IBM SPSS Collaboration and Deployment Services 5.0 on IBM POWER8 and IBM AIX 7.1

Best practices for performance optimization and scalability

Mayank A Patel, and Thomas Kochie
IBM SPSS Collaboration and Deployment Services Product Team

Bill Phu
IBM Systems and Technology Group POWER Performance

November 2014

@IBMSystemsISVs

Table of contents

Abstract **1**
Executive summary **1**
Introduction **2**
Overview of IBM SPSS Collaboration and Deployment Services **2**
Clustering **3**
Workload aspects **4**
System specification, software used, and topology **4**
Server system specifications and software 4
Client workload driver system specifications and software 5
Topology 5
Clustered IBM SPSS Collaboration and Deployment Services **6**
Test methodology **7**
Test workload 7
Performance concepts in Power Systems **8**
The POWER8 processor chip 8
Logical processors 8
Cache affinity and processor binding 9
Memory affinity 9
Alternatives to explicit processor binding and memory affinity 9
Best practices and recommended tuning to optimize Collaboration and Deployment Services **10**
Tuning the operating system 10
Tuning Java heap and the garbage collection policy 10
Tuning memory allocation 11
Tuning resource sets and memory affinity 11
Tuning the application server and the application 12
Results of running Collaboration and Deployment Services on POWER8 .... **13**
Benchmark results: Collaboration and Deployment Services scoring service 13
Competitive comparison **14**
Summary **15**
Resources **16**
About the authors **17**
Trademarks and special notices **18**

Abstract

This paper describes the results of performance evaluation, optimization, and best practices of IBM SPSS Collaboration and Deployment Services 5.0 on an IBM POWER8 processor-based server running IBM AIX 7.1. IBM conducted this evaluation of SPSS Collaboration and Deployment Services on a single server environment to distill optimum performance tuning options for best practices described in this white paper. Majority of content of this paper, such as examples, workload description, test methodology, and tuning best practices were all derived from a previous paper detailing performance results running on a POWER7+ processor. Refer to the "Resources" section for the location of that paper.

Executive summary

This paper describes the methods and settings that influence the performance of the IBM® SPSS® Collaboration and Deployment Services 5.0 software on an IBM POWER8™ processor-based server running IBM AIX® 7.1 platform. The goal was to optimize SPSS Collaboration and Deployment Services real-time scoring services to deliver the optimal scoring service transaction rates on an IBM WebSphere® Application Server (WAS) cluster hosted on a single server. This paper is intended for businesses that are evaluating or those that are already running SPSS Collaboration and Deployment Services on IBM Power Systems™ today. The performance optimization and scalability results documented in this paper provide insights into the performance characteristics of deploying the solution with the recommended approaches on AIX environment. An overview of IBM SPSS Collaboration and Deployment Services software and the system specifications of the test environment are also provided for the benefits of the readers. The best practices provided in this paper define optimization and scoring scalability in different partition size profiles with stand-alone and multiple WebSphere instances deployed on a multi-core Power Systems server.

Key findings

- Collaboration and Deployment Services can take advantage of memory allocation tuning, Java virtual machine (JVM) tuning, cache affinity (through processor binding) and memory affinity settings to significantly improve scoring throughput.
- For systems using POWER8 processors, it is recommended that two cores per WebSphere Application Server instance be used for optimal performance.
- The 24-core POWER8 processor-based server delivered 1.8 to 2.3 times more scoring throughput per second than a similarly configured Intel® Ivy Bridge system.
- Linear scaling can be achieved by properly tuning system and balancing the workload across processor cores.

Introduction

IBM SPSS Collaboration and Deployment Services is an enterprise-level application that enables widespread use and deployment of predictive analytics. Its performance on IBM Power Systems servers with the AIX operating system environment can be significantly enhanced through following a few well documented best practices. This paper describes how IBM tested, analyzed, and optimized the performance of Collaboration and Deployment in a laboratory environment.

The information in this white paper has been gathered during customer representative performance tests. The tests were conducted on different partition size profiles which demonstrate how to maximize the scores per second and determine the scaling factor for the environment. The purpose of running Collaboration and Deployment Services in multiple WebSphere Application Server environments is to determine the scaling characteristics and help provide recommendations to the customer. The details of the multiple WebSphere instances on a cluster environment are explained in the "System specification, software used, and topology" section.

Overview of IBM SPSS Collaboration and Deployment Services

IBM SPSS Collaboration and Deployment Services provide centralized, secure, and auditable storage of analytical assets and advanced capabilities for management and control of predictive analytic processes, as well as sophisticated mechanisms for delivering the results of analytical processing to the users. The following are benefits of IBM SPSS Collaboration and Deployment Services:

- Safeguarding the value of analytical assets
- Ensuring compliance with regulatory requirements
- Improving the productivity of analysts
- Minimizing the IT costs on managing analytics

IBM SPSS Collaboration and Deployment Services allow your business to securely manage diverse analytical assets and foster greater collaboration among those developing and using them. Furthermore, the deployment facilities ensure that the right people get the necessary information that they need to take timely, appropriate action.

IBM SPSS Collaboration and Deployment Services provide the enterprise platform for integrating SPSS analytical products into complete business solutions. For example, IBM SPSS Collaboration and Deployment Services is a prerequisite for IBM SPSS Decision Management tools, (at: **ibm.com**/software/analytics/spss/products/deployment/decision-management/) the suite of customizable predictive applications for insurance, banking, and market research verticals, including fraud detection, credit risk evaluation, and campaign effectiveness analysis.

Clustering

Collaboration and Deployment Services clustering can be used to provide reliability and scalability. Each application server in the cluster has an identical configuration for the hosted application components and the repository can be accessed through a hardware or software-based load balancer. Clustering enables scaling by using multiple application servers, also reducing the potential of outages from maintenance or hardware or software component failures. A typical clustered Collaboration and Deployment Services (as shown in Figure 1) topology uses multiple physical systems, however, the test environment created for performance study of Collaboration and Deployment Services was on a single physical system with a single AIX partition.

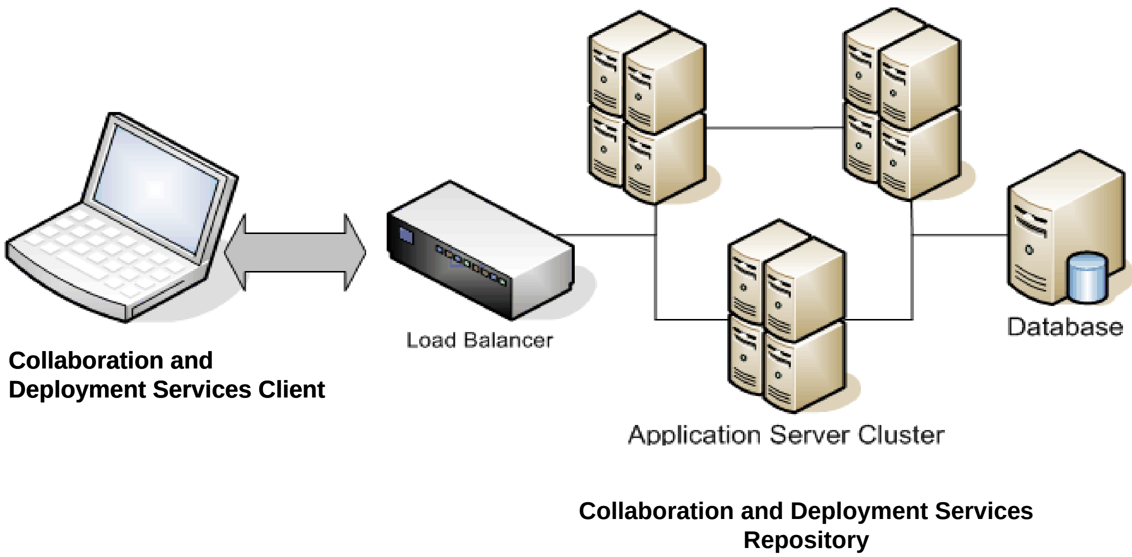

Figure 1: Collaboration and Deployment Services clustering with WebSphere Application Servers

Collaboration and Deployment Services currently supports WebSphere and WebLogic Application Server clustering; however, the testing environment demonstrated in this white paper was set up with WebSphere Application Servers. Deploying Collaboration and Deployment Services into an application server cluster includes a number of additional configuration steps. For more information, refer to the *Clustering* topic in the document **RepositoryConfig.pdf** from the Collaboration and Deployment Services product documentation.

Workload aspects

The objective and goal of Collaboration and Deployment Services performance testing was to maximize real-time scoring service transaction rates using POWER8 and AIX 7.1 settings. Real-time scoring service enables the delivery of on-demand scores to applications, making it extremely easy to deploy analytical scores within business processes. Scoring models can be created within SPSS Modeler, and then stored in the platforms repository. Scoring models are used in a variety of ways, and here are some examples:

- Customer retention analytics: predict opportunities to improve customer retention
- Operational analytics: make operational improvements in real time that maximize productivity and profitability
- Threat and fraud analytics: monitor your business environment and detect suspicious activity to minimize loss

Generally, scoring is implemented as a probability calculation based on a number of variables. The infrastructure allows an outside application or user to query the server through the web services interface and receive a score back in real time.

System specification, software used, and topology

This section describes the test environment used for the performance study, providing details about the hardware and software. The tests were conducted on an IBM Power® S824 server and the OS level was AIX 7.1. An AIX 7.1 partition was created using the internal Small Computer System Interface (SCSI) associated with the server. No external storage was used. The software used was an IBM software stack to showcase a complete IBM software and hardware solution. IBM SPSS Collaboration and Deployment Services Server in a standalone and cluster configuration with all prerequisite software were installed on a single AIX partition as depicted in Figure 2. The following section describes the system specifications and software configuration details and test topology.

Server system specifications and software

The test environment used the following hardware devices:

- Main system for Collaboration and Deployment Services – IBM Power S824 server, containing 2 3.5 GHz POWER8 processor modules with 12 cores each (24 cores in total) with dynamic power savings set to favor performance
  - 24 core partition
  - 256 GB physical memory
  - SMT8 mode
  - Internal SCSI storage
  - One AIX logical partition (LPAR)
  - Connnectivity through 1GB Ethernet

This section describes the major server software components that were used:

- IBM AIX 7.1
- IBM WebSphere Application Server Network Deployment 8.0.0.5
- IBM Hypertext Transfer Protocol (HTTP) Server 8.0

- IBM DB2® 9.7
- IBM SPSS Collaboration and Deployment Services Server 5.0
- IBM SPSS Modeler Adapter 15.0
- IBM SPSS Decision Management 7.0
- IBM Java Runtime Environment 1.6

Client workload driver system specifications and software

For the POWER8 results, one IBM System x® iDataPlex® model 7912 server using the Intel Xeon® 6x2x2 core processor X5650 at 2.67 GHz running Red Hat Enterprise Linux (RHEL) 6.5 was used to drive the workload against the target POWER8 server.

For the Ivy Bridge results, one IBM Flex System® x220 Compute Node using the Intel Xeon 2x8 core processor E5-2680 at 2.7 GHz running RHEL 6.5 was used to drive the workload against the target Ivy Bridge server.

For both sets of results, the workload was generated using soapUI 4.5.1.

Topology

The test topology in this white paper uses the Collaboration and Deployment Services 5.0 configured with multiple WebSphere instances deployed to form the WebSphere Application Server cluster.

Clustered IBM SPSS Collaboration and Deployment Services

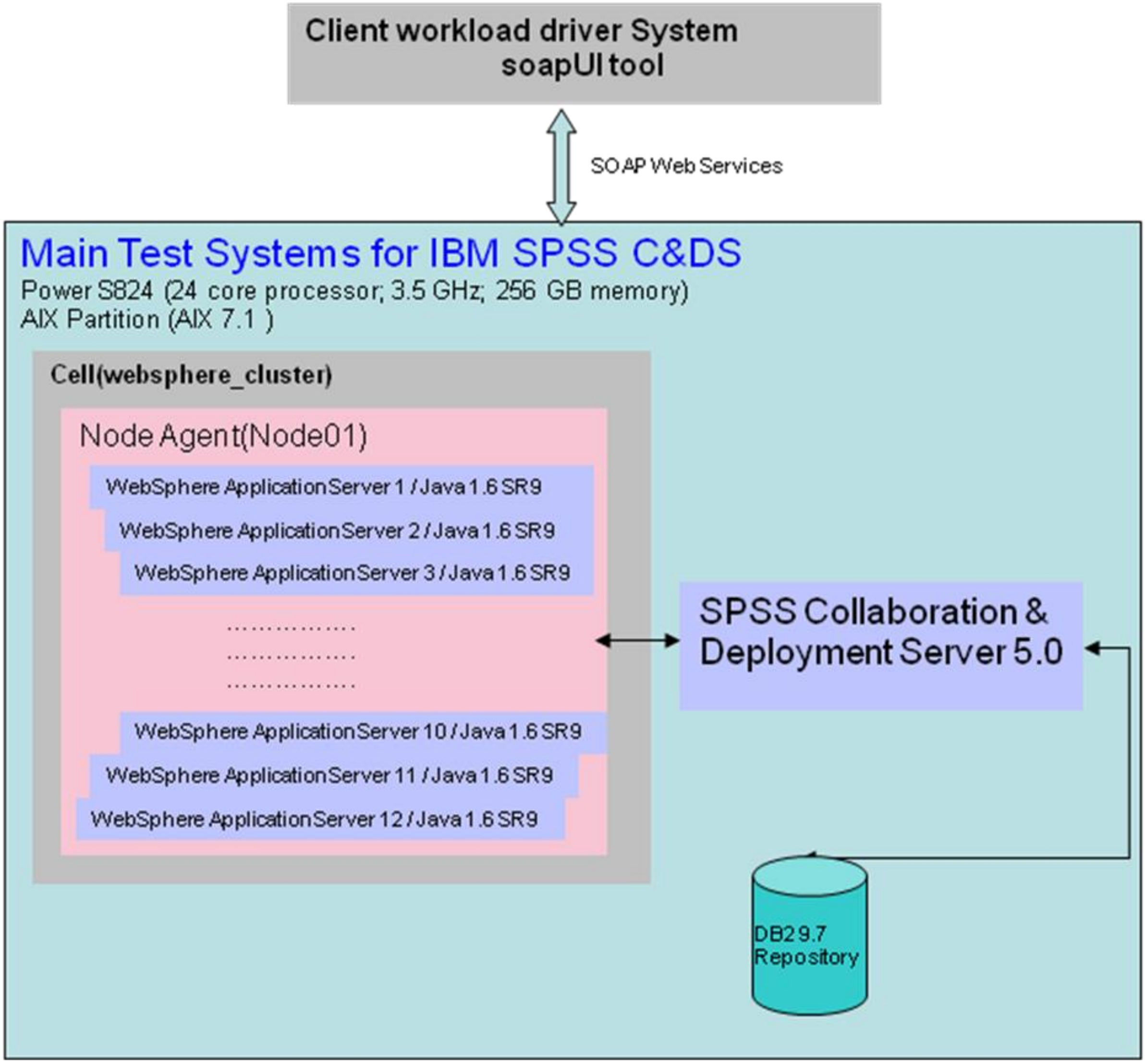

Figure 2: Clustered Collaboration and Deployment Services topology configured with multiple WebSphere instances topology on POWER8 and AIX 7.1 partitions

System under test: This system is a single Power S824 24-core server on which a single AIX 7.1 partition was created. This was the system under test on which the test team installed the WebSphere Application Server Network Deployment 8.0.0.5 with clustering capability enabled. There were a total of 12 WebSphere Application Server instances created which use Java™ 1.6. On the same AIX partition, the Collaboration and Deployment Services 5.0 application was deployed in the WebSphere Application Server cluster. IBM DB2 9.7 was used as a repository server for Collaboration and Deployment Services. There was 1 client system setup with the SoapUI tool installed on an IBM Flex System x220 compute node. The client system was used to drive the system under test at varying load points. The SoapUI client requests were load balanced against the WebSphere Application Servers on the system under test.

Test methodology

The testing methodology focused on achieving the highest scoring transaction rate on a stand-alone server with a cluster environment. Test results were measured using the average number of scores per second for the duration of the test run. The test duration of different test variants is 600 seconds. 300 seconds was used as the warm-up period and 600 seconds as the measuring period. The warm-up period allows the components of the solution to reach a level of steady-state performance that result in repeatable measurements. The test environment created an environment that replicated scoring service from a user perspective and produced an optimal level of scores per second. Multiple configurations were employed to identify the AIX environment and WebSphere settings producing the best results.

Test workload

This section explains the workload employed for performance testing and the workload tuning parameters arrived at during testing.

ProjectProfile1:

ProjectProfile1 is a simple scoring model that was used in all performance variation testing documented in this paper. The usage of a relatively lightweight model allows us to explore the maximum scalability of the infrastructure. Customer models might see different peak transaction rates, depending on the model definitions.

Scoring configuration model cache setting:

The model cache is the key performance tuning setting for Collaboration and Deployment Services scoring tests, which helps in driving workload. During the course of performance testing, different model cache settings (2, 100, 200, and 300) were applied to determine the optimal model cache, and based on scores obtained for different model cache settings, 300 model cache value performed the best among other settings. All the test results documented in this white paper were used with the 300 model cache setting.

Scoring configuration response data setting:

The performance of real-time scoring for a specific stream is influenced by the number of fields that are provided as predictor inputs and responses. The number of predictor fields used as input is determined by the stream function during model building. A best practice is to limit the response fields to the minimum. Specifically, if possible, do not echo the input fields in the response fields. This might not be necessary in the case where all the input field values are provided in the web service invocation.

WebSphere WebContainer thread pool setting:

The WebSphere Application Server web container manages all HTTP requests to web services. HTTP requests are processed by a pool of server threads. The minimum and maximum thread pool size for the web container can be configured for optimal performance. During the performance testing of Collaboration and Deployment Services, this parameter was set to 50.

SOAP client:

The SoapUI tool installed on an IBM Flex System x220 compute node was used as the workload generator tool to create a consistent workload that enabled performance to be measured. SoapUI is a very lightweight client simulator that uses a tool called TCPMon to capture the SOAP XML for the score request on ProjectProfile1. The tool simulates virtual users running a workload and captures score rates for the workload. In tuning, the test team experimented with the number of SoapUI virtual users. The final setting was 50 virtual users for each WebSphere Application Server instance.

Performance concepts in Power Systems

This section provides an overview of Power Systems and AIX performance concepts as they relate to POWER8 optimization.

The POWER8 processor chip

At the time of publishing this paper, IBM POWER8 is the latest processor in the IBM Power Systems family. The S814, S822 and S824 models have two chips in each socket, with 4, 5, or 6 cores on each chip. Along with the increased number of cores in each socket as compared with previous generations of the POWER® processors, the POWER8 processor chip implements simultaneous multithreading (SMT) supporting eight hardware threads per core, as compared to the IBM POWER5™ and IBM POWER6® that can support two hardware threads per core each and IBM POWER7® that supports four. Each POWER8 processor core supports running in a single-threaded mode with one hardware thread, an SMT2 mode with two hardware threads, SMT4 mode with four hardware threads, or an SMT8 mode with eight hardware threads. The POWER8 processor chip supplies each core with 8 MB of dedicated L3 cache. This large cache tends to significantly benefit commercial applications such as SPSS Collaboration and Deployment Services.

Logical processors

Each SMT hardware thread is represented as a logical processor in AIX. When running in SMT8 mode, the operating system will have eight logical processors for each POWER8 processor core assigned to the partition. To gain full benefit from the throughput improvement of SMT, applications need to use all of the SMT threads of the processors.

Cache affinity and processor binding

The hardware threads for each core of a POWER8 processor share core-specific hardware caches. For multithreaded applications where different threads are accessing the same data, it is usually advantageous to arrange for those threads to run on the same core. By doing so, the shared data can remain resident in the core-specific caches, as opposed to moving between different cores' caches in the system. This enhanced cache affinity can have the effect of allowing more efficient utilization of the cache space in the system and reducing the latency of data references.

Similarly, the multiple cores on a POWER8 processor chip have internal interconnects that allow them to exchange cache data between cores with relatively low latency. Again, arranging the software threads such that those which share data are run on the same POWER8 processor chip (when the partition spans processor chips) usually results in the best performance as long as the capacity requirements can be handled by a single processor chip.

Application threads can be restricted to run on a specific set of logical processors by using processor binding. This is most commonly done using the AIX `execrset` command. Processor binding is most effective when different applications (or different instances of the same application) are bound to non-overlapping sets of logical processors. The bindings need to be individual cores or sets of cores within a single chip, and should avoid binding a single application to cores on different chips.

Memory affinity

Memory affinity is a software technology that allocates physical memory that is most local to the processor where a software thread is running. IBM POWER processor-based symmetric multiprocessing (SMP) hardware systems typically are constructed with multiple microprocessor chips. In most cases, each has physical memory attached directly to each chip. Memory directly attached to a chip has the lowest latency and the highest bandwidth when accessed from processors on that chip. To enable memory affinity, the MEMORY_AFFINITY environment variable needs to be set in the application environment. Applications must be bound to processors to use the AIX memory affinity support effectively. The memory capacity configured on the server was well in excess of what was required and not a factor in the workload.

Alternatives to explicit processor binding and memory affinity

Explicit binding of processes to core or chips, when combined with memory affinity tends to result in maximum performance. However, there are conditions where explicit binding and memory affinity might produce operational side effects. An alternative that provides less performance, but more than an untuned environment, is AIX Dynamic Systems Optimization. In the course of testing, the team found SPSS Collaboration and Deployment Services worked well with Dynamic System Optimizer and resulted in significant performance improvements over a non-tuned environment. AIX Dynamic System Optimizer is dynamic with optimization based on current usage. So, it adapts very well to changing workload, or changes to the underlying partition through dynamic reconfiguration operations.

Best practices and recommended tuning to optimize Collaboration and Deployment Services

Performance tuning can be very workload-specific. Potentially, many parameters can be changed in myriad combinations. Arriving at an optimal combination may require experimentation. To maximize the likelihood of achieving excellent, if not optimal, performance results, it is always recommended to follow the best practices for an environmental setup. The following section details general best practice recommendations for running clustered Collaboration and Deployment Services on a POWER8 processor-based system with the AIX 7.1 operating system. The optimized environment includes Java heap tuning, garbage collection settings, memory tuning, processor and memory affinity settings, and client side tuning (which are explained in detail in the following section). The test results outlined in the “Results of running Collaboration and Deployment Services ” section explains the best results and recommendations in detail.

There are several AIX environment variables that influence the performance of this workload in an AIX 7.1 environment. The test team configured various AIX partition sizes. The following sections explain each of the major AIX environment settings that have been set to achieve optimal performance for Collaboration and Deployment Services.

Tuning the operating system

The following ulimit parameters were established for basic functionality of the workload for the user running the environment.

```
# ulimit -a
core file size          (blocks, -c) unlimited
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
max memory size         (kbytes, -m) unlimited
open files              (-n) unlimited
pipe size               (512 bytes, -p) 64
stack size              (kbytes, -s) hard
cpu time                (seconds, -t) unlimited
max user processes      (-u) 262144
virtual memory          (kbytes, -v) unlimited
```

Tuning Java heap and the garbage collection policy

Initial test runs showed that the default Java heap size and the default garbage collection (GC) policy was incurring high additional costs associated with garbage collection. The `-verbose:gc` option was used to collect detailed information on garbage collection. Further experiments helped identify appropriate Java heap size and garbage collection policy settings. Finally, the Java heap was tuned to an appropriate size of *2560* and garbage collection policy was set to the following value**.**

```
HEAP_SIZE       2560

JDK_OPTIONS     Xgcpolicy:gencon -Xgcthreads4 -Xcodecache32m -XtlhPrefetch -Xmn2g -
Xnoloa
```

With these settings, garbage collection overhead was reduced significantly. The team also added 32 MB code cache segments (`-Xcodecache32m`) to the Java Software Development Kit (Java SDK) options.

These settings can be done manually before starting WebSphere Application Server through the WebSphere Application Server Admin console.

Tuning memory allocation

Tests were run with different MALLOCTYPE (default, watson, and 3.1_64BIT) and MALLOCOPTIONS (multiheap, buckets, and pool) AIX environment setting combinations. The bucket allocator gave optimal performance for the workload.

Tuning resource sets and memory affinity

This section provides information about configuring multiple WebSphere Application Servers to bind to logical processors using AIX commands. In this testing on AIX 7.1, the test team had set the enhanced affinity setting to default values and used manual affinity settings.

The recommended way to bind WebSphere Application Server instances to processors is with the `execrset` command. The following example creates resource set with logical processors 0 to 7 and starts the WebSphere Application Server instance1. This example shows the binding of WebSphere Application Server to two cores of the POWER8 processor (eight logical processors in the SMT4 mode).

Example 1:

```
execrset -F -c 0-7 -m 0 -e
/wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance01
```

Optimal use of binding depends on the number of multiple WebSphere Application Server instances and the number of cores in the partition running the workload.

Example 2: The following example details binding for eight JVMs (WebSphere Application Server) configuration having two cores per WebSphere Application Server instance on a 16-core partition size with SMT4 enabled:

execrset -F -c 0-7 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance01

execrset -F -c 8-15 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance02

execrset -F -c 16-23 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance03

execrset -F -c 24-31 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance04

execrset -F -c 32-39 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance05

execrset -F -c 40-47 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance06

execrset -F -c 48-55 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance07

execrset -F -c 56-63 -m 0 -e /wasdata/was_server/profiles/AppSrv01/bin/startServer.sh spssInstance08

With these bindings, the WebSphere Application Server instance is limited to run on a defined number of logical processors. To completely make use of binding, memory affinity must be enabled. To enable memory affinity, set the MEMORY_AFFINITY environment variable to MCM.

```
export MEMORY_AFFINITY=MCM
```

When a WebSphere Application Server instance is bound to logical processors, it is recommended that the number of threads used by the Java garbage collector be explicitly set to that number of logical processors. By default, the garbage collector might normally use the number of logical processors available in the partition. Use the following JDK option to set it:

```
JDK_OPTIONS          Xgcthreads<#>
```

Where, `<#>` is the number of logical processors the WebSphere Application Server instance is bound to.

Running multiple WebSphere Application Server instances and balancing the workload across these servers is a common way to address scaling factor.

Tuning the application server and the application

The following Application Server and Collaboration and Deployment Services application tuning parameter settings were used to optimize driving workload.

- Collaboration and Deployment Services model cache was set to 300.
- WebSphere Application Server Web container threads option was set to 50.

Results of running Collaboration and Deployment Services on POWER8

This section provides the key results obtained by applying tuning configuration mentioned in configurations on the Collaboration and Deployment Services 5.0 server and WebSphere Application Server.

Benchmark results: Collaboration and Deployment Services scoring service

Testing was completed on a 24-core partition size profile with multiple WebSphere Application Server instances.

Peak performance is realized on the 24-core partition with two cores per WebSphere Application Server instance. This result is achieved with 12 instances of WebSphere Application Server JVMs and bound using the `execrset` command. Each WebSphere Application Server instance is assigned to two cores.

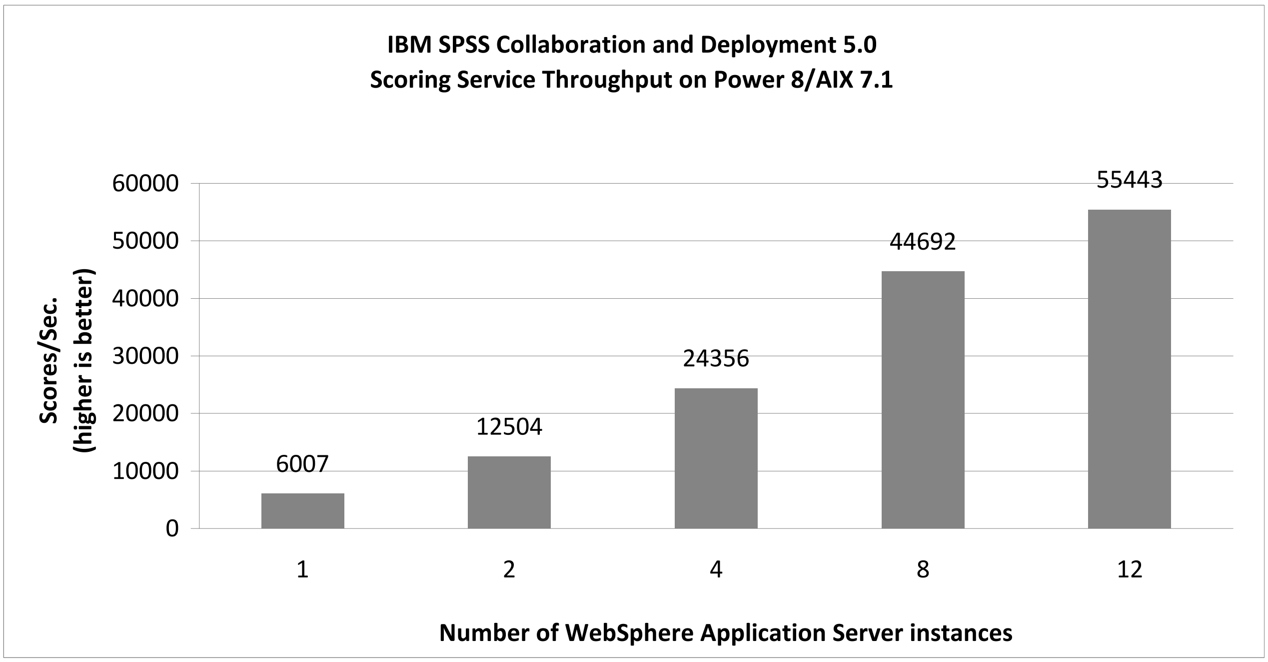

Figure 3: Collaboration and Deployment Services on multiple WebSphere Application Server instances

55443 is the maximum score per second obtained by running 12 WebSphere Application Server instances with two cores per WebSphere Application Server instance. For systems using POWER8 processors, it is recommended that two cores per WebSphere Application Server instance be used for optimal performance.

Linear scaling of scoring throughput results is also realized up through eight WebSphere Application Server instances. Processor utilization scaled similarly, for example one WAS instance 8%, two WAS instances 16%, four WAS instances 32%, and so on..

Competitive comparison

To illustrate the value of the POWER solution, the workload tuning and measurement activities were repeated on an IBM System x3650 server with two Intel Ivy Bridge processors for a total of 24 cores running at 2.7 GHz. The system was running Windows Server 2008 R2 Standard 64-bit. The measurements of the Ivy Bridge environment are summarized in Figure 4.

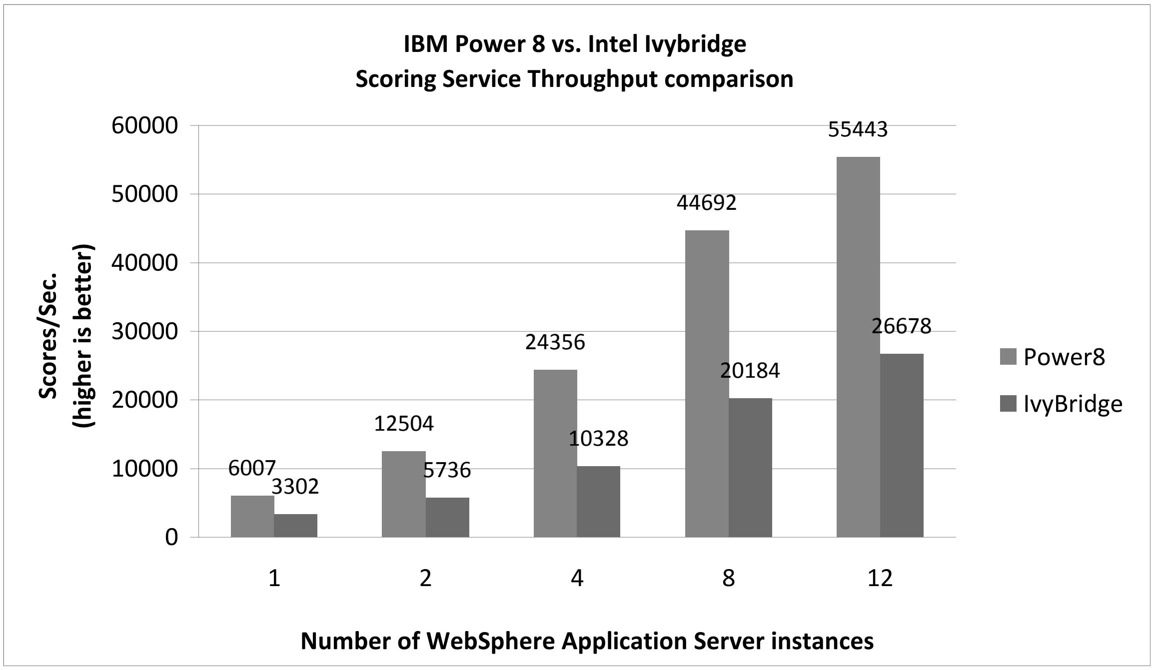

Figure 4: Comparing IBM POWER8 and AIX with Intel Ivy Bridge / Windows scoring service performance

Comparing the IBM Power S824 server with the Intel server, the test team found that the POWER processor-based server generated 1.8 to 2.3 times more scores per second when comparing measurements with the Ivy Bridge system.

Summary

The recommendations covered in this paper have demonstrated significant improvements in performance for IBM SPSS Collaboration and Deployment Services on POWER8 over untuned results. Peak results included performance of 55443 scores per second on a 24-core partition size with two cores per WebSphere Application Server instance (total of 12 instances).

Collaboration and Deployment Services is a multithreaded application that can be optimized through tuning beyond baseline settings. This paper documents the best practices employed in test environments to optimize performance settings. The key tunings recommended are Java options, memory allocator settings, and affinity (cache and memory) settings to achieve improved performance for Collaboration and Deployment Services 5.0.

In summary, this exercise of performance optimization and scalability study has shown that IBM SPSS Collaboration and Deployment Services 5.0 can be deployed to get the best performance through tuning and has provided the following key findings.

- Collaboration and Deployment Services can take advantage of memory allocation tuning, JVM settings, cache affinity (through processor binding) and memory affinity settings.
- For systems using POWER8 processors it is recommended that two cores per WebSphere Application Server instance be used for optimal performance.
- The 24-core POWER8 system delivered 1.8 to 2.3 times more scoring throughput per second than a similarly configured IBM Intel Ivy Bridge system.
- Linear scaling can be achieved by properly tuning system and balancing the workload across processor cores.

Resources

The following websites provide useful references to supplement the information contained in this paper:

- IBM SPSS Collaboration and Deployment Services 5.0 on IBM POWER7+™ and IBM AIX 7.1
  ibm.com/partnerworld/page/stg_ast_sys_wp_spss-services-on-ibm-power7-aix

- Overview of IBM SPSS Collaboration and Deployment Services
  http://www.spss.com/software/deployment/cds/

- Repository Configuration document location
  ftp://public.dhe.ibm.com/software/analytics/spss/documentation/cds/5.0/en/RepositoryConfig.pdf

- IBM Power hardware
  ibm.com/systems/power/hardware/

- IBM Power Systems Knowledge Center
  http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp

- IBM AIX 7.1 Knowledge Center
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.ntl%2FRELNOTES%2FGI11-9815-00.htm

- System memory allocation using the malloc subsystem (AIX online help)
  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.prftungd/doc/prftungd/mem_place_envir_var.htm

- Memory allocation mechanisms in AIX
  ibm.com/developerworks/aix/library/au-memoryallocation/index.html#Overview%20of%20default%20allocation%20policy

- AIX Thread support tunable parameters
  http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp?topic=/com.ibm.aix.prftungd/doc/prftungd/thread_supp_tun_params.htm

- Java Diagnostics Guide
  http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/index.jsp

About the authors

Mayank Patel is the IBM SPSS Collaboration and Deployment Services performance lead and has responsibilities that include performance testing and analysis of Collaboration and Deployment Services features, identifying and implementing performance improvements for Collaboration and Deployment Services development.

Bill Phu is a software performance analyst with the Power Systems Performance organization. He has worked on optimizing several different software products to fully exploit the IBM Power platform including but not limited to SPSS Collaboration and Deployment Services, DB2 Universal Database, and the AIX virtualization software stack. He was previously a kernel software developer for the DB2 Universal Database product.

Thomas Kochie is the development manager for IBM SPSS Collaboration and Deployment Services Business Analytics product, and has several years of experience working in the IBM Systems and Technology Group and IBM SWG focusing on AIX networking and WebSphere Application Server performance.

@IBMSystemsISVs

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at **ibm.com**/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.