Paper 370-2011

Configuration and Tuning Guidelines for SAS®9 in Microsoft Windows Server 2008

Margaret Crevar, SAS Institute Inc., Cary, NC

Updated: August 2011

ABSTRACT

SAS® customers are increasingly moving their SAS®9 application workloads to Windows Server 2008 64-bit, and sometimes they experience less than ideal response times for their application workloads. Working with Microsoft to determine the causes of this non-ideal performance, SAS has discovered several performance issues that are related to how the operating system file cache behaves in Windows Server 2008. These issues can result in limitations to the maximum I/O throughput in Windows Server 2008. And working with Hewlett Packard to get the best performance from the new Intel processors, we uncovered that some of the new energy saving features of these new processors may lead to performance degradations as well. This paper discusses these issues and gives suggestions for configuring the CPU, memory, and I/O subsystems for optimal SAS performance. Proper configuration of your Windows Server 2008 system and your IO subsystem (particularly your storage array) is essential to provide the best performance for your SAS application workloads.

INTRODUCTION

Before you examine the system demands and performance issues that occur with SAS®9 in Windows Server 2008, you should be aware of system provisioning for throughput and performance in SAS®9 in Windows Server 2008. When you install a new system or upgrade an existing one, your initial hardware configuration is someone's best guess at what you need, based on the information that is available. While experience can make that guess more accurate, the initial configuration is based on some assumptions about how many jobs, how much data, and the type of processing being done.

For existing SAS customers, it is important to monitor your system to determine what computer resources are being used the most with your SAS application workloads. Monitoring computer resources is briefly discussed at the end of this paper. We do have about monitoring computer resources on the SAS Customer Support Web site at support.sas.com/rnd/scalability/papers/index.html.

System Demands and Related Performance Issues with SAS®9 in Microsoft Windows Server 2008

When you are configuring a system for a new SAS application, you must understand that different SAS application workloads place different demands on a system. You must also realize that optimizing the performance of one SAS application might degrade the performance of another application. Knowing how to tune each independent application so that they all run efficiently takes significant effort.

With regard to SAS®9 in Windows Server 2008, performance issues occur in two main areas:

- low available memory footprint
- maximum I/O throughput in Windows Server 2008

The following sections discuss the system demands and the related performance issues in these two areas.

Low Available Memory Footprint

SAS application workloads access data via the Windows Server 2008 application programming interfaces (APIs). Through these APIs, SAS uses the Windows operating system file cache to optimize file-system reads and writes. Under large I/O workloads, the file cache in Windows Server 2008 can expand to a very large size, up to 1 TB. When it expands to be nearly 100% of available RAM in the system, then the Windows system may slow down dramatically, to an almost hung state. This is the scenario we refer to as the low available memory footprint.

In working with Microsoft to understand why all the available memory is used in a Windows 2008 system, we uncovered an issue with using Microsoft's Random Access flag to access a SAS data file. We have changed the SAS 9.2M3 code to avoid using this flag and the issue has gone away. This fix is available from SAS Technical

Support via their hotfix download site.

If you are experiencing this low memory footprint issue, please do the following:

- Apply the Windows Server 2008 patch number 974609 (support.microsoft.com/default.aspx?scid=kb;EN-US;974609).
- Apply the SAS 9.2M3 hotfix number 39615 (http://support.sas.com/kb/39/615.html). Please carefully read this hotfix note as a Windows Environment Variable needs to be set to activate hotfix.

If you still experience heavy memory use after you apply the above patches, then SAS strongly recommends using the SGIO SAS system option (discussed later in this paper) globally. This option keeps SAS from using the default Windows API that goes through the file cache and forces direct I/O processing. This option also should significantly reduce the amount of memory that your SAS application workloads use. However, using SGIO also might cause your SAS application workloads to run slower.

Sustaining Maximum I/O Throughput

SAS customers are running larger and larger I/O workloads. While working with these customers, SAS has discovered several interesting features with some of the more common files systems used with Windows Server 2008.

- New Technology File System (NTFS) is the preferred file system for SAS®9 in Windows. Customers find that the file system works well when they have light to medium I/O workloads. It does work well with heavy I/O, but it requires a very robust storage array attached where you only use hardware striping to create the large volumes required for the large IO bandwidth.
- Common Internet File System (CIFS) is used with Microsoft clusters. Performance issues can occur when you access a large file that requires many pages to be held in memory. SAS is also working with Microsoft on this issue, and customers who encounter a similar issue are advised to contact Microsoft Support for help.
- General Parallel File System (GPFS) when a clustered file system is needed for SAS applications like SAS GRID, this file system from IBM is often used. It seems to work well for light to medium I/O workloads.

The architecture and setup of your storage subsystems is extremely crucial to good performance for file systems that are used by SAS. Performance of your disk subsystem (for example, storage area network [SAN] disk configuration, host bus adapter [HBA] queue depth, firmware, drivers, and so on) in Windows environments is extremely crucial to the I/O performance of any application that runs in Windows Server 2008.

Tuning Tips

After you understand the demands of your system and the possible performance issues, you can begin tuning your system. This section provides tips for tuning the following:

- Windows 2008 options
- SAS system options
- CPUs
- I/O subsystems
- Memory

Tuning Tips for Windows 2008

Here are a few tips to insure you have the maximum IO throughput from your storage array:

- If you need to use multiple LUNs within your storage array to create a file system, **do not** do the striping at the operating system level
- Apply the Microsoft Windows 2008 R2 hotfix to increase the Read Ahead activity available from Microsoft in mid-August 2011 (KB 2564236)
- Increase the number of Windows 2008 R2 worker threads to support the additional Read Ahead activity in the Windows Registry file.

```
        [HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\Executive]

        For Windows 2008 R2, the values are:
                AdditionalDelayedWorkerThreads = 0x28
```

AdditionalCriticalWorkerThreads = 0x28

For Windows 2008, the values are:

AdditionalDelayedWorkerThreads = 0x10
AdditionalCriticalWorkerThreads = 0x10

- In the creation of the LUNs on your storage array:
  - use a 64K Allocation Unit size
  - use RAID5
  - when possible have MANY disks underneath each LUN

Tuning Tips for SAS System Options

For optimal SAS performance, you must tune the following system options:

- **MEMSIZE:** The default value for the MEMSIZE system option is approximately 80% of the available memory in your computer at the invocation of the SAS session. This default value is the upper limit of the amount of memory that a SAS session can use. Because a 64-bit system in Windows Server 2008 has 128 GB of physical RAM, this value can be too large for SAS. SAS recommends that you set this value to either **512M** or **2G**. Remember that more memory does not equate to faster SAS run times.
- **SORTSIZE=**: The default value is **64M**. This value is too small for sorting a file larger than 1 GB. SAS recommends that you set it in **!SASROOT/config.sasv9** to either **256M** or **512M**. The only instance in which you should set SORTSIZE= larger than **512M** is when you have enough physical RAM to put the file that you want to sort into memory, thereby avoiding writing temporary utility files to disk.
- **SGIO:** You should turn on this option globally only when you cannot keep the file cache from consuming all of the memory. Before you turn on this option, be sure to read the section "Tuning Tips for Memory." **Note:** You must add several items to your SAS code and your system for SAS to work optimally with the SGIO system option. These additions are discussed in the section "SAS SGIO System Option."
  SGIO does not have any effect on the random access of data (for example, reading to or writing from an indexed SAS data set). **Note:** You must be running SAS 9.2 M3 for this option to work with Windows 2008 64-bit.
- **BUFSIZE=, UBUFSIZE=, and IBUFSIZE=:** When you create SAS data files, use these three system options to increase the buffer size for the created data files. Increasing these values to more closely align with your underlying storage arrays improves performance when you use SAS for very large, sequential reads and writes. In addition to these SAS parameters, there are a few SAS procedures that also have options that need to be added to their PROC statement to specify buffer sizes (such as SORT, NEURAL, …)

Tuning Tips for CPUs

CPU technology is constantly evolving with ever-increasing processor speeds. These ever-increasing processor speeds require more energy to support them, so the manufacturers have introduced a set of processor power states called "C" states to help reduce power consumption. These "C" states range from C0 (where all processors are in maximum power at all times) to C1 (where the processors are idle) all the way to C6 (where the processors are put in a very deep sleep when idle). The deeper the sleep the processors are put in, the longer the latency to wake them up to work. For optimal performance, SAS recommends that you keep your processors at C0 or C1 state. We have seen the performance of SAS jobs increase by 40% when switching from the default C state to C0 or C1. You will need to work with your hardware manufacturer on how to verify what "C" state your system is in and how to reset the value (some require it to be changed in the BIOS, others set it via operating system commands).

Occasionally, SAS can be a CPU-intensive application. Therefore, SAS recommends that you obtain the fastest CPU available to you. However, the fastest CPU might not be the one where a single SAS job runs fastest. This is true particularly in a situation where your overall use of SAS is for running multiple SAS sessions concurrently on the system. This statement comes from experience. SAS Technical Support frequently hears from customers who benchmark a single SAS job in a Windows environment, and then they compare it to benchmarking single jobs in other operating environments such as UNIX or Linux. These customers observe SAS performing better in Windows when single SAS jobs are run. However, for a true benchmark, you must also test multiple SAS sessions that are running simultaneously on both platforms.

Please note SAS does not fully exploit hyper-threads (HT) on Windows systems. SAS gets an additional 15-20% boost if you turn HT on over the number of true cores in your system, so it is okay to have them turned on.

Tuning Tips for I/O Subsystems

The key to configuring your I/O subsystem for SAS is having a good understanding of how SAS processes I/O. Then, you can more fully understand why multiple file systems are necessary. Balancing I/O is a two-part process. The first part relates to sizing and configuring the hardware; the second part relates to how you configure your file systems and distribute the SAS files across the file systems. Two papers that can help you with balancing your I/O are available on the SAS Customer Support site:

- "Best Practices for Configuring your IO Subsystem for SAS®9 Applications" (support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf) discusses the details of how SAS processes I/O and why multiple file systems are required to balance your I/O subsystem.
- "Frequently Asked Questions Regarding Storage Configurations" (support.sas.com/resources/papers/proceedings10/FAQforStorageConfiguration.pdf) addresses additional questions that go beyond the scope of the best-practices paper.

In addition to following the practices in these papers, follow these steps when you build an NTFS system that is required by SAS in Windows Server 2008:

1. Align the file offset with underlying disk topology. Both SAS and Microsoft strongly recommend that you set the offset to **1024K** (the default value for Windows Server 2008).
2. Specify a cluster size (allocation unit) of **64K** (this is the largest cluster size for Windows). Set this when you initially format the file system.
3. Validate these settings with a tool such as the Microsoft Windows **fsutil** command-line utility.
4. If you are using attached storage arrays, it is a good idea to follow the manufacturer's settings for the queue depth of the HBA. The HBA queue depth defines the maximum number of outstanding queue requests. Typically, this depth is set with a tool that is provided by the HBA vendor. For example, suppose you use either Emulex or QLogic HBA management tools to validate or set the queue depth. The default value for the queue depth is typically **32**, which helps prevent the array from becoming too busy. However, if you have multiple hosts and servers connected to the same storage array or subsystem, it might be good to reduce this value. On a dedicated storage array (for your server only) increasing the value to **64** or even **128** can help increase performance. As mentioned earlier, be sure to consult with your storage manufacturer and HBA vendor for the optimal settings.

Note: When you upgrade from Windows Server 2003 to Windows Server 2008, Microsoft strongly recommends that you delete the volume and then reformat the file system. These actions ensure that both the cluster size and partition offset are set properly for Windows Server 2008. It is a common oversight not to re-validate the file offset. If you leave the default value of **31.5K** as the file offset for Windows Server 2003, you still will have file-system performance issues when you upgrade to Windows Server 2008. For more information about this issue, contact Microsoft Technical Support.

Tuning Tips for Memory

In a discussion about memory, it is very important to distinguish between a 32-bit and a 64-bit Windows Server 2008 environment. SAS has different recommendations for the two Windows operating systems. For 32-bit Windows Server 2008 systems, SAS recommends no more than 8 GB of physical RAM in the system, regardless of the number of cores in the system. For 64-bit Windows Server 2008 systems, SAS recommends at least 4 GB of physical RAM per core in the system.

You also need to understand the amount of virtual memory that is required in the system. SAS recommends that virtual memory be 1.5 times the amount of physical RAM. If, in monitoring your system, you see that your computer is paging a lot, then SAS recommends that you either add memory or move the paging file to a drive with a more robust I/O throughput rate compared to the default drive C. At times, you might need to do both: add more memory and move the paging file.

In addition to supporting paging, memory also supports the operating system's file cache. It is important to note that SAS does not directly read data from or write data to disks. Instead, SAS uses a Windows API that goes through the operating system's file cache (that is, the memory). With Windows Server 2008, SAS has found that that the process of flushing data to the disk associated with Windows Server 2008 can be a compute-intensive endeavor.

- A SAS 9.2M3 hotfix is available (http://support.sas.com/kb/39/615.html) to help SAS use less file cache when doing Random Access activities. Please carefully read this hotfix note as a Windows Environment Variable needs to be set to activate hotfix. Please note this hotfix does not apply to SAS Performance Data Server (SPDS).
- SAS also strongly recommends that you apply the Windows Server 2008 patch 974609 (support.microsoft.com/default.aspx?scid=kb;EN-US;974609).

- If you run Windows Server 2008 R2, SAS recommends that you apply Windows Server 2008 R2 patch 979149 (support.microsoft.com/default.aspx?scid=kb;EN-US;979149).

Please note if you are using SPDS, we strongly recommend that you use the Windows Server 2008 tool called Dyncache to limit the amount of physical RAM can be used for file cache. Please contact Microsoft for details on this tool. Also, please monitor this SAS Technical Support note (http://support.sas.com/kb/43/943.html) regarding an upcoming hotfix to SPDS to overcome this issue.

Best Practices to Improve Performance

In addition to the tuning tips offered in the previous sections, you can also improve performance by implementing the following practices:

- For a 32-bit Windows Server 2008 system, the general recommendation from Microsoft for pagefile size is approximately 1.5 times the amount of physical RAM that is installed on the computer. You can determine a more specific requirement for the pagefile size based on the amount of RAM on the server and the system's Committed Bytes that are in use:

  Commit limit = size or RAM + pagefile size

- For a 64-bit Windows Server 2008 system, Microsoft provides a different recommendation (support.microsoft.com/kb/889654).

- Do not run your antivirus software in real-time mode. If you **must** run it in real-time mode, then be sure to exclude the following file types from your virus scan list:

  - SAS*
  - LCK*
  - UTL*

- Do not run backups during heavy SAS processing, including the extraction, transformation, and loading (ETL) processing at night as well as during end-user processing throughout the day.

- Run a disk defragmentation tool often on the file systems that are used by SAS, particularly the file system that is associated with the SAS WORK library (where SAS creates the temporary files).

Virtualization Concerns

Virtualization (via VMware) is supported by SAS. However, you should ensure that your guest systems on your VMware computer are properly set up with the amount of RAM, the number of cores, and I/O throughput that is required to support your peak SAS load.

In addition, be aware of other applications running on the same VMware computer that will share the underlying computer hardware resources along with SAS. This awareness helps ensure that there are no computer resource conflicts in this virtual environment.

Note: A SAS application or solution is generally not a good fit for a VMware environment, especially when you use SAS for heavy I/O processing. Performance cannot be tracked through a virtual environment. Therefore, if you run a SAS application or solution in a virtual environment and you encounter problems, SAS Technical Support cannot help you with these problems.

System Monitoring

Both SAS and Microsoft strongly advise that you proactively and closely monitor the computer resources that are used in your Windows environment to avoid running out of resources, thereby causing poor performance. You should regularly collect and analyze the performance measures for the following areas:

- overall server (aggregate server-level measures)
- CPU (total and individual CPUs)
- I/O throughput (total throughput and by file system)
- SAS file systems (especially SASWORK)
- memory and system file cache
- network

- individual SAS processes

You can use these measures to detect gradual or sudden performance degradations. In these areas, if you encounter degradations that you cannot alleviate by using only one instance of the Windows operating system (that is, one server), then you should consider one of the following options:

- run your SAS application workloads in multiple Windows environments
- migrate your SAS application workloads to a UNIX or a Linux platform

The monitoring tool that SAS recommends is the Performance Monitor (PerfMon) tool from Microsoft that comes with the Windows operating system. Hundreds of performance counters are associated with this tool, but from a SAS perspective, only the following subset of PerfMon objects and their associated counters is of primary interest:

- **Cache**
    - Data Flush Pages/sec
    - Dirty Pages
    - Fast Read Not Possible/sec
    - Fast Read Resource Misses/sec
    - Lazy Write Pages/sec
    - Read Aheads/sec
- **Client-Side Caching**
    - Application Bytes Read from Cache
    - Application Bytes Read from Server (Not Cached)
- **Job Object:** Process Count Active
- **Job Object Details**
    - %Processor Time
    - %User Time
    - Elapsed Time
    - I/O Data Bytes/sec
    - I/O Read Bytes/sec
    - I/O Write Bytes/sec
    - Page Faults/sec
    - Page File Bytes
    - Working Set
- **Logical Disk and Physical Disk (for each disk and for _Total)**
    - Avg. Disk Queue Length
    - Current Disk Queue Length
    - Disk Read Bytes/sec
    - Disk Write Bytes/sec
    - Disk Transfers/sec
    - Free Megabytes (Logical Disk)
- **Memory**
    - %Committed Bytes in Use
    - Available Megabytes
    - Cache Bytes
    - Cache Bytes Peak
    - Cache Faults/sec
    - Committed Bytes
    - Page Faults/sec
    - Demand Zero Page Faults/sec
    - Free System Page Table Entries
    - System Cache Resident Bytes
- **Network Interface**
    - Current Bandwidth
    - Output Queue Length
- **Paging File (for each paging-file instance, if more than one)**
    - %Usage
    - %Usage Peak
- **Process (for each process and for _Total)**
    - %Privileged Time
    - % User Time

  - Elapsed Time
  - I/O Data Bytes/sec
  - I/O Read Bytes/sec
  - I/O Write Bytes/sec
  - Working Set
  - Page Faults/sec
  - Page File Bytes
- **Processor (for each processor and for _Total)**
  - %Processor Time
  - %Privileged Time
  - %User Time
  - Interrupts/sec
- **Redirector**
  - Bytes Received/sec
  - Bytes Transmitted/sec
  - Read Bytes Cache/sec
  - Read Bytes Non-Paging/sec
  - Read Bytes Paging/sec
  - Read Operations Random/sec
  - Write Bytes Cache/sec
  - Write Bytes Non-Paging/sec
  - Write Bytes Paging/sec
  - Write Operations Random/sec
- **Server**
  - Errors System
  - Files Open
  - Pool Paged Failures
  - Server Sessions
- **Server Work Queues**
  - Bytes Received/sec
  - Bytes Length/sec
  - Queue Length
  - Read Bytes/sec
  - Write Bytes/sec
- **System**
  - Context Switches/sec
  - File Read Bytes/sec
  - File Write Bytes/sec
  - Processes
  - Processor Queue Length
  - System Up Time

To determine how to interpret the information that is gathered by these counters, see "Solving SAS Performance Problems: Employing Host Based Tools" (support.sas.com/rnd/scalability/papers/TonySUGI31_20060403.pdf).

SAS SGIO System Option

As mentioned previously in "Tuning Tips for SAS System Options," one method for reducing the amount of memory that is used while SAS executes is to force SAS to perform direct I/O processing, which avoids the Windows file cache as much as possible. You can do this by using the SAS system option SGIO, which activates the scatter-read/gather-write feature. For specific instructions on the proper use of this option, see "Achieving Better I/O Throughput Using SGIO in the Microsoft Windows Environment" (support.sas.com/resources/papers/IOthruSGIO.pdf). When SGIO is turned on, it forces most of the SAS I/O activity to bypass the file cache. However, some files (for example, utility files that are created by the SORT procedure) will still use file cache. To force these files to use direct I/O (via the SGIO option), add the READMODE=HINT and the USEDIRECTIO options in the PROC SORT statement.

Avoiding the file cache removes the read-ahead and write-back features of many file systems. One way to mimic these features is to increase the number of buffers that SAS has active in memory. SAS suggests that you increase the buffers by setting the SAS system option BUFNO=16. Another good practice (even if you do not turn on the SGIO option) is to increase the size of the buffer that is being transferred so that it matches your storage setup. This action results in fewer I/O transfers and faster processing of data. The suggested increase of the default buffer size is 128 KB.

Note: You must be running the third maintenance release of SAS 9.2 (TS2M3), and you must apply Hot Fix B25010 to fix an issue with SGIO. To download this hot fix, see SAS Note 40157, "SAS® 9.2 ignores SGIO processing in a Windows x64 operating environment" (support.sas.com/kb/40/157.html).

In addition, be aware that not using the file cache for small files that can stay in memory will cause performance degradation in your SAS application workloads. So setting SGIO=YES globally may not be the best solution.

CONCLUSION

Taking the time to plan the hardware configuration and file-system layout for your Windows Server 2008 system is well worth the time and effort. These activities, which should occur prior to installing SAS, are important determinants of your system performance. As data requirements grow, planning the I/O subsystem and file-system layout becomes even more important. Modifying the systems after an I/O bottleneck has been identified is more difficult than initially taking the time to set them up correctly.

Once your system is set up and running, tuning activities will be ongoing. Regular monitoring of your file systems and other computer resources will have a positive impact on performance. Testing the impact of SAS system options such as BUFSIZE, MEMSIZE, and SORTSIZE= in your environment will determine which options provide the best performance improvements. In addition, ongoing monitoring with the Windows Server 2008 Performance Monitor tool will help you correctly identify the impact of your SAS tasks and avoid encountering system resource bottlenecks.

ACKNOWLEDGMENTS

I would like to extend many thanks to Tony Brown, Tom Keefer, BJ Mulligan, Maggie Underberg, and John Maxwell from SAS Institute Inc., and Pushkar Prasad from Microsoft for their review of this paper.

RECOMMENDED READING

Brown, Tony. 2008. "SAS® Performance Monitoring – A Deeper Discussion ®." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/forum2008/387-2008.pdf**.**

Crevar, Margaret, Tony Brown, and Leigh Ihnen. 2007. "Best Practices for Configuring your IO Subsystem for SAS®9 Applications." Cary, NC: SAS Institute Inc. Available at support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf**.**

Crevar, Margaret. 2009. "How to Maintain Happy SAS®9 Users." *Proceedings of the SAS Global Forum 2009 Conference*. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/proceedings09/310-2009.pdf.

Somak. 2009. "Microsoft Windows Dynamic Cache Service." Microsoft Corporation. Available at blogs.msdn.com/b/ntdebugging/archive/2009/02/06/microsoft-windows-dynamic-cache-service.aspx.

CONTACT INFORMATION

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

Margaret Crevar
SAS Campus Drive, R-2427
SAS Institute Inc.
Cary, NC 27513
919.531.7095
Margaret.Crevar@sas.com
www.sas.com