Cognos 8 Business Intelligence (BI) on IBM AIX best practices

Optimizing and scaling Cognos 8 BI on IBM AIX

Authors

Dean Browne, Beth Hoffman, Mark McFadden, Pavan Pamula

IBM Corporation
October 2008

Table of contents

Abstract 1

Introduction 1

Overview of Cognos 8 BI 1

- Cognos 8 Gateway 2
- Cognos 8 Dispatcher 2
- Cognos 8 Report Server 3
- Cognos 8 Content Manager 3
- Cognos 8 data source performance 3
- Cognos 8 BI deployment practices 4

Optimizing the environment for performance 4

- Test environment 5
- Typical BI day workload 6
- Tuning AIX to optimize Cognos 8 BI performance 7
- Configuration 1 – the baseline configuration 9
- Configuration 2 – tuning memory allocation settings 9
- Configuration 3 – tuning AIX thread handling 9
- Additional (optional) configuration settings 10
- Results of memory and thread handling optimization for Cognos 8 BI 11
  - Test observations 12

Summary 14

Resources 14

Abbreviations and acronyms 15

Appendix A 15

About the authors 19

Acknowledgements 19

Trademarks and special notices 21

Abstract

This paper describes methods and settings that influence the performance of the Cognos 8 Business Intelligence (BI) software running on IBM AIX 6L Version 6.1or AIX 5L version 5.3 on IBM Power Systems (IBM POWER6) platforms. Clients and IBM consultants can use this document to learn the recommended approaches to tuning the AIX environment that Cognos 8 BI is running in. They can also learn how to create a computing environment that allows Cognos 8 BI to scale to meet the desired number of concurrent users.

Note: *Information about scaling and sizing of Cognos 8 BI is planned to be added in a future version of this paper.*

Introduction

Cognos 8 BI is the premier business intelligence (BI) software solution in the industry. Many IBM clients are deploying the application on the IBM AIX® platform and are looking for information on how to select the appropriate IBM Power™ Systems model and configuration to run it on. Installing Cognos 8 BI on AIX using default configuration settings does not enable the Cognos software to perform at optimal speeds. IBM has done a thorough analysis of how Cognos 8 BI runs on AIX on Power Systems platforms and has created a set of best practices that you should follow to ensure the Cognos software is running in an environment that maximizes transaction throughput and minimizes response times for end-users.

This paper begins with an overview of the Cognos 8 BI architecture and software tiers and refers to resources that provide guidance on doing fundamental database and Cognos tuning. The paper continues with describing the BI workload and test environment that was used in determining the optimal operating environment. Several configuration options that have proven to optimize the Cognos 8 BI performance on AIX are provided, including descriptions on how specific AIX settings affect the performance. The configuration settings focus on improving memory, thread and lock management. Although the testing done by IBM leveraged AIX 6L Version 6.1 on IBM Power Systems (IBM POWER6™ hardware), the performance improvements can potentially also be seen using AIX 5L Version 5.3 on POWER6 hardware.

Overview of Cognos 8 BI

Cognos 8 Business Intelligence is a software solution that provides enterprise-scale reporting, analysis, score carding, dashboards and event notification. The Web-based Cognos 8 architecture was designed for scalability, availability and openness. It uses platform-independent, industry-proven technology, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP) and Web Services Definition Language (WSDL). For this reason, Cognos 8 can integrate with and leverage other existing technology infrastructures on multiple platforms.

The Cognos 8 architecture features a consistent, zero-footprint, Web-based user interface for viewing, creating, and administering reports, analyses, scorecards and events. It has a common dispatcher and supports leading relational databases as well as Online Analytical Processing (OLAP) and dimensionally modeled relational cubes. It ensures dynamic load balancing and provides failover recovery for 24 x 7 operations. It also provides a single point of administration, as well as Web-based delegated administration. Cognos 8 is fully open to vendor products and custom development. It also integrates with Web farms and supports multilingual reporting and score carding.

Cognos 8 has a multi-tiered Service Oriented Architecture (SOA) with three tiers: Web, application and data. The tiers are based on business function, and are typically separated by network firewalls. The Cognos 8 end-user interfaces sit above the Web tier as shown in Figure 1.

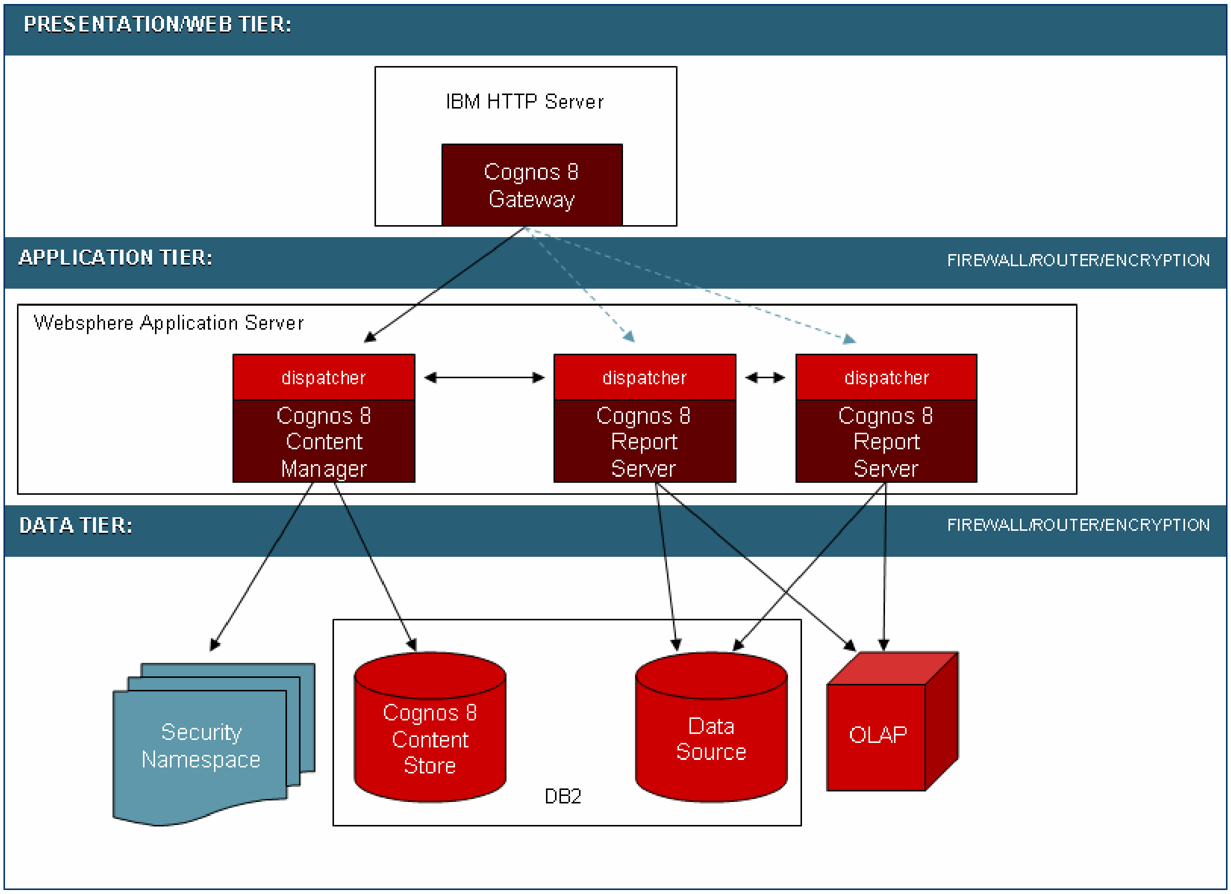


Figure 1. Cognos 8 BI software environment

Cognos 8 Gateway

All Web communication in Cognos 8 is managed by the Gateway component. Load on the Cognos 8 Gateway server is comparatively light weight but you can deploy multiple redundant Gateways along with an external HTTP load-balancing router to meet availability or scalability requirements.

Cognos 8 Dispatcher

Load balancing of requests at the application tier is performed by the Cognos 8 Dispatcher. The Dispatcher is a light-weight Java™ servlet that manages application services and provides communication between application services. At startup, each Cognos 8 Dispatcher registers locally available services with the Cognos 8 Content Manager. During the normal operation of Cognos 8 BI services, requests are load balanced across all available services using a configurable, weighted roundrobin algorithm to distribute requests. Tuning for Cognos 8 is performed by defining how Dispatcher handles requests and manages services. A normal configuration for Dispatcher is two Report Server (see next section) processes per CPU and 8 to10 threads per process. Threads within Cognos 8 are managed by the type of traffic that they handle — referred to as *high affinity* and *low affinity*. High-affinity connections are used to process absolute and high-affinity requests from the report services, whereas low-affinity connections are used to process low-affinity requests. You can manage the number of threads per Cognos 8 report service (BIBus) process through the Cognos 8 Administration Console by setting the number of high- and low-affinity connections. For more details, refer to the *Cognos 8 BI Architecture and Deployment Guide*.

Cognos 8 Report Server

The main services responsible for application tier processing are the *report service* and the *batch report service*. When these services receive requests from the Dispatcher, they start processes to handle the request load. You can specify the maximum number of processes that these services can start. The number of processes for a Report Server should be configured based on the available CPU capacity for the Report Server. In general, report processing is a CPU-bound process. The number of CPUs in a server and their clock rates are the two primary variables to consider when planning for additional Report Server hardware capacity. For example, you generally configure a server with four available CPUs to use more report service processes than a server with only two available CPUs. Similarly, given two servers with an equal number of CPUs, you should configure the server with a significantly faster CPU clock rate to use more report and report service processes.

When configuring the Cognos 8 server environment, you have to set a Java heap size. The Cognos 8 Report Service is made up of two underlying components — the Java servlet-based Dispatcher services and report services that are launched using the Java Native Interface (JNI). Set the Java virtual machine (JVM) heap size allocation for Cognos 8 so that Java memory is only as large as is necessary to accommodate the processing requirements of the Java based services. This ensures that as much memory as possible is available to the Cognos 8 BI Report Service.

Cognos 8 Content Manager

Content Manager is the Cognos 8 service that manages the storage of customer application data, including security, configuration data, models, metrics, report specifications and report output. Content Manager is needed to publish models, retrieve or store report specifications, manage scheduling information, and manage the Cognos name space. Content Manager maintains information in a relational database referred to as the *content store database*. A minimum of one Content Manager service is required for each Cognos 8 implementation. Content Manager performance benefits from the availability of high-speed RAM resources and typically requires one CPU for every four Report Server CPUs.

Cognos 8 data source performance

Query performance is typically bound to the performance of the server that is hosting the data source for Cognos 8, which can access many data sources including relational and OLAP sources. Relational data source that are tuned to meet the requirements of Cognos 8 BI reports naturally perform better than those that are not. Relational database management systems (RDBMSs) should be monitored and tuned by an experienced database administrator to achieve optimum database performance.

Cognos 8 BI deployment practices

Server components within all three tiers can easily scale to meet a variety of very large BI processing requirements. At the *Web tier*, Cognos 8 Gateways are comparably light weight and can therefore handle large user loads. You can also deploy multiple Gateway instances to meet requirements for additional user load, server availability or other service level agreements (SLAs). At the *application tier*, both the Cognos 8 Report Server and Content Manager components can scale to meet application-processing requirements. You can deploy multiple Content Manager instances in an active/standby configuration to provide high availability. Content Manager Server hardware can also scale vertically to provide increased throughput. The heavy lifting within a Cognos 8 server platform is done by the Cognos 8 Report Server. You can deploy multiple Report Servers to meet the processing requirements of very large applications and user populations. Report Server performance is tied to overall system performance and therefore can be affected by CPU clock speed and I/O performance.

The first step to creating a proper Cognos 8 BI environment begins with a successful installation. Use the *Cognos 8 BI Installation and Configuration Guide* to lead you through the installation and initial configuration process. Selecting the appropriate deployment options comes next. The section above provided you with a brief introduction to deployment suggestions for Cognos 8 BI. *For more detailed information about Cognos 8 architecture and server deployment options, read the Cognos 8 BI Architecture and Deployment Guide.* In addition, useful information about tuning the Cognos 8 server platform can be found in chapter 7 and chapter 8 of that guide.

After you have Cognos 8 BI installed and tuned, you can use the built-in tools to monitor performance. For information related to monitoring Cognos 8 system performance metrics using the Cognos 8 Administration Console, read chapter 4 of the *Cognos 8 BI Administration and Security Guide*.

You can access all of these technical reference guides at http://support.cognos.com by navigating to the Documentation page and selecting the product **Cognos 8 Business Intelligence 8.4**. You can access further information from the Knowledge Base Search tool on http://support.cognos.com.

The following section describes the tests performed and AIX configuration setting for thread management and memory management to optimize Cognos 8 performance and make the best use of the processing power of IBM POWER6 systems.

Optimizing the environment for performance

The goal is to create a BI software environment that provides optimal response time for its users and scales well to allow a growing number of concurrent users. Installing, configuring and tuning the Cognos 8 BI application is the first step for establishing a BI software environment that performs well. You learned the basics on how to tune the Cognos BI application itself in the previous section. The software environment also requires proper tuning of middleware and the operating system. Tuning the database used by Cognos 8 BI is critical for reaching optimal performance. End-user response time is especially important for those who create and view BI reports that typically access large amounts of data in the database. IBM provides a set of best practice documents for IBM DB2® for Linux®, UNIX® and Microsoft® Windows® operating systems. There are also best practices for tuning Java and IBM WebSphere® Application Server. These documents are listed in the “Reference” section.

Running Cognos 8 BI in the default AIX operating system environment on an IBM Power Systems model does not enable sufficient performance. An IBM team tested and analyzed the performance of Cognos 8 BI using multiple AIX configurations to determine the best scenario. The results show that when using specific AIX environment settings, performance response times of Cognos 8 BI drop in half. The following section provides details on the test environment that produced these results. The results you experience in your own environment can vary due to other environment differences.

Test environment

The Cognos 8 BI test environment was created by installing the following software:

- AIX 6L Version 6100-01-01-0823
- Cognos 8 BI 8.4
- IBM Java 1.5 SR7
- WebSphere Application Server v6.1.0.17
- IBM HTTP Server 6.1.0.17
- DB2 v9.5 FP1
- DB2 v9.5 Client

The test environment was set up across two IBM Power 570 systems (model 9117-MMA), each containing four POWER6 processors. The core Cognos BI Server and WebSphere Application Server were installed on one system using all four processors. The other system was divided into two logical partitions (LPARs), each receiving half the system resources for running the remaining software components and middleware products. Cognos 8 BI components were installed using the default settings with no additional application tuning other than defining two Report Servers.

4 processors:	Cognos 8 BI Application Tier, Report Server and WebSphere Application Server
2 processors (LPAR):	Cognos 8 BI Content Manager and IBM Directory Server
2 processors (LPAR):	Cognos 8 BI Gateway, IBM HTTP Server and DB2

The IBM middleware in the test environment was configured using appropriate WebSphere, Java and DB2 tuning best practices. This included using appropriate settings for the Java heap, garbage collection, Java process definition, Web-container threads, logging levels, security and the JDBC connection pool. DB2 was configured and tuned using the best practices as applied to the test data stored in the database used by the workload generator called *Typical BI day workload*.

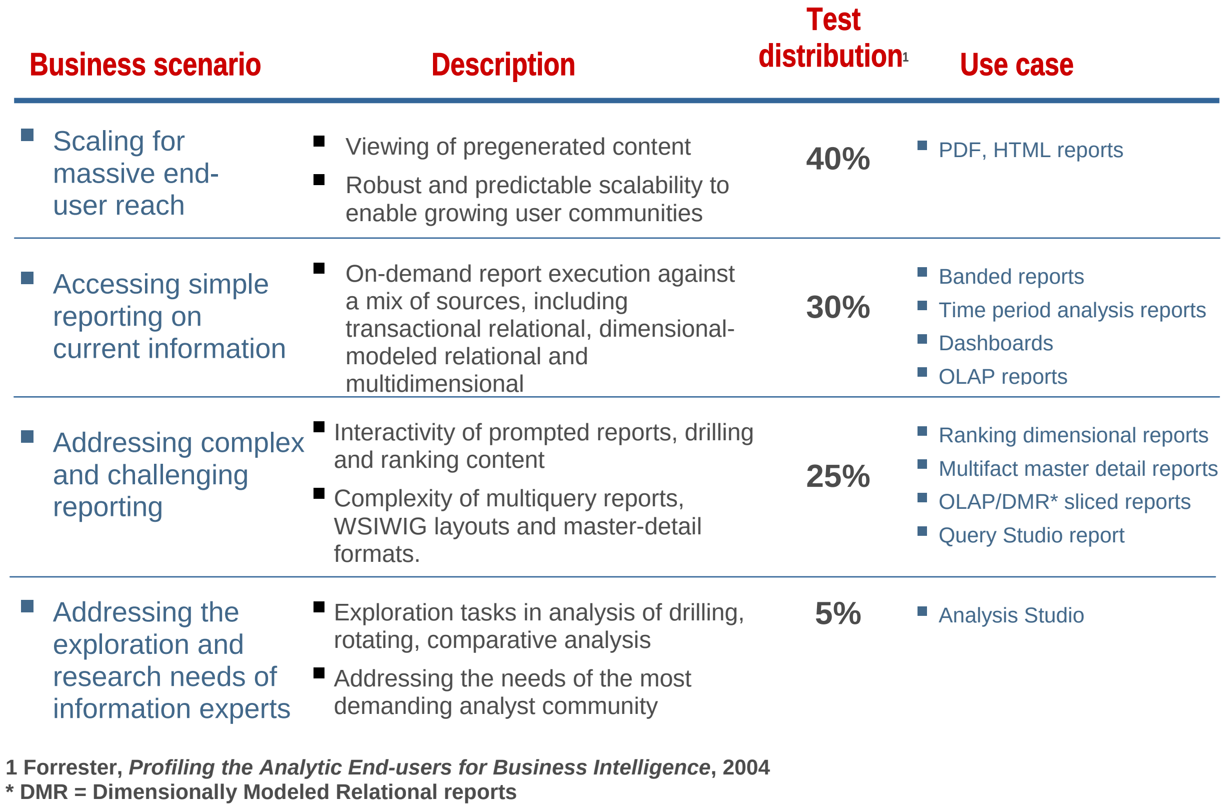
Typical BI day workload

Performance optimization is a process of configuring all application and environmental components to effectively handle application load requirements. In this case, optimization means the tuning of Cognos 8 BI, IBM AIX, server resource configuration, and the management of database objects and structures. This section demonstrates this process through performance and scalability load testing based upon BI industry-recognized utilization profile referred to as *The Typical BI Day*. This test case uses a blend of multiple BI use cases of varying size and complexity to best represent typical BI user activity as described by Forrester Research. The blended use cases are run simultaneously over an extremely large number of concurrent simulated user sessions to fully stress all application and environmental server components. When the test runs were completed, the results were analyzed and changes were made to the solution components to optimize CPU, memory and I/O performance. During testing, both the full run of blended tests and each individual test case were used to confirm the effect of optimization settings. Individual test cases were performed by running simulated load using each of the test cases from the blended Typical BI Day test case one at a time. This is not what would be typically seen in a real customer environment but the use of individual tests confirmed that configuration settings are indeed optimal for all types of user loads — rather than simply being useful for the blended test load.

The Typical BI day work load helps identify the best optimization settings. The total volume of reporting requests or transactions from simulated users are aligned with the typical company's profile as defined by Forrester where transactions are typically spread across as following percentages:

- Scale for massive end user reach: 40%
- Access simple reporting on current information: 30%
- Address complex and challenging reporting: 25 %
- Address the exploration and research needs of information experts: 5%

See Figure 2 for details describing each transaction type.



Business scenario	Description	Test distribution[1]	Use case
Scaling for massive end-user reach	- Viewing of pregenerated content - Robust and predictable scalability to enable growing user communities	40%	PDF, HTML reports
Accessing simple reporting on current information	- On-demand report execution against a mix of sources, including transactional relational, dimensional-modeled relational and multidimensional	30%	Banded reports Time period analysis reports Dashboards OLAP reports
Addressing complex and challenging reporting	- Interactivity of prompted reports, drilling and ranking content - Complexity of multiquery reports, WSIWIG layouts and master-detail formats.	25%	Ranking dimensional reports Multifact master detail reports OLAP/DMR* sliced reports Query Studio report
Addressing the exploration and research needs of information experts	- Exploration tasks in analysis of drilling, rotating, comparative analysis - Addressing the needs of the most demanding analyst community	5%	Analysis Studio

1 Forrester, *Profiling the Analytic End-users for Business Intelligence*, 2004
*** DMR = Dimensionally Modeled Relational reports**

Figure 2. Forrester's end-user profiling for BI scenarios

Tuning AIX to optimize Cognos 8 BI performance

Enabling Cognos 8 BI to perform at an optimal level on AIX requires skills and experience with AIX. The primary optimization approach requires tuning of thread, memory and lock management because most of the critical processing is implemented in a multithreaded distributed architecture. Optimized thread handling within AIX ensures that Cognos 8 Report Server (BIBus) processes can most efficiently use available POWER6 CPU resources while tuning memory allocation balances available memory between the Report Server processes and Java based Dispatcher services.

To determine the best configuration for Cognos 8 BI on AIX, the optimization testing ran through a series of configurations. After each configuration, load testing was repeated and performance metrics were analyzed. The following code shows the AIX user limits (ulimits) required for Cognos 8 to run optimally on AIX. The ulimits need to be set for the user account that is being used to run Cognos 8 BI. These resource limit settings are required so that Cognos 8 processes can run with maximum available resources on the POWER6 system. Set the stack to 4194304 Kbytes to permit Cognos 8 to use the maximum addressable memory for a 32-bit process. Set all other parameters to `unlimited`.

```
# ulimit –a
time(seconds) unlimited
file(blocks) unlimited
data(Kbytes) unlimited
stack(Kbytes) 4194304
memory(Kbytes) unlimited
coredump(blocks) unlimited
nofiles(descriptors) unlimited
```

There are several key AIX environment variables that influence performance. You can set the following environment variables to influence the way that AIX manages a highly threaded Java environment. A brief description of each is provided:

```
MALLOCMULTIHEAP=1 (AIX 5.2 and earlier)

MALLOCOPTIONS=multiheap (AIX 5.3)
```

This environment variable enables the use of multiple memory heaps. In a highly threaded environment, many threads can request memory at the same time. With only one heap, these requests can serialize and create wait situations. With this variable, multiple heaps are generated that are used to serve memory-allocations requests and reduce memory-allocation latency.

```
MALLOCTYPE=buckets
```

This environment variable using the `buckets` settings is an extension to the default memory allocator. It is designed to improve performance of applications which issue many small allocation requests. Buckets are predefined portions of memory divided into smaller parts. If a memory allocation request can be served by a bucket, the allocation can be performed more quickly than the default allocation which is used if the request cannot be served by a bucket.

```
AIXTHREAD_SCOPE=S (will be set automatically by the IBM JVM)
```

This environment variable sets the thread scope to system wide and creates a one-to-one mapping between user threads and kernel threads. The default value affects all processes and creates one kernel thread per process.

The *Typical BI day workload* was used to simulate loads varying from 40 and 200 concurrent users of the Cognos 8 BI application. A series of 14 different configurations were tested, each enabling a different environment variable to allow analysis of individual settings. The results of each test were compared to the same *out-of-the-box* configuration, which means installing the operating system and Cognos software using default settings. Three configurations reached a similar optimal performance level. The results of those best performing configurations used the AIX environment variables shown as follows:

```
NO AIX_THREADSCOPE

MALLOCTYPE=buckets
(on all Cognos 8 Servers)

MALLOCMULTIHEAP=heaps:n
(where n=2 times number of CPU cores) (on all Cognos 8 Servers)

AIXTHREAD_MINKTHREADS=32
```

```
AIXTHREAD_MNRATIO=1:1

AIXTHREAD_MUTEX_FAST=ON
```

Configuration 1 – the baseline configuration

The baseline configuration is used as a point of reference when comparing the impact of changing AIX thread handling and memory allocation settings on Cognos 8 BI performance. Configurations are default out–of-the-box settings for both AIX and Cognos 8 BI.

Cognos 8 Dispatchers are configured with two BIBus processes and 10 threads per CPU core.

No changes are made to AIX settings other than user limits and AIXTHREAD_SCOPE=S, which is set by the IBM JVM and is therefore considered to be a default configuration.

Configuration 2 – tuning memory allocation settings

The first configuration change tuned memory parameters MALLOCTYPE and MALLOCMULTIHEAP to optimize memory handling performance for Cognos 8 BI.

Cognos 8 BI server processes are multithreaded and are resource intensive under load; issuing a large number of small allocation requests to AIX. To accommodate this, MALLOCTYPE=buckets was set for the DB2 server, IBM HTTP Server, LDAP server and all Cognos 8 application-tier servers. The configuration of MALLOCTYPE=buckets provides an optional buckets-based extension of the default memory allocator. Setting MALLOCTYPE to “buckets” improves performance for Cognos 8 BI because this instructs the memory allocator to accommodate threads issuing large numbers of small allocation requests, matching the requirements of Cognos 8 processes. When malloc buckets is enabled, allocation requests that fall within a predefined range of block sizes are processed by malloc buckets. All other requests are processed in the usual manner by the default memory allocator.

MALLOCMULTIHEAP=heaps:*n* (where *n*=2 times the number of CPU’) were set for all Cognos 8 servers.

Multiple heaps are required so that Cognos 8 BI processes can issue malloc(), free(), and realloc() subroutine calls from more than one thread. With a single heap, all threads trying issue malloc(), free() or realloc() calls are serialized (that is, only one call at a time). The result is a serious performance impact on multiprocessor systems. With multiple heaps, each thread gets its own heap. If all heaps are being used, then any new threads trying to issue a call has to wait until one or more of the heaps are available. Serialization still exists, but the likelihood of its occurrence and its impact when it does occur is greatly reduced.

Configuration 3 – tuning AIX thread handling

During the next set of tests, memory allocation settings that were optimized in Configuration 2 are retained and focus is shifted to optimizing CPU utilization for Cognos 8 BI by changing the way that AIX manages threads.

Cognos 8 BI server processes perform a lot of data processing activity in the CPU kernel mode. Using the AIX environment variable setting AIXTHREAD_MINKTHREADS=32 -, the number of kernel threads available to Cognos 8 processes is raised to 32 from the default of 8 threads. This provides the Cognos 8 server processes with more kernel threads and hence improves the efficiency of Cognos 8 CPU utilization. This is especially true in multiprocessor systems such as POWER6. In addition to increasing the number of kernel threads, setting the ratio of user threads to kernel threads also improves Cognos 8 performance. Using the AIX environment variable setting of AIXTHREAD_MNRATIO=1:1 provides a 1-to-1 matching of user threads to kernel threads, which is the recommended setting for applications using a large number of threads such as Cognos 8.

Given the large number of threads created by Cognos 8, thread lock contention can cause requests to queue up, thereby degrading the efficiency CPU utilizing under default AIX configurations. Use of the AIX environmental variable setting AIXTHREAD_MUTEX_FAST=ON enables the use of the optimized mutex locking mechanism and results in improved Cognos 8 performance due to more efficient CPU utilization.

Note: The combination of the optimized memory-allocation settings and optimized thread-handling settings using AIX environment variables improved the performance of the Cognos 8 test cases by 60%.

Additional (optional) configuration settings

Cognos 8 BI applications and POWER6 system deployments can differ by each individual implementation. In the tests, the settings already described provided the most optimal performance for Cognos 8 BI. However, there are additional environment variables that can provide additional performance benefit for other implementations of Cognos 8 BI on AIX. The following environment variable settings are optional and you should test them before deployed them for a production Cognos 8 server. None of these settings resulted in substantial performance improvements or degradation for the test cases.

```
AIXTHREAD_COND_DEBUG=OFF
```

Because Cognos 8 BI contains a large number of active condition variables and frequently creates and destroys condition variables, this sometimes causes higher overhead for maintenance of the list of condition variables. Setting the variable to OFF disables the list and can sometimes improve performance.

```
AIXTHREAD_RWLOCK_DEBUG=OFF
```

Because Cognos 8 BI issues a large number of active read-write locks and frequently creates and destroys read-write locks, this might create higher overhead for maintaining the list of read-write locks. Setting the variable AIXTHREAD_RWLOCK_DEBUG to OFF disables the list.

```
SPINLOOPTIME=500
```

Increasing SPINLOOPTIME from its default setting of 40 permits Cognos 8 processes to wait longer for threads in the case of thread contention. Cognos 8 processes that are under unusually high load might experience a lot of pthread mutex contention. Allowing the Cognos 8 process to wait longer for thread contention to free up results in fewer failed requests and generally improved throughput. Setting

SPINLOOPTIME=500 did not result in notable performance improvements for the test cases for Cognos 8.

```
MALLOCOPTIONS=buckets, multiheap:32,considersize
```

Using these memory allocation options fine tunes the way that heaps are allocated for Cognos 8 processes. Use of the *multiheap:32* and *considersize* MALLOCOPTIONS settings did not result in notable performance improvements for the test cases for Cognos 8.

Results of memory and thread handling optimization for Cognos 8 BI

The following charts show the result of properly tuning AIX thread handling and memory allocation for a Cognos 8 server installation on a POWER6 platform. Based on the test cases, Cognos 8 BI performance on AIX can be improved by 60% simply by tuning AIX environmental variables. Tuning first AIX memory allocation and then AIX thread handling showed an average user response time of 6.83 and 6.81 seconds, respectively, compared to the out-of-the-box baseline configuration with an average response time of more than 16 seconds. Overall this means that with the settings used with any of the three configurations, end-users saw a reduction in response time by nearly 10 seconds. Performance improved by nearly 60% for all 160 concurrent users (see Figure 3, Figure 4 and Figure 5).

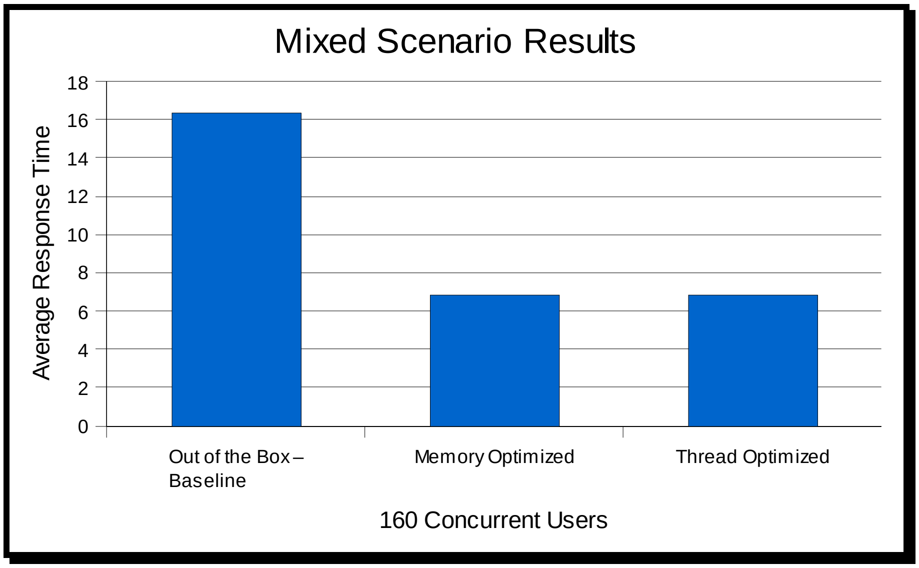


Figure 3. Mixed-scenario test results

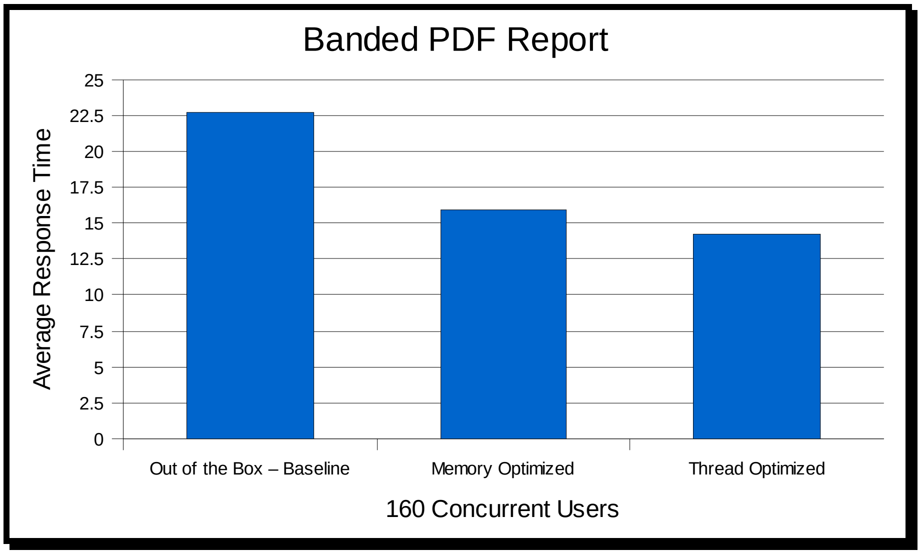


Figure 4. Example test results for a single-use case

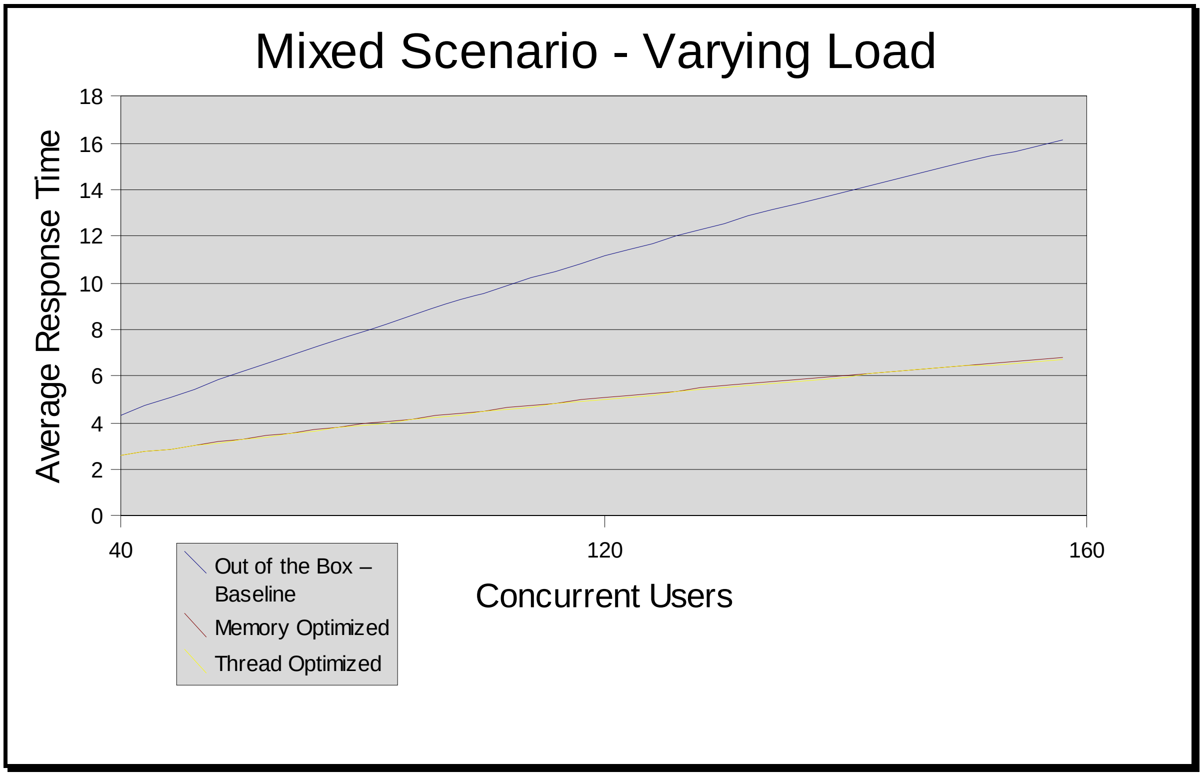


Figure 5. Mixed scenario results with increasing load

Test observations

The following observations were made for the scenarios:

- Note the significant performance improvement from base installation and configuration when AIX 6.1 is properly tuned for Cognos 8

- Proper tuning of AIX for Cognos 8 BI should include optimized settings for both memory allocation and thread handling.
- The various heap and thread environment variable changes other than AIXTHREAD_SCOPE=S or the combination of AIXTHREAD_MINKTHREADS=32, AIX_THREAD_MNRATION=1:1 & AIXTHREAD_MUTEX_FAST=ON had little to no effect on viewing the saved content functionality that is performed by Java based code.
- The combination of the thread settings above and MALLOCTYPE=buckets & MALLOCMULTIHEAP=heaps:n (where n=2 times Num of CPUs) set for all Cognos 8 servers had the most positive performance benefit for functionality such as running reports and ad hoc query analysis that is largely C++ code based.
- Various parts of profiling data (CPU - tprof, Memory – svmon, and Java - gcmv) showed the efficient use of resources when the tests are run within optimized configurations.
- Optimization of memory allocation and thread handling resulted in increased system resource utilization due to more efficient resource use and therefore greater overall throughput for Cognos 8.
- The heap setting changes had the most positive affect on report execution and Analysis Studio operations.

The performance test scenarios were run on AIX 6.1 on an IBM Power Systems (POWER6) platform and the conclusions were derived from that environment. Even though the performance environment was AIX 6.1, the recommended configurations can also improve performance of the application running on AIX 5.3 on IBM Power Systems (POWER6) models Results in any given test environment can vary, but the general performance conclusions and specific settings should help improve response times with either AIX version. The syntax for AIX 5.3 and 6.1 environment variables are very similar.

Appendix A describes the syntax of AIX thread and memory tuning parameters that were used during the tests. These were the environment variables that had the largest impact on allowing performance to be at its optimal.

The AIX environment variables are also available online at AIX and System p and AIX Information Center at:

http://publib.boulder.ibm.com/infocenter/systems/scope/aix/topic/com.ibm.aix.prftungd/doc/prftungd/thread_supp_tun_params.htm?tocNode=int_231649

Summary

Performance of the Cognos 8 BI software running on AIX is influenced by the type of workload, how the software components and middleware products are configured, how the operating system is tuned, and the hardware capacity of the environment. Cognos 8 BI is a multitiered, multithreaded, resource-intensive distributed solution that requires tuning beyond its default settings. This paper has provided specific recommendations for tuning the AIX operating system that Cognos 8 BI runs on. Three key tuning areas that are important to improve the performance of Cognos 8 BI on AIX are *thread management*, *memory management* and *lock management*.

The paper outlined various configurations with memory and thread parameters and explained how each configuration improved the Cognos 8 BI performance on AIX. There was an observed 60% increase in performance with some of the best optimized configurations compared to the out-of-the-box deployment. These best practices serve as an excellent starting point for optimizing the Cognos 8 BI product on AIX 6.1 or 5.3 environments. You can use them to help optimize your Cognos 8 BI to meet performance goals of your business.

It is also important to understand the type of workload and deployment environment because changing one parameter might help the performance of a certain transaction but might be slowing down another. Tuning actions always come with a tradeoff, the art of performance tuning is to move these tradeoffs to a part of the system where they do not have an impact on the business-critical process chain.

Resources

These Web sites provide useful references to supplement the information contained in this paper:

- Cognos BI Information Center
  http://support.cognos.com

- IBM System p and AIX Information Center
  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp

- DB2 LUW Best Practices
  http://www.ibm.com/developerworks/db2/bestpractices/

- WebSphere 6.1 Information Center
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp

- WebSphere IBM Redbooks®
  http://www.redbooks.ibm.com/

  http://www.redbooks.ibm.com/abstracts/sg247304.html?Open

- IBM Java Diagnostic Guide
  http://www.ibm.com/developerworks/java/jdk/diagnosis/

Abbreviations and acronyms

BI	business intelligence
BIBus	Cognos 8 SOA based API and purpose built BI services
DMR	dimensionally modeled relational
JNI	Java Native Interface
JVM	Java virtual machine
OLAP	online analytical processing
RDBMS	relational database management system
SMP	symmetric multiprocessing
SOAP	Simple Object Access Protocol
WSDL	Web Services Definition Language
XML	Extensible Markup Language

Appendix A

The following tables provide the AIX environment parameters that were applied in the optimization scenarios.

1. **AIXTHREAD_MINKTHREADS (AIX 5.3 and AIX 6.1)**

Purpose Controls the minimum number of kernel threads that should be used.

Values: Default: 8. Range: A positive integer value.

Display: **echo $AIXTHREAD_MINKTHREADS**

This is turned on internally, so the initial default value will not be seen with the **echo** command.

Change: **AIXTHREAD_MINKTHREADS**=*n***export AIXTHREAD_MINKTHREADS**

Change takes effect immediately in this shell. Change is effective until logging out of this shell. Permanent change is made by adding the **AIXTHREAD_MINKTHREADS** =*n* command to the **/etc/environment** file.

Tuning: The library scheduler will not reclaim kernel threads below the value set in this variable. A kernel thread might be reclaimed at virtually any point. Generally, a kernel thread is targeted as a result of a pthread terminating.

2. AIXTHREAD_MNRATIO (AIX 5.3 and AIX 6.1)

Purpose: Controls the scaling factor of the library. This ratio is used when creating and terminating pthreads.

Values: Default: 8:1 Range: Two positive values (*p:k*), where *k* is the number of kernel threads that should be employed to handle the number of executable pthreads defined in the *p* variable.

Display: **echo $AIXTHREAD_MNRATIO**

This is turned on internally, so the initial default value will not be seen with the **echo** command.

Change: **AIXTHREAD_MNRATIO**=*p:k***export AIXTHREAD_MNRATIO**

Change takes effect immediately in this shell. Change is effective until logging out of this shell. Permanent change is made by adding the **AIXTHREAD_MNRATIO**=*p:k* command to the **/etc/environment** file.

Tuning: Might be useful for applications with a very large number of threads. However, always test a ratio of 1:1 because it might provide better performance.

3. AIXTHREAD_SCOPE (AIX 5.3 and 6.1)

Purpose: Controls contention scope. A value of P signifies process-based contention scope (M:N). A value of S signifies system-based contention scope (1:1).

Values: Default: S. Possible Values: P or S.

Display: **echo $AIXTHREAD_SCOPE**

This is turned on internally, so the initial default value will not be seen with the **echo** command.

Change: **AIXTHREAD_SCOPE**=*{P|S}***export AIXTHREAD_SCOPE**

Change takes effect immediately in this shell. Change is effective until logging out of this shell. Permanent change is made by adding the **AIXTHREAD_SCOPE**=*{P|S}* command to the **/etc/environment** file.

Diagnosis: If fewer threads are being dispatched than expected, try system scope.

Tuning: Tests on AIX 4.3.2 have shown that certain applications can perform much better with system-based contention scope (S). The use of this environment variable has an impact only on those threads created with the default attribute. The default attribute is employed when the *attr* parameter to **pthread_create** is NULL.

4. AIXTHREAD_MUTEX_FAST (AIX 5.3 and 6.1)

Purpose: Enables the use of the optimized mutex locking mechanism.

Values: Default: OFF. Possible values: ON, OFF.

Display: **echo $AIXTHREAD_MUTEX_FAST**

This is turned on internally, so the initial default value will not be seen with the **echo** command.

Change: **AIXTHREAD_MUTEX_FAST**=*{ON|OFF}***export AIXTHREAD_MUTEX_FAST**

This change takes effect immediately and is effective until you log out of this shell. Permanent change is made by adding the **AIXTHREAD_MUTEX_FAST**=*{ON|OFF}* command to the **/etc/environment** file.

Diagnosis: Setting the variable to ON forces threaded applications to use an optimized mutex locking mechanism, resulting in increased performance.

Tuning: If the program experiences performance degradation due to heavy mutex contention, then setting this variable to ON will force the pthread library to use an optimized mutex locking mechanism that works only on process private mutexes. These process private mutexes must be initialized using the pthread_mutex_init routine and must be destroyed using the pthread_mutex_destroy routine.

About the authors

Dean Browne, Product Manager – Cognos Platform: Architecture, Advanced Deployment, Performance and Scalability

Beth Hoffman: Solution Relationship Manager for SWG and IBM Certified Consulting IT Specialist. Beth works in IBM System & Technology division in the ISV Global Business Strategy & Enablement organization. She leads joint technical projects involving IBM software on IBM platforms.

Mark McFadden is the manager of IBM/Cognos BI Performance Team. He graduated from Carleton University with a Bachelor in Engineering. During his over 8 years career as a Performance Test Engineer, Mark has been Testing and Project prime on numerous large scalability projects.

Pavan Pamula is currently a technical lead for the IBM Java software development kit (SDK) enablement team. He leads IBM Systems family and independent software vendor (ISV) initiatives with IBM Java SDK development and support teams.

Acknowledgements

Thank you to the following key contributors and reviewers to the paper.

Nigel Campbell – Cognos Architect

Eric Racine – Software Engineer

Srinava Navaratnam – Technical Lead

Mark Pilon – Software Engineer

Tom Bentley – Technical Lead

Bret Olszewski – AIX Performance Lead

Martin McGinn – Project Manager

Godfrey Lee – Associate Vice President

Quinn Li – Cognos Optimization Lead

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 www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.