SAS® Financial Management 5.5

Performance Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS® Financial Management 5.5: Performance Guide*. Cary, NC: SAS Institute Inc.

SAS® Financial Management 5.5: Performance Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

April 2015

Contents

Chapter 1
Introduction

Overview

This chapter describes the following topics:

- about the *SAS Financial Management: Performance Guide*
- intended audience
- task list for tuning for scalability and performance
- related documentation

About This Document

Note: Use this guide after you install and configure SAS Financial Management.

The SAS 9.4 middle-tier environment provides an execution environment for the SAS web applications that customers can use to perform business analytics. Optimizing the middle-tier environment and the SAS web applications for scalability and performance requires careful planning and effort. The planning and the execution of optimizing an environment can be complex because it involves balancing the demands for availability, reliability, security, and performance.

This document outlines the methodology and options that you can use to tune SAS Financial Management for performance and scalability. To ensure the best SAS Financial Management performance starting point possible that is based on available resources, you should tune for performance and scalability after you install and configure SAS Financial Management.

For general recommendations that are based on products and the size of a customer site, see Appendix 1, “JVM Tuning Examples,” on page 25.

Note: If you prefer or require prompting when tuning, you can use the JVM argument tables to adjust the memory variables during the SAS Deployment Wizard installation and configuration. For more information about using the JVM argument tables, seeAppendix 1, “JVM Tuning Examples,” on page 25.

Audience

This guide is meant to assist SAS installers and consultants with tuning SAS Financial Management as needed for the hardware used in a customer environment and the size of a customer environment. Users should complete the tasks that are documented in this guide after installing and configuring SAS Financial Management.

This guide is also meant to assist SAS administrators tune SAS Financial Management for their site environment.

Tuning for Performance and Scalability Task List

The following list summarizes the most important tasks that you must complete when you tune the performance and scalability of SAS Financial Management in a site environment:

- establish your performance objective.

  For information about establishing your performance objective, see “Determining Your Performance Objective” on page 5.

- review and tune your JVM options.

  For information about tuning JVM options, see “Reviewing and Adjusting JVM Options” on page 6.

- tune your operating system options.

  For information about your operating system options, see Chapter 3, “Configuring the Operating System,” on page 7.

- tune the SAS Web Server HTTP reverse proxy server.

  For information about modifying the configuration of the HTTP reverse proxy server, see “Tuning the SAS Web Server” on page 12.

- optimize the SAS Web Infrastructure Platform Data Server.

  For information about tuning the data server, see Chapter 5, “Optimizing the SAS Web Infrastructure Platform Data Server,” on page 17.

- configure the JVM tuning tables based on the hardware and size of your site.

  For information about JVM tuning tables, see Appendix 1, “JVM Tuning Examples,” on page 25.

Related Documentation

SAS Financial Management

For information about installing, administrating, or migrating SAS Financial Management, see the documentation located at

http://support.sas.com/documentation/onlinedoc/fm/

Note: This site is password-restricted. You can find the user name and password in the pre-installation checklist, the Instructions.html, or by contacting SAS Technical Support at http://support.sas.com/techsup/contact/

SAS Intelligence Platform

For information about administering the SAS Intelligence Platform, see the documentation located at

http://support.sas.com/documentation/onlinedoc/intellplatform/index.html

SAS Information Delivery Portal

For information about the SAS Information Delivery Portal, see the documentation located at

http://support.sas.com/documentation/onlinedoc/portal/index.html

SAS Notes

SAS Technical Support develops SAS Notes to inform customers of issues that they need to be aware of when using SAS software. SAS Notes contain additional information about a SAS product and support fixes.

To view SAS Notes for SAS Financial Management, see the product page at

http://support.sas.com/software/products/fm/index.html

On the left side of the product page, select **Samples & SAS Notes** from the menu.

Chapter 2
Completing the First Steps

Overview

This chapter describes the following topics:

- determining your performance objective
- working with JVM options

Determining Your Performance Objective

Before you begin the process of tuning the performance and scalability of SAS Financial Management, ensure that you determine your *performance objective*. A performance objective is typically based on *performance requirements* and *scalability*.

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

Scalability is the ability of a component to adapt readily to a greater or lesser intensity of use, volume, or demand and still meet business objectives. The objective of scaling a component is to increase the component's capacity for growth and increase the speed of the component. Improving the efficiency, or shifting or reducing the load on the component is also an objective of scaling a component.

The hardware that you choose to implement in your environment is a determining factor in meeting your performance objective. Without adequate hardware, achieving the acceptable and consistent performance that you require might not be possible. The SAS Enterprise Excellence Center provides assistance in determining your performance objective, including sizing and identifying the hardware that you need for your environment.

For more information about the services that the SAS Enterprise Excellence Center provides, contact SAS Technical Support at http://support.sas.com/techsup/contact/.

Reviewing and Adjusting JVM Options

In most implementations, the default configuration of a SAS 9.4 web application server meets only the requirements of the initial installation and sample data. You can achieve many performance and scalability improvements by adjusting the JVM options of a web application server.

Note: These tuning options are provided in accordance with the Java Platform Standard, Edition 7, and SAS application-specific options.

In this guide, SAS recommends that you adjust the values for certain JVM options to meet your scalability and performance objective. For the ideal JVM options with which to start the tuning process, see Appendix 1, “JVM Tuning Examples,” on page 25. Before you begin adjusting JVM options, ensure that you review the JVM options that are recommended for your server. Once you have applied the recommended JVM options, test the application, and readjust the options as necessary.

For more information about adjusting JVM options, see the *SAS Intelligence Platform: Web Application Administration Guide*.

For information about even finer performance tuning, see Chapter 4, “Scaling the Middle Tier,” on page 11.

Chapter 3
Configuring the Operating System

Overview

This chapter describes the following topics:

- configuring the operating settings
- configuring TCP parameters
- allocating memory

Configuring the Operating System Settings

The settings of the operating system settings—in particular the virtual memory settings, are often overlooked when tuning for performance. The virtual memory settings are very important, in particular because the move to 64-bit operating systems changed how SAS Financial Management uses the memory settings in Java. This section describes the operating system configuration in a Microsoft Windows environment. For information about tuning the settings for additional types of operating systems, see the *SAS Web Applications: Tuning for Performance and Scalability* document.

In addition, in a Windows environment, you should adjust the Transmission Control Protocol (TCP) settings to account for the higher throughput of the highly granulated SAS Financial Management functions.

Configuring TCP Parameters in a Windows Environment

In the Windows environment, adjusting the TCP settings to account for the higher throughput of the highly granulated SAS Financial Management functions can improve performance.

CAUTION:

Microsoft recommends that you perform a system backup before you edit the registry.

To tune TCP parameters in the Windows registry, complete the following steps:

1. From a command prompt, run *regedit*.
2. In the Registry Editor window, navigate to **HKEY_LOCAL_MACHINE** ⇨ **SYSTEM** ⇨ **CurrentControlSet** ⇨ **services** ⇨ **Tcpip** ⇨ **Parameters**.
3. In the list of parameters, right-click and select **New** ⇨ **DWORD (32–bit) Value**.
4. In the Name column, enter *MaxUserPort* as the name of the new parameter.
5. Double-click the name of the new parameter. The Edit Value window is displayed.
6. In the **Value data** field, specify *65534*.
7. In the **Base** section of the Edit Value window, select **Decimal** and then click **OK**.
8. In the list of parameters, right-click and select **New** ⇨ **DWORD (32–bit) Value**.
9. In the Name column, enter *TcpTimedWaitDelay* as the name of the new parameter.
10. Double-click the name of the new parameter. The Edit Value window is displayed.
11. In the **Value data** field, specify *30*.
12. In the **Base** section of the Edit Value window, select **Decimal** and then click **OK**.

Allocating Java and Virtual Memory

Java allocates and maintains all of its heap allocations (**`-Xms`** and **`-Xmx`**) in virtual memory, which is a page file on Windows and a swap space on UNIX.

For all SAS tiers, SAS recommends that you allocate 1.5 times (and at minimum 1 times) the amount of physical memory as virtual memory allocation. On machines with less available hard drive space, ensure that you allocate more virtual memory.

Note: You must allocate more virtual memory than there is physical memory when you are tuning for performance and you want more than 90% of memory allocated to JVMs.

In addition, to avoid paging losses, ensure that you configure the virtual memory on the drive with the fastest I/O throughput. Setting up the virtual memory correctly can result in the greatest increase in performance, even without modifying the default configuration of the operating system.

For more information about tuning Java and virtual memory, see the “Configuration and Tuning Guidelines for SAS 9 in Microsoft Windows Server 2008” (Paper 370–2011) in the *Proceedings of the SAS Global Forum 2011 Conference* located at

http://support.sas.com/resources/papers/WindowsServer2008ConfigurationandTuning.pdf

Chapter 4
Scaling the Middle Tier

Overview

This chapter describes the following topics:

- changing the default configurations
- tuning the SAS Web Server
- tuning the SAS Web Application Server

Changing the Default Configurations

By default, the settings of the JVMs for SAS Intelligence Platform and SAS Financial Management do not exceed 2 GB. After you install and configure SAS Financial Management, you should tune each instance of the software based on the size of the server on which you installed the software.

See Table 4.1 for a list of the default servers and some of the applications on each of the servers that SAS Financial Management uses.

Table 4.1 Default Servers and the Applications on Each Server That Are Used by SAS Financial Management

Server	Contents
SASServer1	Web Infrastructure Platform (WIP), WIPServices, SAS Environment Manager (Hyperic), the SAS Information Delivery Portal, and SAS BI Dashboard
SASServer3	SAS Financial Management and ODCS (read and write)
SASServer4	ODCS (read only)
SASServer5	SAS Financial Management Web Data Entry

When changing default configurations, note the following:

- On the middle tier, 90% to 95% of the memory is allocated to the JVMs. The remaining memory is reserved for the operating system and administrative clients.
- When the system consumes greater than 95% of memory, more paging than normal might occur. This condition has a negative impact on performance. A greater heap size does not cause more garbage collection or lower performance, although that is a common misconception. In fact, in real-world scenarios with in-memory applications and performance testing in 64-bit operating systems, the opposite has proven to be true.

For information about modifying JVM options, see the *SAS Intelligence Platform: Web Application Administration Guide* located at

http://support.sas.com/documentation/onlinedoc/intellplatform/index.html.

Tuning the SAS Web Server

Changes in SAS 9.4

As part of the SAS 9.4 Web Application Server, SAS 9.4 provides a web server. With the web server in SAS 9.4, an HTTP reverse proxy server is available. The SAS Web Server processes nearly all of the traffic to the SAS Web Application Servers. Therefore, you should tune the server configuration to accept more connections and increase the volume of transactions that are possible with SAS Financial Management.

SAS Web Server Tuning

CAUTION:
Before making any changes to a SAS Web Server configuration file, ensure that you make a backup of each of the configuration files that you plan to change. In addition, ensure that you document all changes that you make for audit and support purposes.

To tune a SAS Web Server, make the described changes to the following configuration files located in the **`MidtierConfigLev/Web/WebServer/conf/extra`** directory:

Configuration File	Description of Changes

http-default.conf	change the value of the MaxKeepAliveRequest parameter to zero
	`MaxKeepAliveRequest=0`
	`ThreadsPerChild=4000`

httpd-mpm.conf	change the values of the ThreadLimit and ThreadsPerChild parameters to 4000 `ThreadLimit=4000`

Access logging is one of the logfiles that is potentially generated by the SAS Web Server. Access logging is an intensive logging process, which can slow down an application. When access logging is not required, you can avoid the possibility of slowing down an application by making the described changes to the following configuration files located in the **`MidtierConfigLev/Web/WebServer/conf`** directory:

Configuration File	Description of Changes
sas.conf	disable CustomLog by inserting the number sign (#) in front of the line to comment it out
httpd.conf	disable CustomLog by inserting the number sign (#) in front of the line to comment it out

For additional troubleshooting and debugging issues, contact SAS Technical Support at http://support.sas.com/techsup/contact/.

Tuning the SAS Web Application Server

Changes in SAS 9.4

SAS 9.4 and later includes an embedded middle-tier server called the SAS Web Application Server. The SAS Web Application Server is based on the VMware vFabric tcServer.

Note: With SAS 9.4 and later, SAS no longer supports external third-party application servers such as JBoss, WebSphere, and WebLogic.

Tuning the SAS Web Application Server

CAUTION:

Before making any changes to a SAS Web Application Server configuration file, ensure that you make a backup of each file that you plan to change. In addition, ensure that you document all changes that you make for audit and support purposes.

To tune a SAS Web Application Server, complete the following:

- For higher concurrent user loads, increase the number of SAS Services HTTP connections for all of the associated web application servers (SASServer1, SASServer3, and SASServer4).

  To increase the number of connections, make the following changes in the wrapper.conf file located in the **/conf/** directory:

  ```
  Dsas.svcs.http.max.total.connections=512
  ```

  ```
  Dsas.svcs.http.max.connections=512
  ```

- The value for the **maxIdleTime** attribute in the server.xml file is the maximum number of threads in the executor thread pool that are used to process connection requests. The server.xml file is located in the **SAS-configuration-directory\Levn\Web\WebAppServer\SASServer1_n\conf\** directory.

  If you observe excessive spikes in the number of connection requests, then increase the value of the **maxIdleTime** attribute to the executor line as shown in the following example:

  ```
  Executor maxThreads="300" minSpareThreads="50" maxIdleTime="600"
  name="tomcatThreadPool" namePrefix="tomcat-http--"
  ```

  The default value is 300; the recommended value is a value between 300 and 1024.

Note: If you have multiple instances of a SAS Web Application Server, ensure that you make the same changes for each of the servers that you want to tune.

For any additional troubleshooting and debugging issues, contact SAS Technical Support at http://support.sas.com/techsup/contact/.

Using Multiple Query Processors to Improve Performance

Multiple Query Processors Overview

A multiple query processor configuration is most effective in environments in which there are many concurrent users executing small queries at the same time. Although a multiple query processor configuration does not improve the performance of an individual query, it enables more queries to be executed simultaneously, which improves overall throughput.

Setup and Tuning Considerations

For information about how to implement multiple query processors, see the *SAS Financial Management: System Administration Guide*.

Troubleshooting a Query Processor Configuration

Currently, query processors do not support significant debug logging. For information about configuring logging for a query processor, see the *SAS Financial Management: System Administration Guide*.

If problems occur with a query processor, then you can increase the logging priority level to **debug** in the query processor’s log4j.xml file.

To increase the logging level of a query processor, complete the following steps:

1. On the mid-tier server locate the log4j.xml file that corresponds to the query processor for which you want to increase the logging level.
2. Using a text editor, open the log4j.xml file.
3. Modify the logging **`level value`** parameter for the query processor. For example, to increase the logging to **`debug`** for **`com.sas.solutions.odcs.processor.QueryProcessor`**, find the statement in the log4j.xml file that resembles the following example:

   ```
   <logger name="com.sas.solutions.odcs.processor.QueryProcessor" additivity="false">
     <level value="warn"/>
     <appender-ref ref="SAS_FILE"/>
   </logger>
   ```

   Note: If you do not see an entry similar to the statement above, you can add it directly to the file.

4. Change the value of **`level value`** parameter to **`debug`** as shown in the following example:

   ```
   <logger name="com.sas.solutions.odcs.processor.QueryProcessor" additivity="false">
     <level value="debug"/>
     <appender-ref ref="SAS_FILE"/>
   </logger>
   ```

5. Reset the system for your changes to take effect.
6. To determine whether your troubleshooting efforts were successful, re-create the activity that generated the unexpected results for the query processor and view the log output.

   Note: Leaving an increased logging level turned on for too long can generate too much information, which can consume system resources. Therefore, ensure that you reset the logging level to its previous state (for example, **`warn`**) after you have completed your troubleshooting.

Additional logging for query processors also occurs in the SAS Financial Management read and write server log and in the ODCS read-only server log on your mid-tier machine.

For any additional troubleshooting and debugging issues, contact SAS Technical Support at http://support.sas.com/techsup/contact/.

Chapter 5

Optimizing the SAS Web Infrastructure Platform Data Server

Overview

This chapter describes the following topics:

- optimizing the platform data server default configurations
- modifying memory usage
- tuning and monitoring for performance

Optimizing the Platform Data Server

The SAS Web Infrastructure Platform Data Server provides a transactional store that is used by SAS mid-tier software, including SAS Financial Management. The server is based on PostgreSQL 9.1.9. SAS configures a single server instance, and the SAS Web Application Server instances are configured with JDBC data sources that access the server.

Before optimizing the platform data server for SAS Financial Management in your environment, ensure that you complete the following tasks:

- back up the data server database
- make a backup of the default PostgreSQL configuration file (postgresql.conf) that resides in the data directory

This chapter documents the deviations or explicit recommendations for SAS Financial Management. For detailed information about optimizing the Web Infrastructure Platform Data Server, see the “Tuning and SAS Web Infrastructure Platform Data Server” chapter in *SAS Web Applications: Tuning for Performance and Scalability*.

Modifying Memory Usage

In environments with a medium to high transaction level, you need to modify the postgresql.conf file to use the settings recommended for a “Large Database” with the exception of the **max_connection** and **max_prepare_transaction** parameters. Based on the size of your environment, set the value for each of these parameters as shown in the following example:

```
max_connections=1500
max_prepared_transactions=1500
```

where:

- **max_connections**—sets the maximum number of concurrent connections to the database server. The maximum value is 1500 connections.
- **max_prepared_transactions**—sets the maximum number of transactions that can be in a “prepared” state at the same time. The maximum value is 1500 connections.

Note: The postgresql.conf file settings that are recommended for a “Large Database” are documented in the “Tuning and SAS Web Infrastructure Platform Data Server” chapter in *SAS Web Applications: Tuning for Performance and Scalability*.

In addition, consider the values of the following parameters (also located in the postgresql.conf file) when tuning the data server:

- **shared_buffers**—specifies the total amount of memory that PostgreSQL uses for caching data. If there is additional memory available on the system, you can increase the value of the **shared_buffers** parameter to accommodate the size of the database on the disk, at minimum.
- **work_mem**—specifies the amount of memory that is available to use for sorts, hashing, and materialization before writing to temporary disk files. Several running sessions can perform these operations concurrently. Therefore, the total memory that is used might be greater than the value of the **work_mem** parameter. Keep this in mind when specifying a value for the **work_mem** parameter. SAS recommends that you set the value for the **work_mem** parameter between 16 MB and 64 MB. For specialized use cases such as frequent and very large sorts, you can specify an even larger value for the **work_mem** parameter.

If the platform data server is installed on the data tier, ensure that you consider the other demands on the data tier when you specify values for the postgresql.conf parameters.

Note: After updating the postgresql.conf file, restart the system to implement your changes.

Tuning and Monitoring for Performance

The SAS Web Infrastructure Platform Data Server (PostgreSQL) benefits from the same sort of system tuning as SAS, which includes the following:

- a fast disk resource
- plenty of memory for caching

- a sufficient number of cores with good single thread performance
- an operating system tuned to favor I/O

Tuning the platform data server might make a difference in performance. However, do not be surprised if tuning the platform data server configuration does not change the performance. Typically, there is little benefit in tuning the platform data server configuration because PostgreSQL automatically adjusts to use the available hardware in the most efficient way. Even the default settings handle a wide variety of use cases efficiently.

If the platform data server is detected as, or suspected to be, a performance bottleneck, there are many tools included with SAS that you can use for troubleshooting.

For more information about performance tuning and monitoring, see the “Tuning and SAS Web Infrastructure Platform Data Server” chapter in *SAS Web Applications: Tuning for Performance and Scalability*.

Chapter 6
Reviewing Miscellaneous Topics

Overview

This chapter describes the following topics:

- monitoring performance and tuning Java
- preloading ODCS caches
- troubleshooting

Monitoring Performance and Additional Java Tuning

The recommendations in this chapter have been field-tested and performance-tested. The recommendations represent development, test experience, and customer experience. Consider these recommendations as a postconfiguration task after you install and configure SAS Financial Management.

Note: The tuning techniques described in this document are just a starting point. Each deployment site is unique. The new data, forms, user concurrency of each customer site adds a level of complexity to the performance monitoring and tuning process.

When monitoring performance and making additional Java adjustments, note the following:

- Java tuning is a continual operation that you should monitor. Typically, it is optimal if a server uses 60% to 70% of the JVM during normal usage. This level of usage leaves some memory available for periods of heavier usage.
- The page file virtual memory recommendation for Microsoft Windows is 1 to 1.5 times the RAM (minimum) and 3 times the RAM (maximum).

Note: Anything under the used Java heap sizes results in performance degradation.

- In most cases, permanent generation (permgen) is elevated above necessary usage. Monitor the permgen usage on SASServer3 and SASServer4. If the usage is significantly lower during a typical restart, then you can reduce its maximum size to 256 MB to 512 MB above the maximum usage that you are observing.

Note: There are many tools that you can use for Java tuning. You can also do most of this monitoring with a basic system administration tool. In addition, you can use a combination of small tools such as the Process Explorer and Task Manager with the Web Application Server log and Heap tool for Java tuning. These tools are included with the SAS Web Application Server.

Preloading ODCS Caches

After restarting a server or loading data into the database, the first login is slower because the new information is cached and the files are compiled. This is normal behavior for you to expect after scheduled outages.

Nightly and even weekly restarts are no longer necessary. Even backup processes no longer bring all systems down; they merely pause the current system state to prevent users from making changes during the backup process.

To alleviate the first load delay and to prime all cache at start-up, you have the following two options:

1. For restarts, you can prime all data at start-up by adding the following option to the Java arguments that are located in the wrapper.conf file for both SASServer3 and SASServer4:

   ```
   -Dodcs.preload.vcubes="1,2,3"
   ```

   This option configures the preloading of Java virtual cubes (v-cubes) at start-up.

2. The first option is suitable for loading. However, it is not threaded. Threading can take a significant amount of time when there are a lot of cubes or a large number of formulas.

   The second option to alleviate a first–load penalty is to use the thread LoadCache program. The LoadCache program can load all cubes across SASServer3 and SAS Server4 at the same time on systems that have enough threads available.

   By using the LoadCache program, you can potentially reduce the total load time to just seconds more than your longest cube load. The LoadCache program is also ideal for calling at the end of any operation that invalidates a cube, such as data loads. Using the LoadCache program is superior to other operations like FMQuery. Therefore, using the LoadCache program is the preferred method. The overhead for running on already loaded cubes is minimal with typical runtimes of less than 500 ms per thread.

   For information about using the LoadCache program, see “SAS Usage Note 53842” located at

   http://support.sas.com/kb/53/842.html

Troubleshooting

The symptoms that you see are rarely the source of the problem. Therefore, never make any assumptions when you are troubleshooting a problem. Troubleshooting is a logical method that uses a scientific method. The *process of elimination* is the easiest method to use.

To begin the troubleshooting process, look at logs. Logs almost always contain the answer to a problem. Even if the logs do not contain an answer to a problem, they can help lead you to the answer to a problem based on timing. Most problems that you might encounter are solved with default logging levels and most problems have most likely been seen before. Therefore, ensure that you include all logs in every track that you open.

The logs that you should include in every track include the following:

- web application logs located under **`config_lev/Web/Logs`**. The web application level logs are the most important logs.
- web application server logs for each server located under **`config_lev/Web/Applications`**. Server–side logs typically do not contain valid application level messages. Typically, web application server logs are useful only when application server problems occur at start-up or when connection problems occur at the operating system or network level.
- SAS web server logs located under **`config_lev/Web/WebServer/logs`**. The logs directory contains all of the access logs and error logs for the time period in question.
- client–tier logs for the Stored Process Server, the Workspace Server, and those created by the Object Spawner by using a Metadata Server configuration file. By default, these logs are located on the client tier in the following directories:

  ***sas-config-lev*\SASApp\StoredProcessServer\Logs**

  ***sas-config-lev*\SASApp\WorkspaceServer\Logs**

  ***sas-config-lev*\SASMeta\MetadataServer\Logs**

Appendix 1
JVM Tuning Examples

Overview

You can use this appendix as a “cheat sheet” for general recommendations based on the products implemented at a site and the size of a site. Each scenario provides examples from the minimum requirement of 32 GB of memory to 256 GB of memory for maximum system utilization and performance.

If an environment has more resources than are documented in this appendix, then double the settings of the table that is the closest to half of the size of the environment. Allocate any excess memory to SASServer3 and SASServer4. For example, if you have 512 GB of memory on one system, double each setting in the FM-256 table to set the system potential to maximum utilization.

This appendix describes the following topics:

- using a balanced approach with SAS BI considerations
- using SAS Financial Management and ODCS with a single middle tier
- using SAS Financial Management and ODCS with a split middle tier

Using a Balanced Approach with SAS BI Considerations

For tables that list the best settings for memory variables for SAS Financial Management and ODCS with a single middle tier, see “Using SAS Financial Management and ODCS with a Single Middle Tier” on page 26.

If you have higher SAS BI or SAS Web Infrastructure Platform usage on SASServer1, then consider that SASServer1 might need more memory to ensure the best performance of the applications that are running on it.

With 64 GB of total system memory, there is not a lot of spare system memory. Therefore, you should increase only the other JVM settings as needed. For example, if SASServer1 is running at the maximum memory, then increase it by only 1 GB to 2 GB on the minimum and maximum heap size (`-Xms` and `-Xmx`) at a time.

Note: Ensure that you reallocate the memory in even increments from SASServer3 and SASServer4.

Using SAS Financial Management and ODCS with a Single Middle Tier

After a full server restart or after loading data into the database, the first login is slower. The first login is slower because the data has to be refreshed from the database to the SASServer3 read/write, the SASServer4 read, and the ODCS in-memory caches. To minimize this delay, you can configure the ODCS start-up parameter as documented in Chapter 6, “Reviewing Miscellaneous Topics,” on page 21.

The following values for JVM parameters are based on the primary memory usage that is typical with SAS Financial Management. Unlike previous versions of SAS Financial Management, there are now two primary caches for cube information. If SASServer3 usage is low and the ODCS usage is high, you can decrease the SASServer3 memory to 1 GB to 2 GB above the maximum observed usage. Then, you can reallocate the remaining memory to SASServer4.

When ODCS requires more memory, another option is to reallocate memory from other servers, in particular from SASServer1 and SASServer2, which moves those servers closer to their default settings. Typically, the SAS Web Infrastructure Platform and SAS Web Report Studio do not need higher resources, but overall system performance might benefit from a moderate increase in memory.

The values in Table A1.1 assume mostly SAS Financial Management usage in very small customer sites. Very few customers can run with 32 GB.

Table A1.1 *Table FM-32 — 32 GB.*

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	2 GB (1 GB minimum) Page file: 48 GB minimum	
SASServer1	3 GB / 4 GB	1 GB / 1.5 GB
SASServer2	Default	Default
SASServer3	8 GB / 11 GB	512 MB / 1 GB
SASServer4	8 GB / 11 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

The values in Table A1.2 assume SAS Financial Management usage with additional balancing toward more heavily used web applications in small to medium-small customer sites.

Table A1.2 *Table FM-48 — 48 GB.*

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	2 GB (2 GB minimum) Page file: 72 GB minimum	
SASServer1	3 GB / 5 GB	1 GB / 1.5 GB
SASServer2	Default	Default
SASServer3	8 GB / 19 GB	512 MB / 1 GB
SASServer4	8 GB / 18 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

The values in Table A1.3 assume medium-sized customer sites with 1 to 2 active cycles.

Table A1.3 *Table FM-64*

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	3 GB (2 GB minimum) Page file: 96 GB minimum	
SASServer1	4 GB / 6 GB	1 GB / 2 GB
SASServer2	Default	Default
SASServer3	16 GB / 27 GB	512 MB / 1 GB
SASServer4	16 GB / 24 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

Table A1.4 Table FM-96 — 96 GB.

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	3 GB (2 GB minimum) Page file: 144 GB minimum	
SASServer1	4 GB / 8 GB	1 GB / 2 GB
SASServer2	Default	Default
SASServer3	16 GB / 42 GB	512 MB / 1 GB
SASServer4	16 GB / 38 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

Table A1.5 Table FM-128 — 128 GB.

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	4 GB (2 GB minimum) Page file: 192 GB minimum	
SASServer1	4 GB / 8 GB	1 GB / 2 GB
SASServer2	Default	Default
SASServer3	16 GB / 58 GB	512 MB / 1 GB
SASServer4	16 GB / 54 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

Table A1.6 Table FM-160 — 160 GB.

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	4 GB (2 GB minimum) Page file: 240 GB minimum	
SASServer1	4 GB / 8 GB	1 GB / 2 GB

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
SASServer2	Default	Default
SASServer3	16 GB / 74 GB	512 MB / 1 GB
SASServer4	16 GB / 70 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

Table A1.7 *Table FM-256 — 256 GB.*

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	4 GB (2 GB minimum) Page file: 240 GB minimum	
SASServer1	4 GB / 8 GB	1 GB / 2 GB
SASServer2	Default	Default
SASServer3	16 GB / 122 GB	512 MB / 1 GB
SASServer4	16 GB / 118 GB	512 MB / 1 GB
SASServer5	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default

Using SAS Financial Management and ODCS with a Split Middle Tier

The following values are based on the primary SAS Financial Management application on the middle tier and the ODCS read-only server (SASServer4) split onto its own tier for additional resources for threads and memory. In addition, the values are based on the assumption that the ODCS tier has enough available resources to match the main middle tier, or at least enough to match the SASServer3 JVM requirements for an extra ODCS cache of equal size. A split middle tier configuration does not require many changes and only one example is needed to show the simplicity of this setup.

The values in Table A1.8 assume 96 GB for both SAS Financial Management and the ODCS.

Table A1.8 Table FM/ODCS-96.

Target Allocation Server and OS	Memory and Virtual Memory Allocations	PermSize / MaxPermSize
Operating System, Admin Server, and NodeManager	3 GB (2 GB minimum) Page file: 24 GB minimum	
SASServer1 (Mid-tier 1)	4 GB / 8 GB	1 GB / 2 GB
SASServer2 (Mid-tier 1)	Default	Default
SASServer3 (Mid-tier 1)	16 GB / 80 GB	512 MB / 1 GB
SASServer4 (Mid-tier 1)	16 GB / 92 GB	512 MB / 1 GB
SASServer5 (Mid-tier 1)	Default (Turn off and, if not using Web Data Entry, evenly distribute between SASServer3 and SASServer4.)	Default