Paper 270-2009

Maximizing the Performance of Your SAS® Solution: Case Studies in Web Application Server Tuning for n-Tier SAS Applications

Nicholas Eayrs, Tanya Kalich, Graham Lester, and Stuart Rogers,
SAS Institute Inc., Cary, NC

ABSTRACT

Are there changes you could make in your Java Web Application Server that would make your SAS® solution more responsive, more robust, or able to handle a larger client load? This presentation highlights the settings that are most likely to affect the smooth functioning of your SAS solution and explains the concepts behind those settings. Case studies from SAS Technical Support are used to illustrate a broad range of potential pitfalls, narrowly avoided disasters, and wild successes. The presentation also delves into the troubleshooting techniques you need in order to identify problems in Java Web Application Server tuning and to pinpoint the best solutions.

INTRODUCTION

Due to the breadth and depth of aspects covered in this presentation, the paper portion of the presentation has been divided into two distinct components. The first component is this short summary, which is included in the event proceedings. The second component is the full text of the paper, which is available on the SAS Customer Support Web site. This summary provides a description of what is included in the full-text paper and a summary of the research done to support the presentation.

FULL-TEXT PAPER

The paper provides an overview of the three Java Web Application Servers that are supported in SAS®9software:

- Apache Tomcat
- BEA WebLogic
- IBM WebSphere

Note: The latest baseline support statement for Java Web Application Servers is available at **support.sas.com/resources/thirdpartysupport/baseline_plus.html**.

Throughout the paper, the theory of what can be tuned or changed is presented, followed by specific case studies that highlight the benefits and potential pitfalls of each area. These case studies are based on real-world experiences from SAS Technical Support in order to ensure accurate illustrations for your benefit.

The body of the paper contains the following sections:

- “Overview of Supported Java Web Application Servers”—Documents the key features of all three servers. This section begins with a table that summarizes all of the key features of the Java Web Application Servers, and then it describes these features in more detail.
- “Real-Time Monitoring of a Running Java Virtual Machine”—Describes some of the tools that are available for real-time monitoring of a Java virtual machine (JVM). These tools (jvmstat, IBM WebSphere Performance Monitoring Infrastructure, and JConsole) are supported by the Sun 1.4.2 JVM, the IBM 1.4.2 JVM and the IBM 1.5 JVM.
- “Improving a SAS Solution’s Responsiveness and Robustness”—Examines methods that are required to make the single Java Web Application Server more responsive and more robust. These methods include reviewing timeouts and relevant configuration elements within each of the three supported Java Web Application Servers. Timeout values are a common pitfall area because they are often adjusted without any root cause analysis and are most likely evidence of a yet-to-be-discovered issue elsewhere within the architecture. This section also provides an applicable timeout case study.
- “Java Memory”—Explores the definitions of Java memory, providing a table with initial starting values for each JVM. The table is followed by a detailed discussion of how memory is assigned, limitations in the way memory is assigned, and how this affects both the responsiveness and robustness of SAS solutions.
- “Memory Allocation Case Studies”—Builds upon the previous section by presenting detailed memory-allocation case studies.
- “Memory Management”—Details key command-line options for memory management. The section discusses process-based garbage-collection options as well as garbage-collection logging options. It also contains tables that highlight initial starting values for size. The section also provides a number of case studies.

- "Threading"—This section documents each of the supported Java Web Application Servers and their associated threading configuration elements. As with timeouts, threading errors are usually indicative of an underlying issue elsewhere within the architecture. Two threading case studies are illustrated in this section.
- "Scalability: Handling a Larger Client Load"—Investigates methods for enabling a Java Web Application Server to handle a larger client load. This part of the investigation focuses on scaling the Java Web Application Server vertically on a single machine, horizontally across a number of machines, and via the deployment of additional components such as HTTP servers and load-balancers. Several scalability case studies are illustrated, one of which is an extended case study directly from a client (Verizon Business).

The full-text paper is available on the SAS Customer Support Site at **support.sas.com/saspresents**.

Despite the complexity of the information and techniques presented in the full-text paper, remember to always start with the simplest solutions. As a case in point, perhaps the way you use your SAS solution is the underlying problem rather than the technology. For example, a trillion-row report in SAS® Web Report Studio never runs instantaneously. You need to consider whether it is really necessary to run such reports in real time. Can you run the reports in batch overnight and then view the static results later? Is it possible to split the user community and have people access different middle tiers?

RESEARCH SUMMARY

In our research, we found that underlying differences in technology have a major impact on the methods for improving the responsiveness and robustness of a SAS solution in addition to increasing a system's ability to handle a greater client load. The options that are available for tuning and scaling a Java Web Application Server are affected by several factors, including the fact that not all SAS solutions are supported across all three Java Web Application Servers and the fundamental differences in the JVM implementations between Sun and IBM.

We discovered that while timeouts are important for the smooth running of a Java Web Application Server, they should not always be viewed as the simplest solution to performance problems. Rather, you should view changing the timeouts as a last resort or as a temporary solution to enable your system to continue working while you investigate the underlying problems further. A good understanding of what is happening during the faulty behavior is important to successfully using the timeouts to solve a business issue. Quite often, a timeout issue is merely a symptom of another issue.

With regard to improving the responsiveness and robustness of a SAS solution, we examined first the allocation of memory to the Java Web Application Server. This examination included the importance of operating-system limitations on the amount of memory that can be allocated and possible methods of alleviating these limitations. However, any Microsoft Windows 32-bit environment is always hampered, to some extent, by the operating system's handling of memory. We identified how to set both the starting and the maximum memory allocation and discussed how these allocations impact Java Web Application Servers. We found that bigger is not always best, as excessive paging of the memory that is allocated to a Java Web Application Server severely affects performance.

The next key performance area we examined is the underlying method of automatic memory management for JVMs. We investigated both the Sun generational and IBM monolithic models for memory management. For each model, we identified the key command-line options that impact the operation of the garbage-collection algorithms and the methods for altering the algorithm that is used. This process is complex and it is important to gain a clear understanding of what occurs in the garbage-collection logs, because the logs can help you understand many of the causes for performance issues with Java Web Application Servers.

Research with the Sun JVM showed that size is everything. The generational model splits the Java heap into a number of different spaces, and correctly sizing these spaces for applications is fundamental. A similar statement is true for the IBM JVM. Despite the difference in model, size is still important. With insufficient space within a kCluster storage area, the Java heap can become fragmented and impact performance. For both the Sun and the IBM JVMs, it is vital to start any tuning exercise with the initial settings that are suggested by extensive SAS research.

Next, we reviewed the impact of threading on the performance of the Java Web Application Servers. This research showed the need for caution because excessive values as well as reciprocally small values can have considerable negative results. Before you tune threading values, you need to fully understand the hardware constraints or opportunities and the impact that the operating system might have on these factors. In everything from the hardware to the business application, any threading capability must be a compilation of these factors.

Finally, we examined how to enable a Java Web Application Server to handle a large client load. We explored the differences between horizontal and vertical scaling and found that scaling is not just a single answer. A multitude of options are available for all sorts of business requirements, and any scaling decision should be firmly based around solving such business scenarios.

Because of the complexity of issues that surround some of the aspects covered in this presentation, we recommend that you always implement a SAS Architecture Review. The last extended case study presented in the full-text paper highlights the effectiveness of involving SAS Professional Services in such an activity.

CONTACT INFORMATION

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

Nicholas Eayrs
SAS Institute Inc.
Wittington House
Henley Road
Marlow SL7 2EB
Work Phone: +44 1628 4-86933
Email: Nicholas.Eayrs@sas.com

Tanya Kalich
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919-531-4490
Tanya.Kalich@sas.com

Graham Lester
SAS Institute Inc.
Wittington House
Henley Road
Marlow SL7 2EB
Work Phone: +44 1628 4-90454
Graham.Lester@sas.com

Stuart Rogers
SAS Institute Inc.
Wittington House
Henley Road
Marlow SL7 2EB
Work Phone: +44 1628 4-90613
Stuart.Rogers@sas.com