

Performance Tuning Of SAS® Under Microsoft Windows™

Paul L. Marsh, N. C. State University, Raleigh, NC 27695-8203

Ever since SAS first appeared on microcomputers, users have been concerned about maximal affordable performance. Much of this is due to the fact that SAS is very resource-intensive; SAS will tax any machine upon which it runs.

Perusal of the SUGI proceedings since 1983 and the SAS-L listserv archives leads to many tips on tuning SAS. This paper highlights those tips and intermixes them with the author's personal experience. Performance tuning (and therefore, benchmarking) is an extremely complex and confusing task. The ideal SAS set-up is not only dependent upon the system hardware and software but also upon the types of tasks that the user performs with SAS. Due to the confounding of these performance factors, it is impossible to define a set of performance guidelines applicable to all users. Instead, this paper will list recommendations that could be followed by most users. Mileage may vary; the reader should concentrate on the themes presented herein and be prepared to do some self-testing.

Areas for Tuning

An inside-out approach that must be followed when tuning SAS under Microsoft Windows (indeed, on any computing platform). First, the user must address hardware considerations. Next, the link between the hardware and the application software must be tuned. For this paper, that means DOS and how Windows sits atop DOS. After those matters are resolved, the user must tweak the SAS configuration (usually through the CONFIG.SAS and AUTOEXEC.SAS files). These first four layers of performance tuning often need addressing only once during the lifetime of a machine; thus the user should be willing to spend some time investigating alternatives. Finally, the user should learn as much as possible about the SAS language itself. Indeed, the most dramatic performance enhancements are often the result of the user writing better SAS code.

Hardware

Let's assume you are a new Microsoft Windows SAS user and you wish to purchase a new computer specifically for running SAS. Fortunately, SAS is not a unique software package; any system that runs SAS fairly well will also perform fairly well for most other

software packages. A good starting point is to examine the hardware requirements listed in Table 1.

Whenever a new version of SAS goes into production, it seems that the user community cries out that SAS is demanding at or beyond the state of the art. SAS under Microsoft Windows was no exception. It is good, though, when software pushes the envelope of hardware development. CPU performance continues to double approximately every 18 months. Dramatic price drops have coincided with recent size increments for hard disk drives. RAM memory has stayed at affordable levels. A \$2000 machine (circa March 1995) can now run SAS very nicely for most users. Moreover, in just three years time that machine will seem very outdated.

Table 1 mainly deals with minimal requirements for the SAS System. But what are the recommended amounts? A good place to start is to examine Technical Notes TS330 and TS423 available from SAS Institute. Included in those notes are the following recommendations:

* Buy the fastest processor (CPU) your budget will allow. CPU speed is measured in megahertz(mz). The order of performance is (move down the columns then to the right):

Pentium 90mz	80486DX2 33/66mz	80386DX 40mz
Pentium 66mz	80386DX2 50mz	80486DX 25mz
Pentium 60mz	80486DX2 25/50mz	80386DX 33mz
80486DX/4 100mz	80486DX 40 mz	80386DX25mz
80486DX/4 75mz	80486DX 33mz	80386DX 16 mz

Note 1: The current rapid hardware evolution has produced choices not listed above. Extrapolate as appropriate.

Note 2: Differences between any adjacent CPUs in list may only be slight.

* Insist on CPU cache memory. While faster machines seem to benefit most from large CPU caches, more than 256k doesn't appear to boost performance significantly. Some literature articles indicate diminishing returns beyond 64k.

* Buy the best hard disk drive your budget will allow. SAS is very disk intensive. Some things to consider:

Table 1.
Requirements for SAS Under Microsoft Windows

	Release 6.08 System 5.0 or higher	Release 6.10 (TS019) 5.0 or higher
DOS version	3.1	3.1 or Windows for Workgroups 3.11*
Windows version(s)	16mhz 386	33mhz 386
Slowest CPU	highly recommended	required
Math coprocessor		
Memory		
minimum	6 megabytes	8 megabytes
mid-range use	8 megabytes	10 megabytes
high performance	12 or more megabytes	12 or more megabytes

*Windows or WIN32s Operating System extension 1.15.111 later required with either Windows 3.1 or Windows for Workgroups. The extension is provided with the SAS distribution software.

-Size No matter how much you plan, you'll never have enough hard disk space. Not only do your applications and data need room on the disk, Windows and SAS also use the disk for temporary files and memory. So look at it this way: if you are even 1 megabyte short of having enough disk space, you will constantly have to deal with disk management and will rue the day you cut corners on this item.

-Transfer Rate This is the rate at which the hard disk will transfer data measured in Megabytes/second (mb/sec). Higher values are better.

-Access Time An average of how fast the hard disk can find the requested data. Lower values are better.

Note 3: Both technical notes state SCSI drives tend to be faster than their counterparts. Recent advancements in hard drives have produced a new type called Enhanced IDE (EIDE). These drives tend to have good performance at affordable prices.

Note 4: Larger drives tend to be faster drives. However, do not think that bigger is necessarily the route to follow. Since the SAS system is highly configurable, it is often times possible to get superior performance with multiple slower, smaller physical drives (not partitions!) than a single large drive. See the section "Configuring SAS" below.

- * Buy a large amount of RAM memory. How much is enough? You should see dramatic speed increases as you add memory to the 12 or 16 megabyte levels. Speed increases will be less noticeable thereafter, depending on how you utilize the additional memory beyond 16 megabytes. However, you will be afforded some interesting DOS, Windows, and SAS systems options as you break beyond 16 megabytes. Some users could benefit substantially from larger amounts of memory.

- * Consider a caching hard disk controller. Caches save frequently requested data in memory. Since SAS is very disk-intensive, substantial increases in performance are typically seen when caching is utilized. Hardware caches such as those found on controllers tend to out-perform software caches that utilize RAM memory. However, the memory found on caching controllers tends to be more expensive than standard RAM memory.

Note 5: Users should examine the types of SAS tasks they tend to perform. If tasks tend to be compute-intensive (e.g., large linear models or simulations), budget allocations should swing toward more expensive CPUs. If tasks tend to be more input-output intensive (e.g., reading/writing large amounts of data via DATA steps), expenditures should center more towards hard disks and memory caches.

- * Consider a computer that uses a new technology other than the standard ISA bus. However, these buses tend to cost more (as do their adapters) and dollars could be spent more advantageously on other items in this list.

- * Consider a video accelerator board. Windows applications benefit tremendously because the video board has its own graphics processor and can redraw the screens quicker. This will lighten the load on the main CPU and allow it to attend to other things. Screen-intensive tasks such as spinning a plot with SAS/Insight will benefit greatly from such hardware.

Some of the above options (such as adding RAM memory) should be considered by those users already owning machines. Additionally, it may be possible to upgrade the CPU chip in the existing machine at

a fairly nominal cost. Be aware, though, that a CPU upgrade will only marginally improve performance if none of the other influencing components are also enhanced.

DOS Configuration

After obtaining the best available hardware, we move on to tune DOS. An operating system interfaces all other software to the computer hardware. As such, it must be as efficient as possible. Among the recommendations listed in Technical Note TS423 are as follows:

- * Don't load drivers or terminate and stay ready programs (TSRs) in the upper memory blocks (the address range of memory between 640 kilobyte and 1 megabyte.) With the introduction of DOS 5.0, DOS users were provided with an upper memory block provider named EMM386.EXE. Unfortunately, if a user exploits the capabilities of EMM386.EXE to provide expanded and additional conventional memory, the user is hampering the operation of Windows. Windows looks at the upper memory address range for free space to optimize its own performance. Fortunately, if users upgrade to DOS 6.x, they will possess a valuable utility, MEMMAKER. MEMMAKER optimizes memory, even to the point of reserving necessary upper memory blocks for Windows.

- * Use the latest HIMEM.SYS driver in your CONFIG.SYS file. If you have upgraded DOS since installing Windows, use the HIMEM.SYS driver that came with DOS.

- * Make sure you have enough file handles. Set the FILES= to 30 or higher in the CONFIG.SYS file. In practice, a value of 50 or larger is often needed.

- * Make sure you have enough buffers. Set the BUFFERS= to 20 if you don't have any disk caching software loaded. If you're using SMARTDRV.EXE or other disk caching software, do not use a value larger than 10.

- * Make sure you have enough stacks. If there is no STACKS option listed in your CONFIG.SYS file, then it defaults to STACKS=9,128. Windows works better if the STACKS are 9,256.

Note 6: Some users may get better performance with STACKS set to 0,0 without loss of Windows stability.

- * If you do not have a hardware caching disk controller, make sure you have a disk caching utility loaded. DOS 5 and 6 come with SMARTDRV.EXE, a solid performer. The size of the disk cache should be no less than 256 kilobytes. Caches larger than 2 megabyte provide little gain at all and sometimes actually perform worse than smaller caches.

- * Check for double-buffering. If you have loaded SMARTDRV.SYS in your CONFIG.SYS file, go to the DOS prompt and type SMARTDRV. A table will be presented that list each drive on your system. The column on the extreme right of the list shows whether or not the drive needs to be double-buffered. If so, consult the SMARTDRV DOS documentation and make the appropriate changes.

Note 7: Double-buffering can slow performance; so, if the table

shows that you don't need it, remove the SMARTDRV.SYS line from your config.sys. Keep the SMARTDRV disk cache active though by leaving the appropriate line in your AUTOEXEC.BAT file.

* Use a hard disk defragmenter regularly. Version 6.x of DOS comes with a program called DEFRAG which can be used to defragment your hard disk drive. Routine compaction of your hard disk ensures near-optimal performance.

Windows Configuration

We are now half-way through the "one-time" tuning steps. We now deal with Microsoft Windows, which is our graphical user interface (GUI). Among the recommendations are:

* Optimize your Windows disk usage. Find the Windows Control Panel icon and select it. Double-click the 386 Enhanced Icon and look at the following items:

- On the right side of the dialog that appears, select Virtual Memory. Click the Change button within that dialog box. Another dialog box will appear. Under the New Size option, you should see a check box and the words Use 32-Bit Disk Access. Windows 3.1 supports the standards introduced by Western Digital for 1003 controllers. By using 32-bit access, Windows 3.1 can achieve tremendous performance gains on machines having this type of disk controller. However, this can cause major disk access problems on some machines. Proceed with caution by backing up all the data on the hard drive just in case you fall into the not-totally-compatible category.

- Within the Change dialog box, check your swap file to see if it is permanent. Having a permanent swap file can increase your performance.

- Make sure your swap file is installed on the fastest hard disk drive you have available on your system.

* Do not use a pattern background. Pattern backgrounds can reduce your Windows System Resources. Use a color background instead.

* Run only one application at a time. Each application you bring up will require resources which won't be available to the other applications that are concurrently running under Windows. Less resources means slower processing in SAS. If you absolutely need to do other things, try to do simple things while SAS is running in the background like word processing or filling in the rows and columns of a spreadsheet.

Note 8. Isn't it a pity that Windows doesn't support multi-tasking better? Otherwise, why don't we just use a task-switcher instead?

SAS Configuration

One of the nicest features of the SAS System is the ability to tailor it for individual needs. It is easy to create an array of CONFIG.SAS and AUTOEXEC.SAS files for specific tasks. Moreover, since SAS options tend to be additive, it is easy to change settings "on-the-fly" within a particular session. Among the recommendations listed in Technical Note TS-330 and found in the SAS-L listserv archives are as follows:

* Upgrade to SAS 6.10. Speed gains are usually observed for both DATA steps and PROCedures.

* If you must use SAS 6.08, use popup menus (as opposed to a menu bar) in the individual SAS windows (e.g., program editor or log) to save Windows resources.

Note 9: Menu bars in individual SAS windows are not available in version 6.10.

* Optimize the statements within CONFIG.SAS file. You may edit this file with any editor that can work with ASCII files (make a back-up first!!). The main statements to examine:

- The load image search paths (which will appear something like this):

```
-PATH (
    !sasroot\core\sasexe
    !SASROOT\BASE\SASEXE
    !SASROOT\AFS\SASEXE
    !SASROOT\GRAPH\SASEXE
    !SASROOT\FSP\SASEXE
    !SASROOT\STAT\SASEXE
)
```

This statement specifies the search order of the modules (PROCedures and other parts of the SAS System). As such, you will want your more heavily used products listed nearer to the top. Since the CORE product really directs all SAS work, it makes sense to place it at the top of the list. Make similar changes to the -SASMSG and -SASHELP statements.

- Change the SORTSIZE option to specify a larger block of memory to sort in. The default is 1m(egabyte). If you have 8 or more megabytes of memory, set this option to 2m (even higher as you go to 12 or more megabytes). The statements to change in the CONFIG.SAS file:

```
/* set the default size of memory to use for sorting */
-sortsiz 1m
```

Note 10: You can also specify this value for a specific sort by setting the appropriate parameter in the PROC SORT statement.

* Set the AutoScroll value to 0 for the LOG window. When set this way, no text will be displayed in the LOG window until after the procedure is complete. Procedures that write to the log a lot will increase in speed.

* Set the AutoScroll value to Max for the OUTPUT window. This delays writing output to the OUTPUT window until the procedure has finished.

Note 11: You can set either of these values through the Options -> Preferences popup menu choices or by typing the AUTOSCROLL # (# is the desired value) on the command line in the appropriate SAS window. Do not forget to issue the WSAVE command to make the change permanent if using the latter method.

* If you are running only production-type SAS jobs, remember to turn off the following options (useful only when debugging or tuning SAS code):

Option	Description
oplist	Show settings of SAS system options in SAS

log echoauto	Show autoexec file in log
stats	Write performance stats to log
fullstats	Write performance stats in expanded form
stimer	Maintain and print timing stats (Don't use this with views)
memrpt	Show memory report
source	Show source code in the log
source2	Show included source code in the log
mprint	Show statements generated by macro facility

* If a second physical disk drive exists, use it for your SAS work files (i.e., temporary SAS datasets and scratch work space). In such cases, it is often a good idea to place all applications on one physical drive (e.g., drive C). Then, assuming the second physical drive is drive D, modify the following line in your CONFIG.SAS file to be:

```
/* Setup the default SAS System user work directory */
-work d:\saswork
```

* If a substantial amount of RAM is available (usually, the memory beyond 16 megabytes not allocated to a disk cache), consider creating a RAM drive. DOS users can easily set one up using RAMDRIVE.SYS. What is the best way to use the RAM drive? The answer to that one is up for debate.

- Some users, particularly those who generally have small datasets, can direct their work directory to the RAM drive (see above suggestion). Great speed enhancements can be achieved. However, since temporary files can accumulate quite rapidly, it is not difficult to fill the RAM drive and thereby crashing SAS.

- A safer use of a RAM drive that would benefit all users would be to copy your SAS DLL's to that drive. Setting up a .BAT file to run SAS will facilitate this type of use. Assuming that SAS is stored on the C drive in the \SAS directory and that the RAM drive is D:, the batch file will look something like this (call it RUNSAS.BAT):

```
copy c:\sas\core\sasdll d:\
sas
```

and the following change would be found in the CONFIG.SAS file:

```
/* Set search path for host dlls */
-pathdll d:\
```

If additional RAM drive space is available, use the LOADLIST option to obtain a listing of the SAS images you use during your SAS sessions. Modify the .BAT file to copy those images to the RAM drive and modify the CONFIG.SAS file to search the drive for modules. For example, if you moved three PROCs to the RAM drive D:, your -PATH statements would now look like:

```
-PATH (
  d:\
  !sasroot\core\sasexe
  !SASROOT\BASE\SASEXE
  !SASROOT\AF\SASEXE
  !SASROOT\GRAPH\SASEXE
  !SASROOT\FSP\SASEXE
  !SASROOT\STAT\SASEXE
)
```

SAS Code

When you reach this point in the inside-out tuning approach, you

have dealt with all systems matters. Your hardware/software environment will be near-optimal. Now comes the bad news. You have to learn how to write good SAS code, particularly if you deal with large datasets or create code for production environments. Fortunately, the SUGI proceedings are just full of articles providing good advice on this topic. Just look for titles addressing the SAS compiler, the program data vector (PDV), or the SAS supervisor.

As a starting point, consider that most SAS code tends to be input/output bound. This tip was recently seen on the SAS-L listserv archive:

- Use LENGTH statement to minimize variable lengths. Particularly useful when dealing with integer-valued variables.

Note 12. Strictly speaking, you should only use numeric variables for those fields requiring arithmetic manipulate. For descriptive or categorical values, even those represented using digits, are best handled as character variables.

- Use DROP/KEEP statements to minimize observation length.

- Only SORT when necessary.

- Read raw data only once; make use of permanent SAS datasets.

- Create multiple datasets from the one dataset if possible.

- Use WHERE statement with procedures. Often no need to create a subset of data with a DATA step.

- Use the _NULL_ dataset if not creating a new SAS dataset.

- Compress some large SAS datasets, but beware that compression can use more space.

- Develop & test programs on a small subset of the data. Use OPTIONS OBS=0; to check the syntax of a program and OPTIONS OBS=n where n is a relatively small value to check the logic of the code.

References

SAS Institute Inc. System

Requirements. The SAS System for Information Delivery Windows, Release 6.08, Cary, NC: SAS Institute, Inc. 1993.

SAS Institute, Inc. System Requirements. The SAS System for Information Delivery Release 6.10 (TS019) (Limited Production), Windows, Cary, NC: SAS Institute, Inc. 1994.

SAS Institute, Inc. Technical Note TS330, Cary, NC: SAS Institute, Inc. 1994.

SAS Institute, Inc. Technical Note TS423, Cary, NC: SAS Institute, Inc. 1995

SAS-L InternetArchives,
gopher://jse.stat.ncsu.edu/11/othergroups/sasl

SAS is a registered trademark of SAS Institute Inc. In the USA and other countries. Windows is a registered trademark of Microsoft. ® indicates USA registration.