THE POWER TO KNOW.

SAS® Financial Management 5.3
Performance Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Financial Management 5.3: Performance Guide*. Cary, NC: SAS Institute Inc.

SAS® Financial Management 5.3: Performance Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, April 2012

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Contents

CHAPTER 1

Introduction

What's in This Document

The SAS 9.3 middle-tier environment provides an execution environment for SAS Web applications that are used to perform business analytics. Optimizing the middle tier and SAS Web applications for scalability and performance requires careful planning and effort to carry out the plan. The planning and execution can be complex as they balance the demands for availability, reliability, security, and performance.

This document discusses the methodology and parameters for tuning SAS Financial Management for performance and scalability. This tuning should be completed at installation and configuration time.

The appendix contains preconfigured settings for many scenarios and sizes.

Note: This document addresses only 64-bit operating systems. For optimal performance, a 64-bit operating system is a necessity.

Additional References

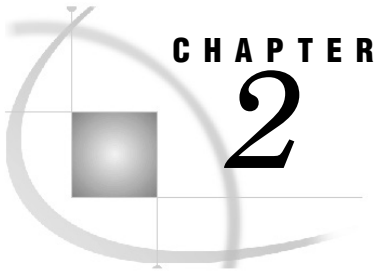
- *Configuration and Tuning Guidelines for SAS 9 in Microsoft Windows Server 2008*, available at http://support.sas.com/resources/papers/WindowsServer2008ConfigurationandTuning.pdf
- *SAS 9.3 Web Applications: Tuning for Performance and Scalability*, available at http://support.sas.com/resources/thirdpartysupport/v93/appservers/index.html.

  If you are using SAS Business Intelligence products, review the scalability and performance document with regard to those products. Ignore settings for Server3 and Server4.

Check for Updates

Feedback is welcome. Updates are made regularly as new data is available.

For additions and updates to these instructions, please visit the SAS Financial Management documentation site: http://support.sas.com/documentation/onlinedoc/fm/index.html.



CHAPTER 2

First Steps

Establish Your Performance Objective

Establishing your performance objective is the most important step of the entire process for tuning the performance and scalability of SAS Financial Management.

Performance requirements are usually identified in terms of transaction response time, number of transactions per second, throughput time, resource utilization, total cost per transaction, availability, and more.

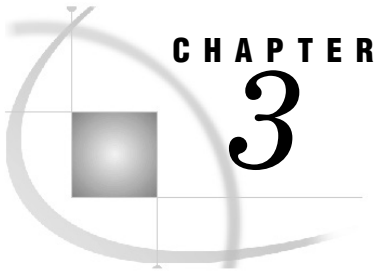
Scalability often refers to a component's ability to adapt readily to a greater or lesser intensity of use, volume, or demand while meeting integral business objectives. The common objective of scaling a component or system is to increase the capacity for growth, increase the speed of the component, improve the efficiency, or shift or reduce the load on the component.

Adjust JVM Options

In most cases, the original configuration of SAS 9.3 Web applications is sufficient only to meet initial installation and sample data requirements. Many of the performance and scalability improvements are accomplished by adjusting the JVM options for the Web application server. Tuning options are provided according to the Java Development Kit (JDK) vendor:

- Sun and HP JDKs: These JDKs use the same JVM options.
- IBM JDK: An IBM JDK is used for all Web application servers deployed on AIX. The IBM JDK is also used for the IBM WebSphere Application Server, unless WebSphere Application Server is deployed on a Sun Solaris platform.

For more information, see *SAS 9.3 Web Applications: Tuning for Performance and Scalability*. This document suggests JVM options for meeting your scalability and performance objectives. Once you have applied the JVM options, test the applications and adjust the JVM options accordingly.



CHAPTER 3

Operating System Options

Overview

The setup of the operating system, especially on the middle tier, is second in importance only to JVM arguments, which are discussed in the next chapter. Operating system settings, particularly virtual memory settings, are often overlooked but extremely important, because the move to 64-bit operating systems has dramatically changed the way that we use memory arguments in Java.

Java and Virtual Memory

Java allocates and maintains all its heap allocations (-Xms -Xmx) in virtual memory (a pagefile on Windows; swap space on UNIX).

For all SAS tiers, we recommend a virtual memory allocation that is 1.5 to 2 times the amount of physical memory. On machines with low available hard drive space, make a greater virtual memory allocation. At a minimum, you must allocate virtual memory that is greater than the physical memory size if you are tuning for performance and have a goal of greater than 90% of memory being allocated to JVMs.

To avoid paging losses, the virtual memory should be configured on the drive with the fastest I/O throughput. Setting up virtual memory correctly can result in the greatest performance increases, even with the configuration defaults still set.

For more information, see *Configuration and Tuning Guidelines for SAS 9 in Microsoft Windows Server 2008*.

CHAPTER 4

Scaling the Middle Tier

Changing the Default Configurations

Out of the box, the JVMs for the SAS Intelligence Platform and SAS Financial Management have a default setting that does not exceed 2 GB. All instances should be tuned during post configuration, based on the size of the server being installed.

The following table lists the default servers and some of their contents:

Server	Contents
SASServer1	Web Infrastructure Platform (WIP), the Logon Manager, the SAS Information Delivery Portal, SAS BI Dashboard
SASServer2	SAS Web Report Studio, SAS WebDoc application
SASServer3	SAS Financial Management, SAS Solutions Services
SASServer4	ODCS
SASServer5	SAS Financial Management Web Data Entry

On the proposed middle tier, 90% of the memory is allocated to the various JVMs. The other 10% is reserved for the operating system and administrative clients, as needed. On systems with more memory (48 GB and above), this recommendation can be trimmed down, reserving a maximum of 6 GB for the JVMs and leaving a minimum of 1.5 GB for the operating system.

Note: When the system consumes greater than 95% of memory, more paging than normal occurs, with a negative effect on performance. Greater heap size does not result in more garbage collection or lower performance, although that is a common misconception. In fact, the opposite has proven to be true in real world scenarios and performance testing in 64-bit operating systems.

Unless the memory of the middle-tier system is less than 32 GB, the -Xms and -Xmx settings for the Remote Services should always be at least 512m and 1024m, respectively. If the total memory is less than 32 GB, the settings can be less, such as 256m and 512m.

For information about changing JVM arguments, see *SAS 9.3 Web Applications: Tuning for Performance and Scalability*, available at http://support.sas.com/resources/thirdpartysupport/v93/appservers/index.html.

SAS Financial Management and ODCS

This scenario has two assumptions and potential performance considerations:

- heavy ODCS usage by SAS Financial Management (which is typical)
  If auto-allocation is being used, SASServer3 usage is higher, because most auto-allocation occurs with the base SAS Financial Management code and uses heavy threading when cores are available. In that case, you could make more of a balanced split between SASServer3 and SASserver4.
- minimal to low SAS Business Intelligence usage
  The defaults for SASServer1 and SASServer2 are not increased unless increases prove to be needed in a subsequent tuning.

Note: This example is for systems with 64 GB of memory. The appendix has additional examples.

Target Allocation Server/OS	Approximate Memory %	PermSize/MaxPermSize Range (for example, 1024m/2048m)
Operating System/Admin Server/NodeManager	10%	
SAS Remote Services	1 - 2.5%	Default
SASServer1	3.2 – 4%	Default
SASServer2	Default	Default
SASServer3	12.5%	3 – 6.25%
SASServer4	25 – 55%	3 – 6.25%
SASServer5	6.25%	1.5 – 2.3%

Multiple Web Application Tier or Separated ODCS Tier

Some sites might choose to separate the middle tier, in order to expand available resources and I/O. In this scenario, SASServer4 and SASServer5 are created on a different machine.

The memory is distributed in much the same way as in the previous scenario. However, on the primary middle tier, the memory for SASServer4 and SASServer5 is allocated to the other servers. The new ODCS tier is used only for SASServer4 and SASServer5, and memory is distributed appropriately. For reference, compare the following tables to the previous scenario.

Primary Middle Tier (SAS Intelligence Platform and SAS Financial Management)

Target Allocation Server/OS	Approximate Memory %	PermSize/MaxPermSize[1]
Operating System/Admin Server/NodeManager	10% Page file: 1.5 times physical memory	
SAS Remote Services	1 - 2.5%	Default
SASServer1	10 - 20%	1.5% / 3%
SASServer2	10 - 20%	Default
SASServer3	50 – 69%	3% / 6.25%

ODCS Tier

Target Allocation Server/OS	Approximate Memory %	PermSize/MaxPermSize
Operating System/Admin Server/NodeManager	10% Page file: 1.5 times physical memory	
SASServer4	60 - 80%	3% / 6.25%
SASServer5	10 - 30%	1.5% / 2.3%

For scenarios with more than two tiers, consult your SAS representative. At all times, the memory options should be set to use 90% of the system memory under maximum load, and virtual memory should be set to a minimum of 1.5 times physical memory.

Balanced Approach with SAS Business Intelligence Considerations

The balanced approach is for a site that uses SAS Business Intelligence products as well as SAS Financial Management. This approach still has focus on a SAS Financial Management customer, and ODCS uses a large amount of available memory, but the recommended settings result in performance improvements for all products.

[1] For example, 1024m/2048m. In systems with less memory, these values might be lower.

Target Allocation Server/OS	Approximate Memory %	PermSize/MaxPermSize
Operating System/Admin Server/NodeManager	10%	
SAS Remote Services	1 - 2.5%	Default
SASServer1	6.25 - 12.5%	1.5% / 3.1%
SASServer2	3.2 - 6.25%	1% / 1.5%
SASServer3	12.5 - 25%	3% / 6.25%
SASServer4	25 - 37.5%	3% / 6.25%
SASServer5	6.25%	1.5% / 2.3%

CHAPTER 5

MySQL Server Optimizations

Overview

Back up the MySQL database using mysqldump before changing any settings. At the same time, make a backup of the my.ini or my.cnf file in the MySQL installation directory.

For further reference, see http://dev.mysql.com/doc/refman/5.1/en/innodb-tuning.html.

Memory Usage

As of SAS Financial Management 5.3, the MySQL configuration file (my.ini or my.cnf)) has the following settings:

```
key_buffer_size=128M
innodb_buffer_pool_size=2G
```

The **innodb_buffer_pool_size** represents the amount of memory MySQL uses. Increase this setting if necessary, if there is more memory available on this system. If MySQL Server is installed on the data tier, keep in mind other demands on the data tier.

Increasing Transactional Database Size

Most current installations on Windows and UNIX automatically extend the transactional database size. Manually changing this setting is not necessary unless errors are encountered in MySQL.

If INNODB errors are encountered, you can change the size of the INNODB transactional database to increase the amount of data stored and increase performance in some cases. You cannot increase the original ibdata1 file because data could be lost.

1. Perform a full MySQL backup.
2. Remove the iblogs files (after backing them up) because they cannot resize themselves.
3. Then extend the transactional database size in the configuration file, following instructions in the MySQL documentation:

   http://dev.mysql.com/doc/refman/5.1/en/innodb-data-log-reconfiguration.html

CHAPTER 6

SAS Financial Management and ODCS Query Processors

Introduction

A configuration with multiple query processors is most effective in situations where there are many concurrent users executing small simultaneous queries. Although it does not improve the performance of individual queries, it does allow more queries to execute simultaneously and thus improves overall throughput.

Setup and Tuning Considerations

The *SAS Financial Management: System Administration Guide* describes the setup of query processors. See http://support.sas.com/documentation/onlinedoc/fm/index.html.

Use the system administration guide in conjunction with this chapter to review options and extra descriptions as needed. There are not many options to tune, but there are some major points of discussion.

Unless otherwise noted, the options listed below apply to the query processor-dedicated machine. In tuning, make a note when you dedicate an entire machine to the query processor. Specify settings such as odcs.queryprocessor.maxthreads in the **`-Doption=value`** format, as outlined in the system administration guide.

- Set **-Xms** to 50% of the available memory on the system. Set **-Xmx** to 90% of the available memory.
- The rule of thumb to having enough query processors is never to let the total CPU utilization exceed 90%. This also means that the Dispatcher queue should always be empty, as all requests are dispatched immediately to an open query processor thread. The setting that controls how many threads that system can handle is **odcs.queryprocessor.maxthreads**, which should not be set in most cases.

  One query processor can be run per server to better utilize system resources. It will automatically detect the number of available processor cores and run a query thread for each core. There is no advantage to running multiple query processors on one server, because the fact cache that is maintained by each query processor is the greatest area of performance loss. Maintaining only one fact cache per server minimizes network overhead and database calls.
- **odcs.queryprocessor.reattach** defaults to false. In production situations, this option should be set to true, to allow each query processor to automatically reattach to ODCS in the event of a restart.

Note: after a hot fix, query processors must be restarted to pick up any JAR changes.

- On the ODCS server, the option **odcs.dispatcher.use.internal.qp** is set to true by default. The internal query processor is approximately two times faster than external query processors. However, the internal query processor can scale only based on the available resources of the ODCS server.

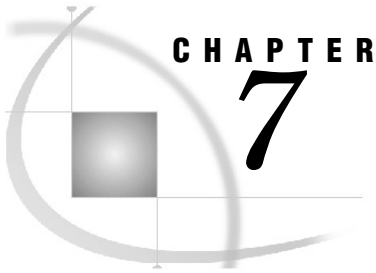
  When you add external query processors, the overhead of dispatching queries can begin to slow down the internal query processor. When you implement external query processors, set **odcs.dispatcher.use.internal.qp** to false. This scenario assumes that you have at least an equal number of cores between query processors and the ODCS server.
- In conjunction with the internal query processor change, you might need to monitor the **odcs.perform.writeback.on.internal.qp** option if writeback performance becomes a problem with internal query processors. This option is true by default. Setting it to false on large implementations can be greatly beneficial. It removes work from the ODCS server, which then has more processing power for dispatching queries to the external query processors.

  In smaller implementations, monitor performance with this setting. Performance of the internal query processor writeback can be superior.

Troubleshooting

Query processors do not offer much debug logging at this time. Referring to the *SAS Financial Management: System Administration Guide*, you can configure logging for query processors. If problems arise, you can set the main package to DEBUG. Other logging for query processors occurs on the ODCS server in the SASODCS log and the SASServer4 logs on your middle-tier machine.

Direct any additional troubleshooting and debugging issues to SAS Technical Support.



CHAPTER 7

Miscellaneous Topics

Performance Monitoring and Additional Java Tuning

The recommendations in this document have been field-tested and performance-tested. They represent our experience and customer experience to date as the best-case starting scenario to be implemented at installation and configuration time. However, these tuning numbers are just a start. Each situation is unique. New data, forms, and users add levels of complexity.

Java tuning is a continual operation and should be monitored. Typically, it is optimal if a server uses 60–70% of the JVM under normal load, leaving some memory available in case of heavier usage.

In some cases, you might notice that SASServer1 is continually at 90–100% usage, causing some performance degradation, but SASServer2 is consistently at 10–20% usage. Some of that free memory can be moved from SASServer2 to SASServer1.

There are many tools for Java tuning, but you can also do most of this monitoring with a simple administration tool for the systems. You can even combine small tools such as Process Explorer and Task Manager with the Web application server logs and Heap tools that are included with the Web application server.

Logging Level Considerations

For development and test environments, the default logging configurations are useful for diagnosing problems. However, be aware that some logging settings (such as for the ODCS server) can log as much as 500 MB per hour under load.

For production systems or systems having no problems, you can change most logging levels from INFO to WARN under ODCS and SAS Financial Management. The configuration files are located in the **SAS-config-dir\Lev1\Common\LogConfig** directory.

Full Application Restart and Database Changes

After a full server restart or data loads to the database, the first login is slower as the new information is cached and files are compiled. This is normal behavior and to be expected after scheduled outages. Nightly and even weekly restarts are no longer necessary. Even backup processes no longer bring all systems down, but merely pause the current system state to prevent users from making changes during the backup process.

Troubleshooting

Unexpected Restart Errors in Application Server Logs

If you see error messages in the Web application server logs about an unexpected restart, look for messages about invalid parameters. The message should tell you exactly which one is incorrect. Check that parameter against the valid parameters for the particular Web application server. Also, double-check that there are no invalid characters or spacing issues in what was edited. Typically, one of these two approaches solves any restart problems after tuning.

Another cause of tuning problems is allocating more memory than you have on the system and as a consequence more virtual memory. If you receive error messages on start-up that there is not enough heap space or permgen space, verify that there are no typographical errors on the size specification. Then verify that the total memory allocated on the middle tier does not exceed (by much) 90% memory usage and that your virtual memory is at least greater than physical memory. (Recall that the recommended size for virtual memory is 1.5 times the size of the physical memory.)

MySQL Does Not Restart after Changes

Before editing the memory settings for MySQL, make sure that 64-bit MySQL Server is installed. Make sure that there is enough available memory on the system to increase the MySQL setting.

Another cause of errors on UNIX systems arises if the user who is starting MySQL does not have access to that amount of memory for a single process. Check the user's "limits" settings.

If you changed the size of the INNODB transactional database, make sure that you removed the iblogs files (after backing them up). In addition, make sure that the original ibdata file has not changed in size and that only the new files have larger sizes. As mentioned earlier, the size of the original ibdata file cannot be changed.

APPENDIX 1

JVM Tuning Examples

Overview

This appendix can serve as a "cheat sheet" for general recommendations based on products and sizing for a site. Each scenario provides examples from minimum requirements to 96 GB of memory for maximum system utilization and performance out of the box.

Note: In the tables in this appendix, PermSize and MaxPermSize recommendations apply to WebLogic and JBoss.

Financial Management and ODCS

The following JVM parameters are based on heavy ODCS memory usage, which is typical with SAS Financial Management. If SASServer3 usage is low and ODCS usage is extremely high, you could decrease SASServer3 memory by setting it to 1–2 GB above maximum observed usage and moving the rest to SASServer4.

Another possibility (when ODCS is demanding more memory than allocated) is to take memory from other servers, in particular SASServer1 and SASServer2, moving those servers closer to their default settings. Typically, the Web Infrastructure Platform and SAS Web Report Studio do not need the higher resources but can benefit on overall system performance from a moderate increase in memory.

Table FM-16. 16 GB. Very Little Room for Tuning. Assumes SAS Financial Management usage.

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	1.6 GB (1 GB minimum) Pagefile: 24 GB minimum	
SAS Remote Services	Default	Default
SASServer1	Default (2 GB)	Default
SASServer2	Default (2 GB)	Default
SASServer3	2048m/2500m	768m/768m
SASServer4	2048m/4096m	768m/1024m
SASServer5	Default (Turn off and distribute if Web Data Entry is not used.)	Default

Table FM-24. 24 GB. Assumes SAS Financial Management usage. Further balancing is possible toward more heavily used applications.

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	2.4 GB (2 GB minimum) Pagefile: 32 GB minimum	
SAS Remote Services	768m/1024m	Default
SASServer1	2048m/2048m	Default
SASServer2	1024/1500m	Default
SASServer3	2048m/4096m	768m/1024m
SASServer4	4096m/11000m	1024m/2500m
SASServer5	Default (Turn off and distribute if Web Data Entry is not used.)	Default

Table FM-32. 32 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	3.2 GB (2 GB minimum) Pagefile: 48 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	2048m/2048m	Default
SASServer2	1024m/2048m	Default
SASServer3	4096m/5000m	1024m/2048m
SASServer4	8000m/16000m	2048m/4096m
SASServer5	2048m/2500m (Turn off and distribute if Web Data Entry is not used and apply to ODCS.)	768m/1024m

Table FM-48. 48 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	4.8 GB (3 GB minimum) Pagefile: 72 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	2048m/2500m	Default
SASServer2	2048m/2048m	Default
SASServer3	4096m/6000m	1024m/2048m
SASServer4	16000m/29000m	2048m/4096m
SASServer5	2048m/3000m (Turn off and distribute if Web Data Entry is not used.)	1024m/1500m

Table FM-64. 64 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	6.4 GB (4 GB minimum) Pagefile: 96 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	4096m/8000m	Default
SASServer2	2048m/2500m	Default
SASServer3	6000m/8000m	1024m/2048m
SASServer4	20000m/40000m	4096m/6096m
SASServer5	2048m/3000m (Turn off and distribute if Web Data Entry is not used.)	1024m/1500m

Table FM-96. 96 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	9.6 GB (6 GB minimum) Pagefile: 154 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	4096m/4096m	Default
SASServer2	4096m/4096m	Default
SASServer3	8000m/16000m	1024m/2048m
SASServer4	24000m/56000m	2048m/4096m
SASServer5	2048m/4096m (Turn off and distribute if Web Data Entry is not used.)	1024m/1500m

Balanced Approach with BI Considerations

Table BALANCED-16. 16 GB Guidelines. Assumes SAS Financial Management usage.

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	1.6 GB (1 GB minimum) Pagefile: 24 GB minimum	
SAS Remote Services	Default	Default
SASServer1	Default (2 GB)	Default
SASServer2	Default (2 GB)	Default
SASServer3	2048m/2500m	768m/768m
SASServer4	2048m/4096m	768m/1024m
SASServer5	Default (Turn off and distribute if Web Data Entry is not used.)	Default

Note: 16 GB does not provide any significant tuning space.

Table BALANCED-24. 24 GB Assumes SAS Financial Management usage. Further balancing is possible toward more heavily used applications.

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	2.4 GB (2 GB minimum) Pagefile: 32 GB minimum	
SAS Remote Services	768m/1024m	Default
SASServer1	2048m/3000m	Default
SASServer2	2048m/2500m	Default
SASServer3	2048m/4096m	768m/1024m
SASServer4	4096m/8000m	1024m/2500m
SASServer5	Default (Turn off and distribute if Web Data Entry is not used.)	Default

Table BALANCED-32. 32 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	3.2 GB (2 GB minimum) Pagefile: 48 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	2048m/4096m	Default
SASServer2	2048m/2500m	Default
SASServer3	4096m/6000mm	1024m/2048m
SASServer4	6000m/12000m	2048m/3000m
SASServer5	2048m/3000m (Turn off and distribute if Web Data Entry is not used)	768m/1024m

Table BALANCED-48. 48 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	4.8 (3 GB minimum) Pagefile: 72 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	4096m/6000m	Default
SASServer2	2048m/4096m	Default
SASServer3	4096m/8000m	1024m/2048m
SASServer4	16000m/20000m	2048m/4096m
SASServer5	2048m/4096m (Turn off and distribute if Web Data Entry is not used.)	1024m/1500m

Table BALANCED-64. 64 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	6.4 GB (4 GB minimum) Pagefile: 96 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	4096m/8000m	Default
SASServer2	4096m/6000m	Default
SASServer3	8000m/12000m	1024m/2048m
SASServer4	20000m/24000m	2048m/4096m
SASServer5	4096m/6096m (Turn off and distribute if Web Data Entry is not used)	1024m/1500m

Table BALANCED-96. 96 GB

Target Allocation Server/OS	Memory and Virtual Memory Allocations	PermSize/MaxPerm Size
Operating System/Admin Server/NodeManager	9.6 GB (6 GB minimum) Pagefile: 154 GB minimum	
SAS Remote Services	768m/1280m	Default
SASServer1	4096m/8000m	Default
SASServer2	4096m/8000m	Default
SASServer3	8000m/16000m	1024m/2048m
SASServer4	24000m/48000m	2048m/4096m
SASServer5	2048m/4096m (Turn off and distribute if Web Data Entry is not used.)	1024m/1500m