memory at a later date, you will need to replace one of the DIMMs purchased in your original configuration. An alternative configuration would be to purchase two 512MB DIMMs or one 1024MB DIMM. Although the initial investment would be higher, there would be memory slots open for future upgrades.

I/O SUBSYSTEM CONFIGURATION

The key to configuring your I/O subsystem is balance. Balancing the I/O is a two-part process. The first part is related to the hardware, the second is related to how you configure the file system and distribute files across the volumes. In the simplest case, if your server had only one 36.4GB disk, all I/O activity would have to wait for that disk, and performance will be limited by that disk's physical ability to transfer data. If you replaced that one 36.4GB disk with four 9.1GB disks, you have the potential to have four disks working simultaneously to service I/O requests. However, having four disks available doesn't help you if you still put all of your data on one disk. The best hardware in the world cannot compensate for poor file placement and distribution across disks. The file system portion of the paper discusses file placement.

Hardware

Spreading the I/O over multiple disks is actually more difficult when your total storage requirements are small (<100GB). In this case your storage needs can typically be met with internal storage. The rule of thumb here is to fill all of the internal media bays with the smallest sized disks that will allow you to meet your total capacity needs. For example the Proliant DL380 can hold up to six internal drives (with the optional drive cage). If you needed 50GB of space, the optimal configuration would have six 9.1GB drives. It is acceptable to mix disk sizes, however if you are going to use RAID, you need to make sure that all of the disks in a RAID volume (discussed in the File System section) are the same size.

When you move beyond the internal capacity of your server, usually the next step is direct attached storage. The disks sit in external storage enclosures and communicate with the server via SCSI controllers placed in the server's PCI expansion slots. The Compaq 4200 and 4300 series of disk enclosure hold fourteen 1" universal drives (9.1GB, 18.2GB or 36.4GB) each. If you balance your data over the fourteen disks in an enclosure, they could flood a single SCSI bus, even at Ultra-3 rates (160MB/s). The shelves come in a split bus variety that will allow you to spread the load from all of the disks over two SCSI buses. This is the recommended configuration for your external shelves. SCSI controllers can have one to four I/O channels. You want to have enough channels coming into your controllers to accommodate the SCSI buses coming out of your shelves. Each dual bus shelf requires two SCSI channels. Make sure your disks, shelves and I/O controller are all using the same SCSI version. While the Ultra-3 devices will inter operate with Ultra-2 devices, data transfers will only take place at Ultra-2 (80MB/s) speeds. Use the fastest disks (currently 15,000 RPM) available.

Different server models have varying numbers and types of PCI buses and slots. Often several PCI slots share a single PCI bus. For example the Proliant DL580 has 6 PCI slots spread over three PCI buses. There are two 64-bit/66MHz slots, three 64-bit/33MHz slots and one 32-bit/33MHz slot. Be sure to research the PCI configuration of your server, and take that into account when you are planning your I/O subsystem. If you have multiple controllers, spread them across multiple PCI buses if possible. Avoid placing a 33MHz controller on in a 66MHz slot, as this will cause all other adapters on the same PCI bus to operate at 33MHz.

SCSI controllers come in a variety of types, Ultra-3 or Ultra-2, 64-bit or 32-bit, 66MHz or 33MHz and you should get the best controller your server and budget allow. SCSI controllers can also be categorized as "dumb" or "smart." Smart controllers, such as those in Compaq's SmartArray family, include hardware RAID engines, while dumb controllers do not. RAID will be discussed in more detail in the file system portion of the paper, but it can be implemented via software or hardware. Software RAID, used with dumb controllers, is implemented by the operating system and uses system CPU and other resources. Hardware RAID is implemented on the SCSI controller itself; the controller has its own specially optimized CPU and memory on the board. Hardware RAID is preferable to software RAID.

When you have large volumes of data (>1TB), a storage area network (SAN) may be the most appropriate method for storing your data. There is a certain amount of infrastructure (external smart controllers, fibre channel switches, software, etc.) that is needed to implement a SAN, but your total storage needs may be large enough so that the cost of that infrastructure is a small portion of the total system cost. A SAN allows you to share disks between systems and platforms (NT, UNIX, and OpenVMS), so is most cost effective in an environment where the storage costs can be shared. A SAN separates the data from the servers and provides consolidated storage management in an environment with improved reliability and performance. A detailed discussion of SANs is beyond the scope of this paper, but SANs are an excellent option in an environment with rapidly growing data requirements.

If you are configuring a server for a SAN environment, consider using direct attached storage for server specific files. Server specific files such as the page file or SASWORK are not useful to other machines. By placing directly connecting them to the server you free up SAN resources.

File System and File Distribution

The first step to building an efficient file system is to spread your workload over multiple volumes. The operating system, system page file, and SASWORK areas should all be on separate volumes from your data. Consider separate volumes for input and output data areas. Creating separate volumes allows you to use the best variety of RAID for each data type and helps to spread the I/O workload among multiple physical disks.

Understanding the type of data on each volume and how it is going to be used will help you to pick the most appropriate type of RAID (Redundant Array of Inexpensive Disks). Because of the system overhead associated with software RAID implementations, hardware RAID is preferable. Not all of the RAID types listed in Table 1 are available on all implementations. RAID-ADG is available only on the SmartArray 5300 family of controllers. Table 1 lists the types of RAID and some of the advantages and disadvantages of each.

Table 1: RAID Types

RAID Level	Definition	Pros & Cons
RAID-0	Data Striping	Increased performance, but no fault tolerance. All data on volume is lost if a single disk fails.
RAID-1	Data Mirroring	High availability and good performance. Expensive to implement because it requires twice as many physical disks.
RAID-1+0	Mirrored Striped	Increased performance and high availability. Expensive to implement because requires twice as many physical disks.
RAID-5	Striped w/Parity	Improved read performance, but slower write performance. Protects against a single disk failure.
RAID-ADG	Striped w/Double Parity	Improved read performance, but poor write performance. Protects against the failure of two drives.

Your system volume is a good candidate for RAID-1. Data mirroring allows system information to be preserved in the event of a disk failure. Read performance is improved because data can be read from either disk. Since the system volume is small the cost of having duplicate data is small.

When you have a large system (4 or more CPUs) and many concurrent SAS jobs, you should consider putting your SAS software (!sasroot) on its own volume. Performance benefits can be gained by separating your SAS related I/O from other system I/O. Like your system volume, the SAS volume is a good candidate for RAID-1.

As discussed in the memory section above, Windows NT uses the page file to manage the virtual address space. This file is often heavily used, with both read and write activity. By creating a volume with multiple physical drives the I/O activity can be spread across several spindles. Because this information is very volatile, it should be configured with RAID-0. Do not put your page file on a RAID-5 volume. Use RAID-1 (or 1+0) if you need to configure your page file for high availability. Ideally you should not put any other data on the volume with your page file, however if you must, then put data that is rarely used (i.e. archive).

The Windows NT page file is configured using the System icon in the Control Panel. The Performance tab allows you to define the size and location of your page file. Windows allows you to set a minimum and maximum page file size. Always set the minimum and maximum to the same size. If you allow the page file to grow and shrink according to system demands, it will become fragmented and performance will degrade. By default Windows will recommend a paging file size of your physical RAM + 11MB. You should change this value to at least 150 percent of your physical RAM.

SASWORK is another volume that will benefit from RAID-0. Files are created in SASWORK as a SAS job runs and then deleted at the end of the job. By default the SAS work area is on the same volume as your SAS installation, to change this update the Sasv8.cfg file to point to your RAID-0 volume. By default all jobs will use the same SAS work area. Personalized configuration files or command line options when SAS is invoked will override this default behavior. If you have a large number of simultaneous jobs you may be able to further spread the I/O load over multiple volumes.

RAID-5 is appropriate for permanent data. Raw data files, SAS programs, and permanent SAS libraries and catalogs are good candidates for a RAID-5 volume. Consider having multiple RAID-5 volumes instead of a single large volume. A SAS job that reads raw data in from one volume and writes a permanent SAS data set out to another volume will perform better than one that reads and writes the data to the same volume.

One of the parameters required when setting up RAID-0, RAID-5 or RAID-ADG volumes is the stripe size. When data is written to a RAID volume, the data is broken into logically contiguous chunks and these chunks of data are placed onto each physical drive in the volume. The size of this chunk of data placed on each physical drive is referred to as the stripe size. The ideal stripe size for SAS is the maximum I/O transfer size for your device. Depending on your device driver the maximum I/O size is either 32KB or 64KB. The Compaq SmartArray series of controllers support I/O transfers up to 64KB.

Once you have created your RAID volumes, you must format them. One of the format parameters is cluster size. A cluster is the smallest unit of file allocation. Files are made up of one or more clusters. In Windows NT the cluster size can vary from 512 bytes to 64KB (NTFS compression is only supported if cluster size is 4KB or less). A 100 byte file will always use one cluster, so the larger the cluster size, the more space is "wasted." The downside of a small cluster size is there is more overhead to manage all of these little file segments. Generally the default cluster size is acceptable, but you want to make sure that SAS data areas and SASWORK have a cluster size of 4096 bytes.

Once you create your volumes and set up your file system, all of the work is not done. There are several commercially available disk defragmentation utilities. Buy one and use it. File fragmentation is a problem because additional file system activity must take place to access a file that is stored in multiple, noncontiguous locations on a volume. When you defragment a volume, all the files on the volume are rearranged so that each file is in one contiguous extent. Schedule your defrag job to run periodically during non-peak times as performance is severely degraded during the defragmentation process. If you cannot defrag your entire system during a single non-peak window, then create multiple defragmentation jobs and run them over several days.

SAS OPTIONS

SAS system options can be used to monitor and optimize job performance. Most SAS system options can be specified either in the configuration file (sasv8.cfg), on the command line when SAS is invoked, or using an OPTIONS statement. MEMSIZE and SGIO are exceptions; they can only be specified in the configuration file or on the command line. BUFSIZE and BUFNO are also data set options and are available in DATA steps and procedures.

FULLSTIMER Option

The SAS FULLSTIMER system option specifies that the SAS system records the list of computer resources, which were used for each DATA step and procedure, in the SAS log. This option is useful for identifying which portions of your SAS job should be targeted for optimization and to monitor the effect of changes you make. FULLSTIMER reports the real or elapsed time and the CPU time for each step. When the sum of the CPU times is less than the elapsed time, then the DATA step or procedure was waiting for one or more system resources. Where the job is the only job on the machine, this typically means it was waiting for I/O. If there are other jobs on the machine, it may have been waiting for some other system resource. Use the Window's NT Performance Monitor Tool to identify the bottleneck.

Memory Size Option

The SAS MEMSIZE system option controls the maximum amount of virtual address space available to a SAS process. The default value of 0 allows a SAS job to use as much memory as it needs up to the 2GB limit placed on it by NT. The appropriate default value for MEMSIZE (in sasv8.cfg) is dependant on what you are doing, but a starting point is the total amount of physical RAM divided by the number of concurrent SAS jobs. This won't hold true for every SAS job as there are several SAS procedures, such as PROC MDDB, will fail to run successfully unless there is sufficient virtual address space.

Sort Size Option

The SORTSIZE option determines the amount of virtual address space available to the SORT procedure. If the amount of space required for sorting is greater than the SORTSIZE value, the SORT procedure creates temporary utility files in the SASWORK directory. Application speed can be increased considerably if the data sets can be sorted in virtual address space instead of slower disk-based sorts. However, setting SORTSIZE to a value greater than the amount of available physical memory can cause system paging. In this case, setting SORTSIZE to a smaller value will allow SAS to utilize the temporary utility files in the SASWORK directory. The default value for SORTSIZE in Windows NT is 2MB. This value is very low. A good value for SORTSIZE is MEMSIZE - 16MB or a minimum of 32MB.

The impact of sort size can be dramatic. In a simple test sorting a 1.1GB file the file took 5:56 minutes to sort using the default SORTSIZE value of 2MB and only 2:26 minutes to sort using a SORTSIZE value of 2GB-16MB (the server had 4GB of RAM). When SORTSIZE was set to 2MB, I/O increased dramatically as SAS used temporary data sets to complete the sort.

Buffer Size Option

The BUFSIZE option determines the size of the input/output buffer that SAS uses for transferring data during processing. This is the minimum number of bytes of data that SAS moves between external storage and memory in one I/O operation. BUFSIZE is a permanent attribute of a SAS dataset and is determined when the dataset is created. Increasing BUFSIZE can improve the elapsed time by reducing the number of times SAS has to read from or write to disk. Increasing BUFSIZE also increases memory consumption, which may reduce overall performance. In some tests, larger (greater than 65532 bytes) BUFSIZE increased elapsed time during memory intensive tasks, such as sorting.

In the ideal environment, each SAS I/O will involve all of the physical disks in a RAID volume. Figure 1 shows how to calculate BUFSIZE to make this happen.

```
BUFSIZE = Stripe Size * Number of Members
```

Stripe Size = stripe size of RAID volume (See I/O configuration section on determining stripe size)

RAID	Number Of Members
No RAID	1
RAID-0	Number of physical disks in volume
RAID-1	Number of physical disks / 2
RAID-1+0	Number of physical disks / 2
RAID-5	Number of physical disks - 1
RAID-ADG	Number of physical disks - 2

Figure 1: Calculating BUFSIZE

Number of Buffers Option

The BUFNO option determines the number of input/output buffers that SAS uses for transferring data during processing. Increasing BUFNO may improve performance by reducing I/O. However, like BUFSIZE this comes at the cost of increased memory usage. Increasing BUFNO may or may not reduce elapsed time. Experiments have shown that setting BUFNO to a value greater than 2 does not improve performance. The default value is 1.

Scatter Gather I/O Option

Under Windows NT Service Pack 4 or higher, the SGIO system option is available. When SGIO is turned on, SAS I/O activity bypasses the NT file cache and reads/writes directly between the disk and the SAS buffers. Data normally moves from disk to the NT file cache and then into the SAS buffer. SGIO saves time by removing one of those transfers. SGIO works in conjunction with BUFNO to improve I/O performance. SGIO only effects I/O that is opened in input or output mode.

Utility Buffer Size Option

The UBUFSIZE option determines the utility buffer size of the input/output buffer that SAS uses for accessing the temporary files created when SAS is unable to sort a data set in memory. Increasing UBUFSIZE may improve the SORT procedure's performance by reducing I/O. The appropriate UBUFSIZE can be calculated using the same formula in Figure 1, using the SASWORK volume for your calculations. As with BUFSIZE, the I/O improvement may be offset by performance degradation from increased memory usage. Increasing UBUFSIZE may or may not reduce elapsed time.

Number of Utility Buffers Option

The UBUFNO option determines the number of utility buffers that SAS uses for accessing the temporary files created when SAS is unable to sort a data set in memory. Increasing UBUFNO may improve the SORT procedure's performance by reducing I/O. However, like the other buffer related parameters this comes at the cost of increased memory usage. Increasing UBUFNO may or may not reduce elapsed time.

CONCLUSION

Taking the time to plan your Windows NT SAS server's hardware configuration and file distribution are well worth the effort. These activities, which take place before SAS is installed, are important determinants on how your system will perform. As data requirements grow, planning the I/O subsystem and file distribution becomes even more important. Modifying them after an I/O bottleneck has been identified is more difficult than initially setting them up correctly.

Once your system is up and running, tuning activities are ongoing. Regular jobs to defragment your disks will have a positive impact on performance. Testing the impact of system options such as MEMSIZE, SORTSIZE and BUFSIZE in your environment will determine which options provide the best performance improvements. Ongoing monitoring with the Windows NT Performance Monitor tool will help you correctly identify the impact of your SAS changes and which system resources are causing bottlenecks.

REFERENCES

Daily, Sean K. (1998), Optimizing Windows NT, Foster City, CA: IDG Books Worldwide, Inc.

"PCI Bus Balancing and Optimization on Compaq Proliant Servers," Compaq White Paper, March, 1998, Doc ID ECG073/0398

"PCI Bus Numbering in a Microsoft Windows NT Environment," Compaq White Paper, April, 1998, Doc ID ECG024/0298

Compaq White Paper," SAS Performance Test Suite With Data Warehouse Emphasis on Microsoft NT V1.0 Solutions Guide," Compaq White Paper, September, 2000, Doc ID 13CW-0900A-WWEN

www.compaq.com, all products and specifications mentioned in this paper were current as of January 23, 2001

ACKNOWLEDGEMENTS

Thanks to Dr. Leigh Ihnen, Manager of Numerical Architecture and Performance, Application Division R&D at the SAS Institute for reviewing this paper.

TRADEMARKS

Compaq, ProLiant and the Compaq logo are registered with the United States Patent and Trademark Office. Microsoft, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Intel, Pentium and Pentium® III Xeon are trademarks and/or registered trademarks of Intel Corporation. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Susan E. Davis
Compaq Computer Corporation
103 Pocono Lane
Cary, NC 27513-5316
Work Phone: (919) 531-5647
Fax: (919) 677-4444
Email: Susan.Davis@Sas.com
Web: http://www.compaq.com/partners/sas/

Carl E. Ralston
Compaq Computer Corporation
106 Clear Sky Court
Cary, NC 27513
Work Phone: (919) 531-5905
Fax: (919) 677-4444
Email: Carl.Ralston@Sas.com
Web: http://www.compaq.com/partners/sas/