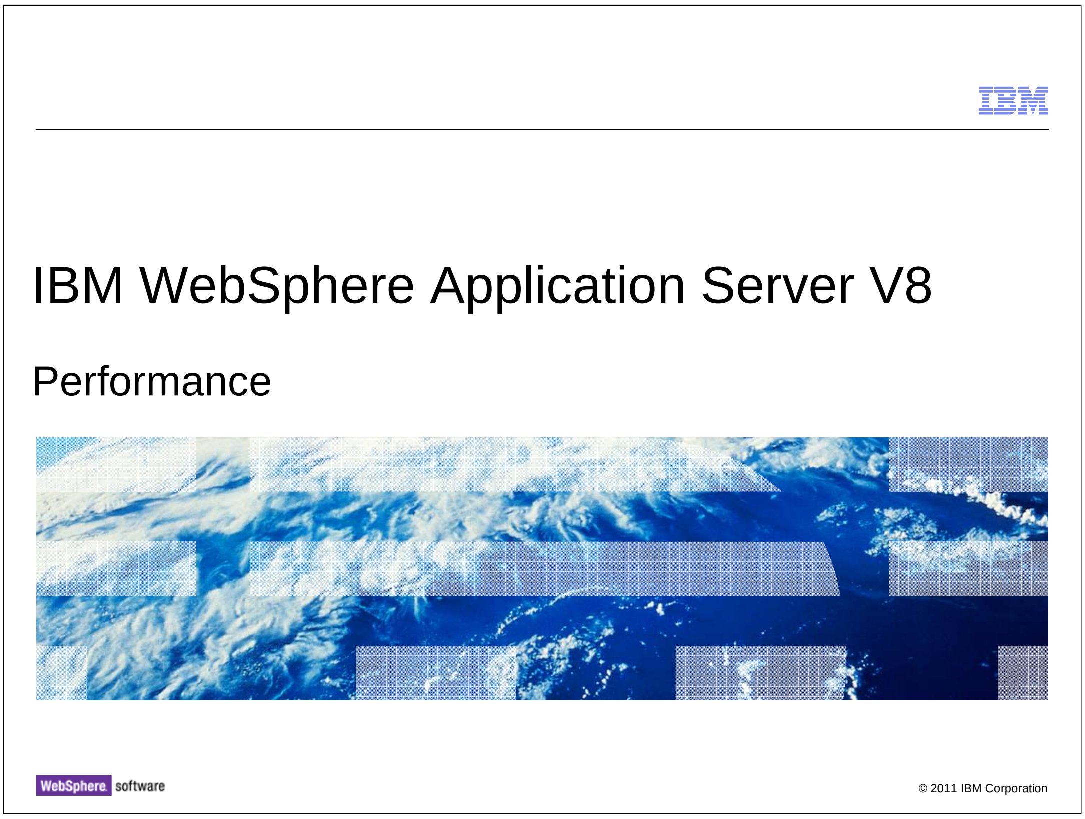
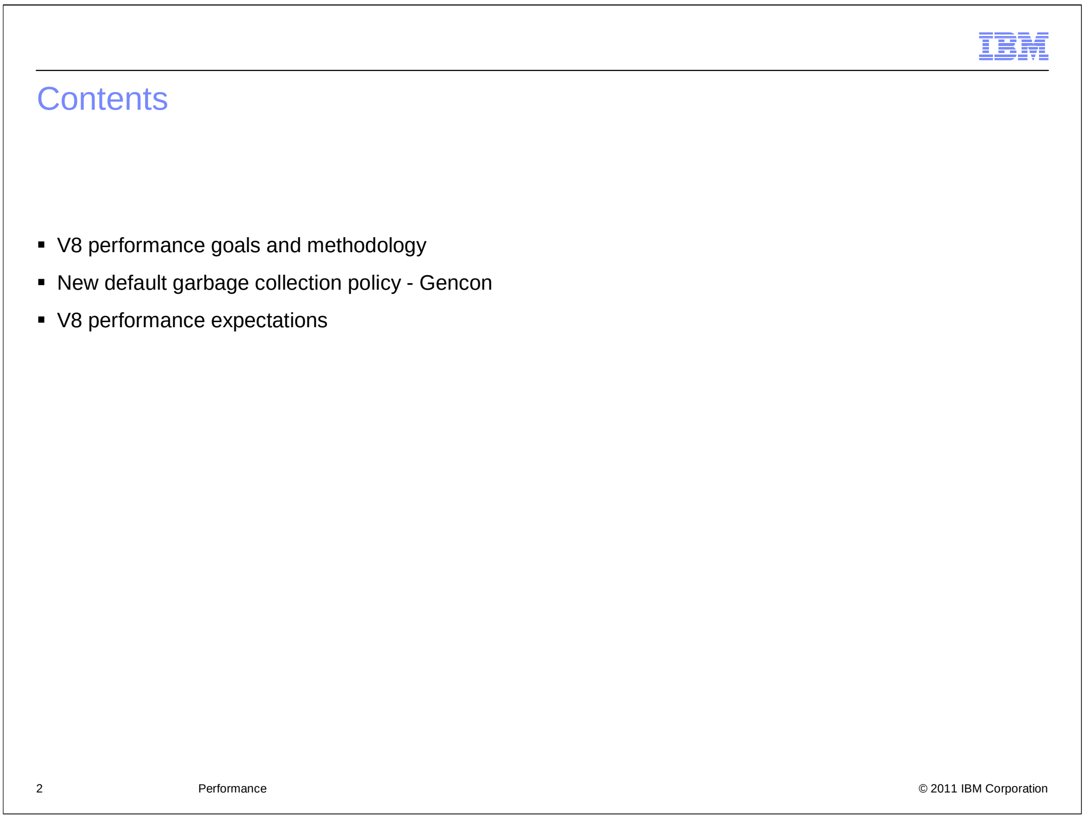


IBM WebSphere Application Server V8

Performance

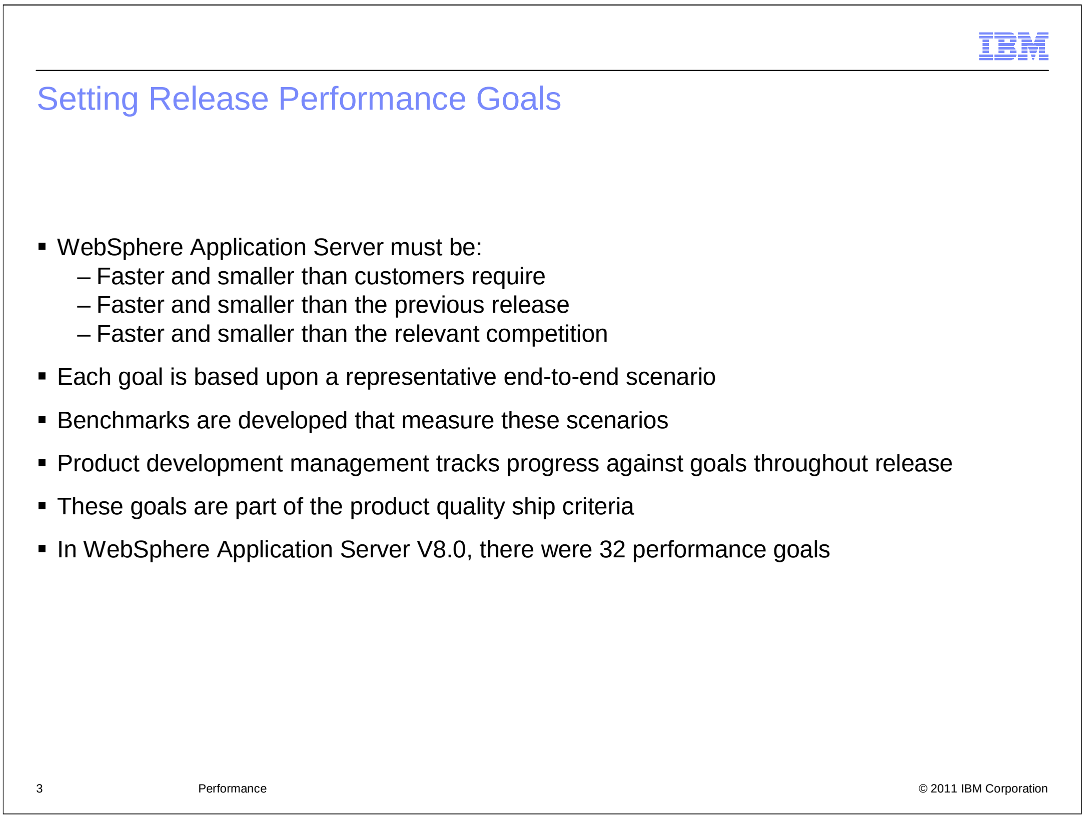
This presentation covers what is new in the area of performance in WebSphere Application Server V8, and quantifies some of the performance improvements compared to version 7.



Contents

- V8 performance goals and methodology
- New default garbage collection policy - Gencon
- V8 performance expectations

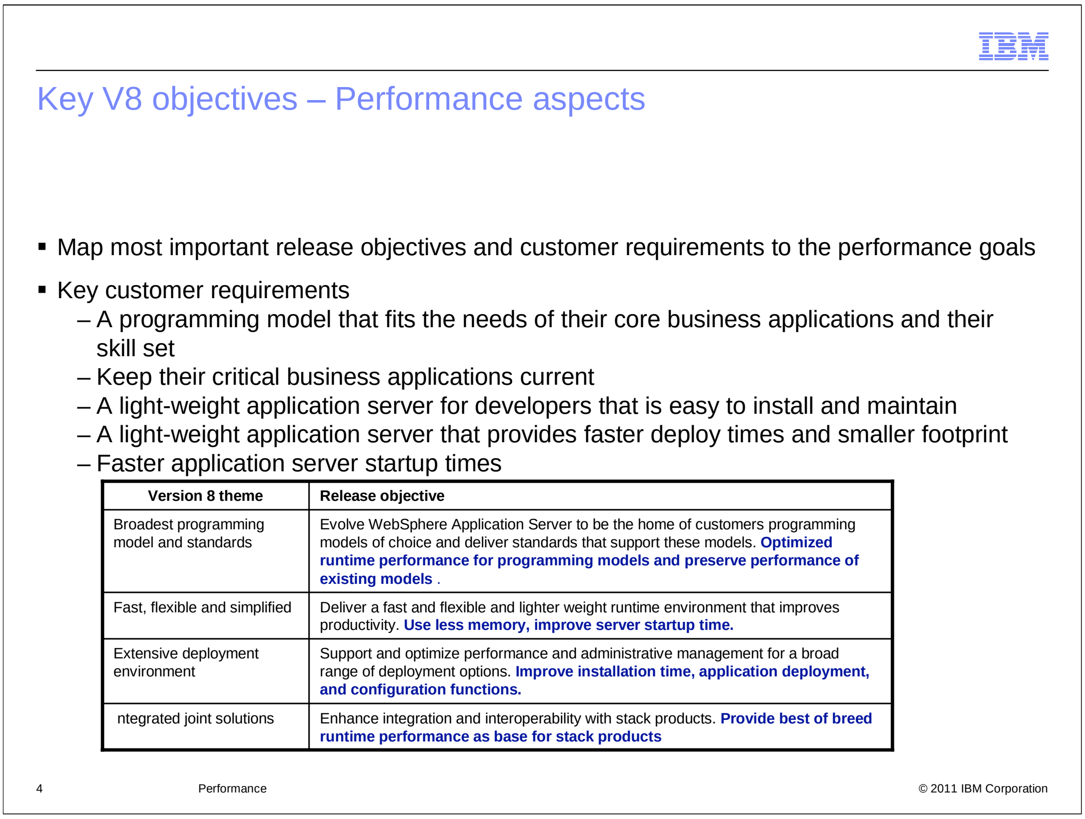
The first three slides in this presentation describes the V8 performance test methodology, how performance goals were set and benchmarks that were measured. Then, there is a single slide that describes the new default Gencon GC policy in V8. Although this policy is not new in version 8, it is an important change that should be highlighted. The final five slides talk about the areas where performance has improved in V8.



Setting Release Performance Goals

- WebSphere Application Server must be:
  - Faster and smaller than customers require
  - Faster and smaller than the previous release
  - Faster and smaller than the relevant competition
- Each goal is based upon a representative end-to-end scenario
- Benchmarks are developed that measure these scenarios
- Product development management tracks progress against goals throughout release
- These goals are part of the product quality ship criteria
- In WebSphere Application Server V8.0, there were 32 performance goals

Associated with each release of WebSphere Application Server is a set of performance goals. The goals are set very early in the planning stages of the release based on the functional content of the release. Several goals are common across almost all releases. For example there are always goals in place to measure the core runtime performance. Other goals are set based on the specific themes of the release. For example one of the themes for version 8 is to be fast, flexible and simplified, so one of the goals was to improve server startup time in a developer environment. In each case, the goal is based on customer representative scenarios, and benchmarks are developed based on these scenarios. These goals are part of the quality ship criteria and are tracked closely by the release management team throughout the product’s development cycle. In version 8 there were a total of 32 performance goals.



Key V8 objectives – Performance aspects

- Map most important release objectives and customer requirements to the performance goals
- Key customer requirements
  - A programming model that fits the needs of their core business applications and their skill set
  - Keep their critical business applications current
  - A light-weight application server for developers that is easy to install and maintain
  - A light-weight application server that provides faster deploy times and smaller footprint
  - Faster application server startup times

Version 8 theme	Release objective
Broadest programming model and standards	Evolve WebSphere Application Server to be the home of customers programming models of choice and deliver standards that support these models. **Optimized runtime performance for programming models and preserve performance of existing models** .
Fast, flexible and simplified	Deliver a fast and flexible and lighter weight runtime environment that improves productivity. **Use less memory, improve server startup time.**
Extensive deployment environment	Support and optimize performance and administrative management for a broad range of deployment options. **Improve installation time, application deployment, and configuration functions.**
ntegrated joint solutions	Enhance integration and interoperability with stack products. **Provide best of breed runtime performance as base for stack products**

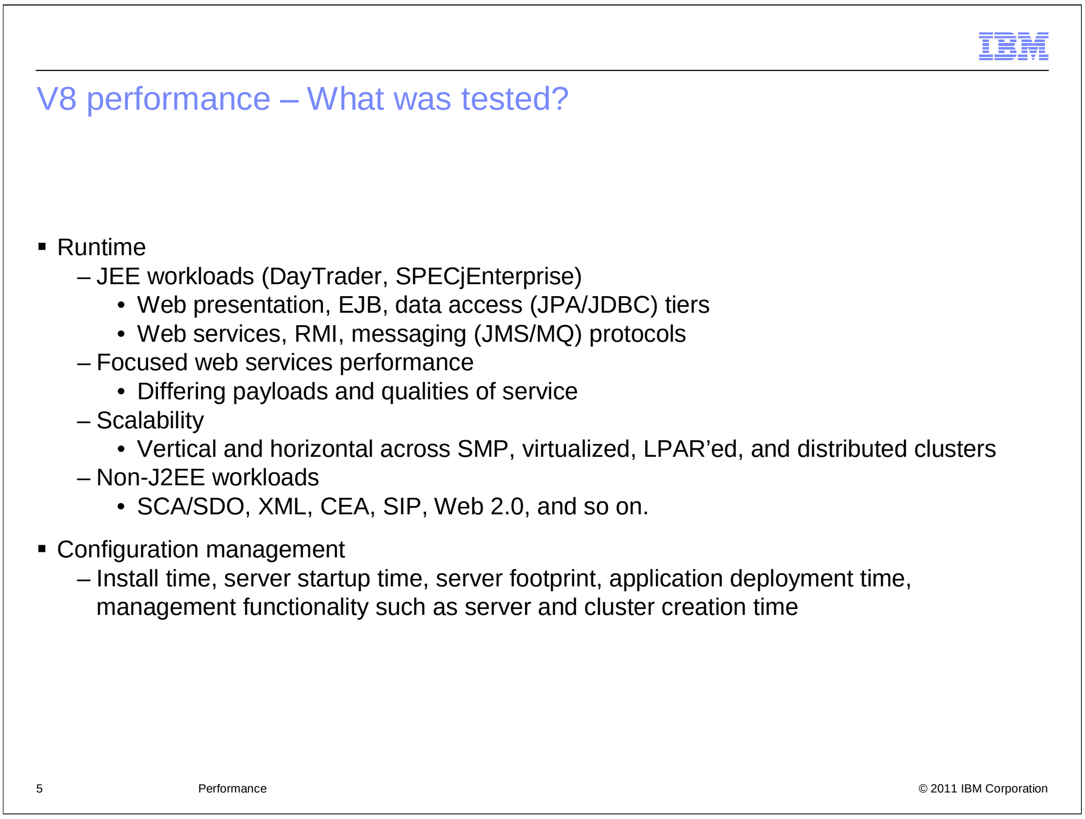
This slide shows the four major themes for version 8 along with a set of key customer requirements that are the bases for these themes. For each release performance goals are set to reflect these themes and requirements.

For the “Broadest Programming Models and Standards,” goals were set that continued the focus on core runtime performance that preserves the performance of existing application models and new models such as the new OSGI application model.

For “Fast, Flexible, and Simplified,” server startup time and memory footprint were measured in a standard Java EE runtime environment and in a much smaller and simpler developer environment.

For “Extensive Deployment Environment” goals were set for product installation time, application deploy time, and configuration functions such as creating servers and clusters.

The “Integrated Joint Studies” goals are supported by the continued focus on the core runtime, which is the base for all of the stack products.



IBM

V8 performance – What was tested?

- Runtime
  - JEE workloads (DayTrader, SPECjEnterprise)
    - Web presentation, EJB, data access (JPA/JDBC) tiers
    - Web services, RMI, messaging (JMS/MQ) protocols
  - Focused web services performance
    - Differing payloads and qualities of service
  - Scalability
    - Vertical and horizontal across SMP, virtualized, LPAR’ed, and distributed clusters
  - Non-J2EE workloads
    - SCA/SDO, XML, CEA, SIP, Web 2.0, and so on.
- Configuration management
  - Install time, server startup time, server footprint, application deployment time, management functionality such as server and cluster creation time

5 Performance © 2011 IBM Corporation

In version 8 the performance effort can be divided into two categories, core runtime performance and configuration management.

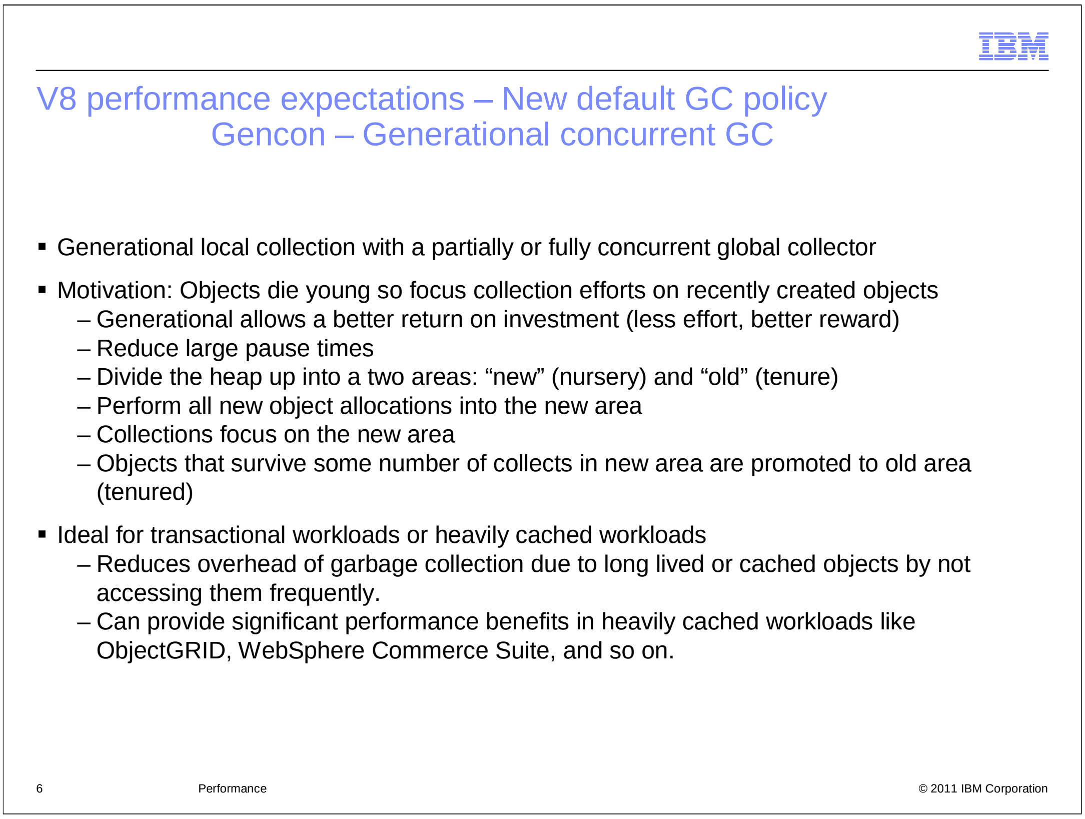
The two key benchmarks that stress the key J2EE functionality are DayTrader and SPECjEnterprise2010. DayTrader is an IBM developed workload that has since been donated to the Apache Open Source. SPECjEnterprise2010 was developed by the Standard Performance Evaluation Corporation. Both benchmarks exploit key functions of the Java EE such as servlets, EJBs, JPA, web services, RMI, and JMS/MQ

A set of web services primitives were used that allow for the specification of different input and output payload sizes. Payload sizes of 3k, 10k and 100k were measured. There are also primitives that measure SOA qualities of service such as the Security Assertion Markup Language (SAML), Message Transmission Optimization Mechanism (MTOM), web services addressing, reliable messaging, and secure conversation.

Both vertical and horizontal scaling were measured. The main focus in V8 was vertical scaling up to a 16-way.

In version 8, SIP was measured in both a stand-alone and clustered configuration. The other functions listed under this bullet were not specific V8 performance goals, however they were monitored for performance.

Because one of the main themes was to be faster and simpler several configuration management goals were set, such as installation, server startup and footprint, application deploy time, server and cluster creation time.

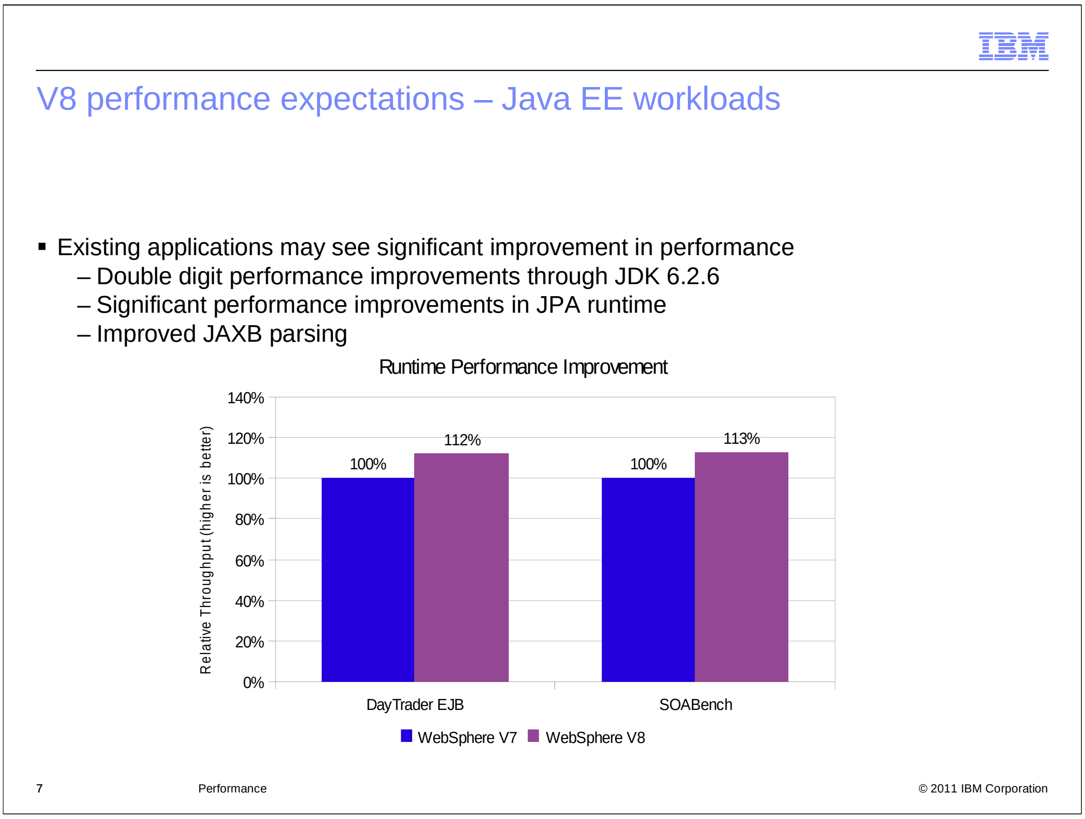


V8 performance expectations – New default GC policy Gencon – Generational concurrent GC

- Generational local collection with a partially or fully concurrent global collector
- Motivation: Objects die young so focus collection efforts on recently created objects
  - Generational allows a better return on investment (less effort, better reward)
  - Reduce large pause times
  - Divide the heap up into a two areas: “new” (nursery) and “old” (tenure)
  - Perform all new object allocations into the new area
  - Collections focus on the new area
  - Objects that survive some number of collects in new area are promoted to old area (tenured)
- Ideal for transactional workloads or heavily cached workloads
  - Reduces overhead of garbage collection due to long lived or cached objects by not accessing them frequently.
  - Can provide significant performance benefits in heavily cached workloads like ObjectGRID, WebSphere Commerce Suite, and so on.

6 Performance © 2011 IBM Corporation

One of the key changes in version 8 that should be highlighted is the change from the default garbage collection policy of optthruput in version 7 to Gencon in version 8. With optthruput all objects are allocated from one large contiguous heap that is shared by all threads. When a garbage collection is required the entire heap is scanned and unreferenced objects are “collected”. Generally this option is best for applications that require optimal throughput (hence the name) but one of the drawbacks is longer GC pause times. Gencon reduces these longer pause times by dividing the heap into two sections, the nursery and tenured areas. New objects are always allocated in the nursery. When objects have aged (been scanned multiple times by GC) they are moved to the tenured area. The JVM constantly monitors the size of both the nursery and tenured areas and adjusts the size of each based on GC frequency and pause times. This philosophy is based on the observation that most objects are short term, their life cycle is short so by allocating them in separate heaps, scanning becomes more efficient since none of the longer lived objects are scanned. This type of GC policy is ideal for transactional workloads or heavily cached workloads as it reduces the overhead of GC by referencing these long-lived objects infrequently.



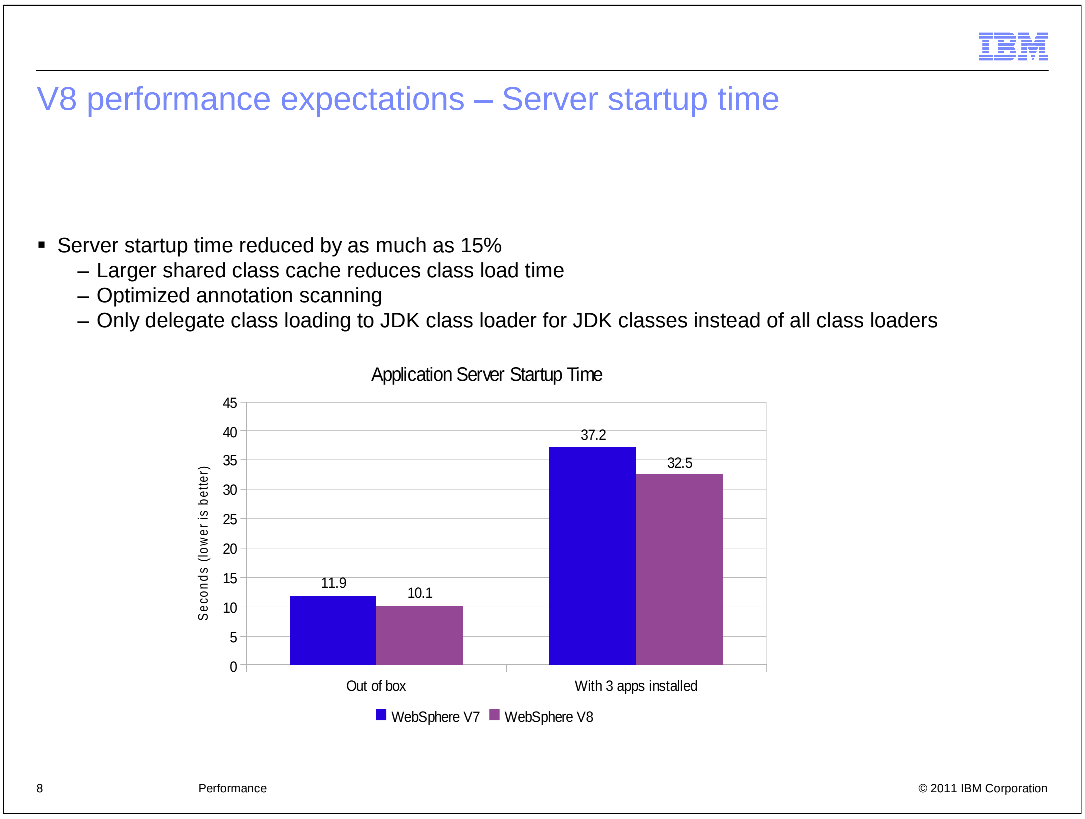
V8 performance expectations – Java EE workloads

- Existing applications may see significant improvement in performance
  - Double digit performance improvements through JDK 6.2.6
  - Significant performance improvements in JPA runtime
  - Improved JAXB parsing

7 Performance © 2011 IBM Corporation

Now, some of the specific performance results for WebSphere Application Server Version 8 are presented.

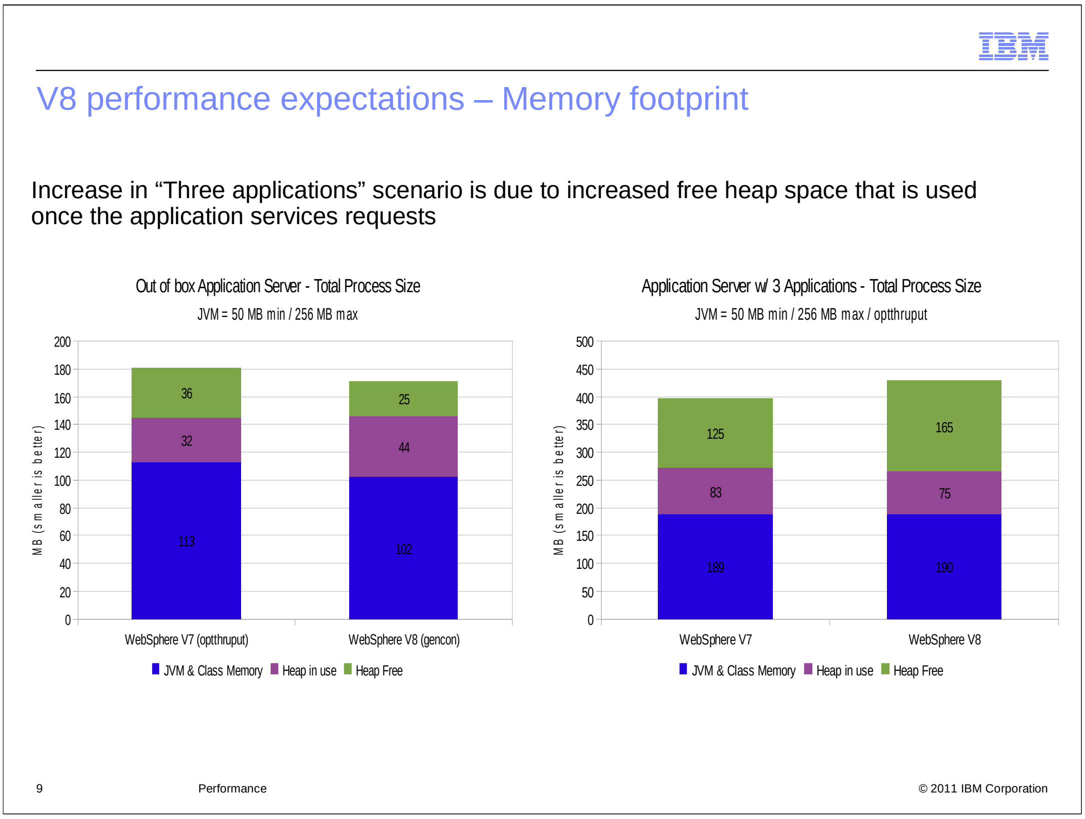
First is core runtime performance. This chart shows the performance of the DayTrader EJB workload and the SOABench workload. DayTrader is one of IBM’s key J2EE workloads and SOABench is the workload for measuring web services. In V8, DayTrader performance has improved by about 12% and SOABench by about 13% as compared to version 7. These improvements come from enhancements to three areas: Java 6 JIT and JVM improvements; optimizations to open JPA code, query cache, and improved lock handling; and improved parser performance for JAXB parsing.



V8 performance expectations – Server startup time

- Server startup time reduced by as much as 15%
  - Larger shared class cache reduces class load time
  - Optimized annotation scanning
  - Only delegate class loading to JDK class loader for JDK classes instead of all class loaders

This slide shows the improvement of server startup time in version 8. The out-of-box scenario measures a base application server configured as it is delivered in the V8 product and shows an improvement of about 15% over version 7. With three applications installed the improvement increases to about 13%. These improvements are due to an increase in the shared class cache, optimized annotation scanning and improved class loading.

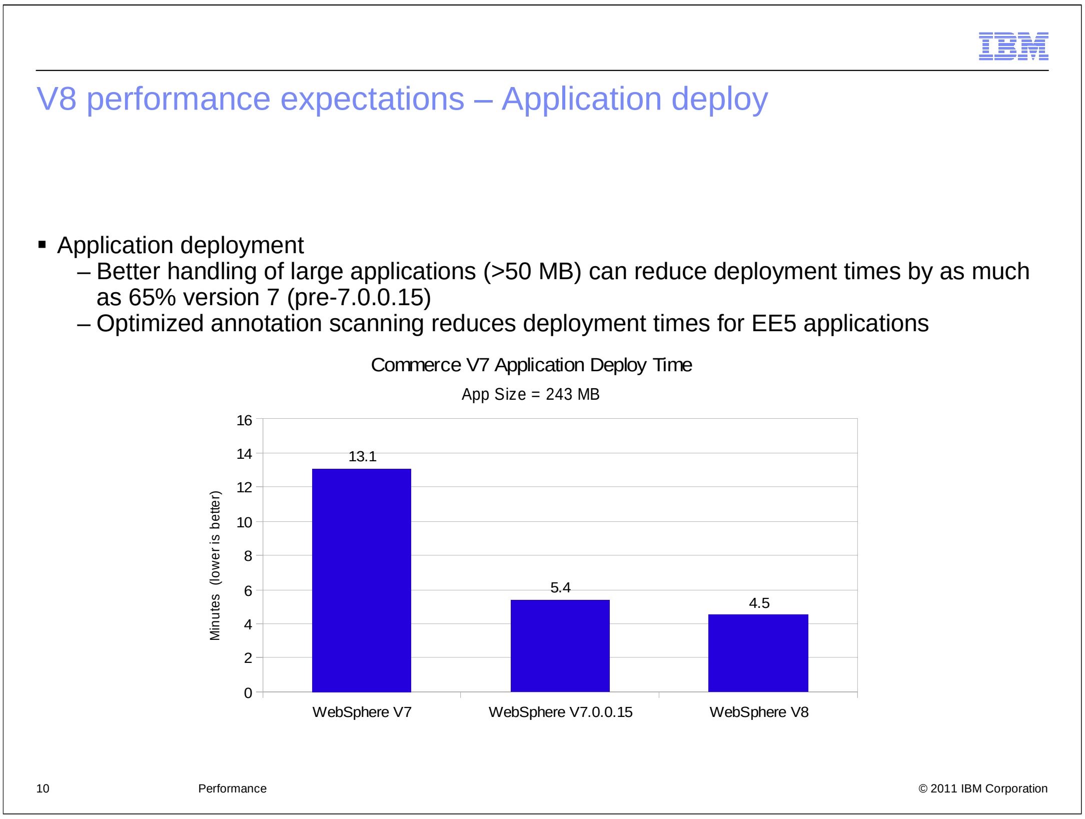


V8 performance expectations – Memory footprint

Increase in “Three applications” scenario is due to increased free heap space that is used once the application services requests

9 Performance © 2011 IBM Corporation

This slide quantifies the memory footprint in the same two server configurations as server startup. It divides the memory footprint into three major components, JVM and class memory, heap in-use, and heap free. The chart on the left shows the memory footprint comparing version 7 with the default optthruput GC policy to version 8 with the default Gencon GC policy. It shows a total of about 5% decrease in memory use coming from a decrease in both JVM and class memory and heap free space. The chart on the right shows an increase in memory footprint of about 8%. This is due almost entirely by the increase in heap free space. However this space is used by the application when the server begins processing requests.

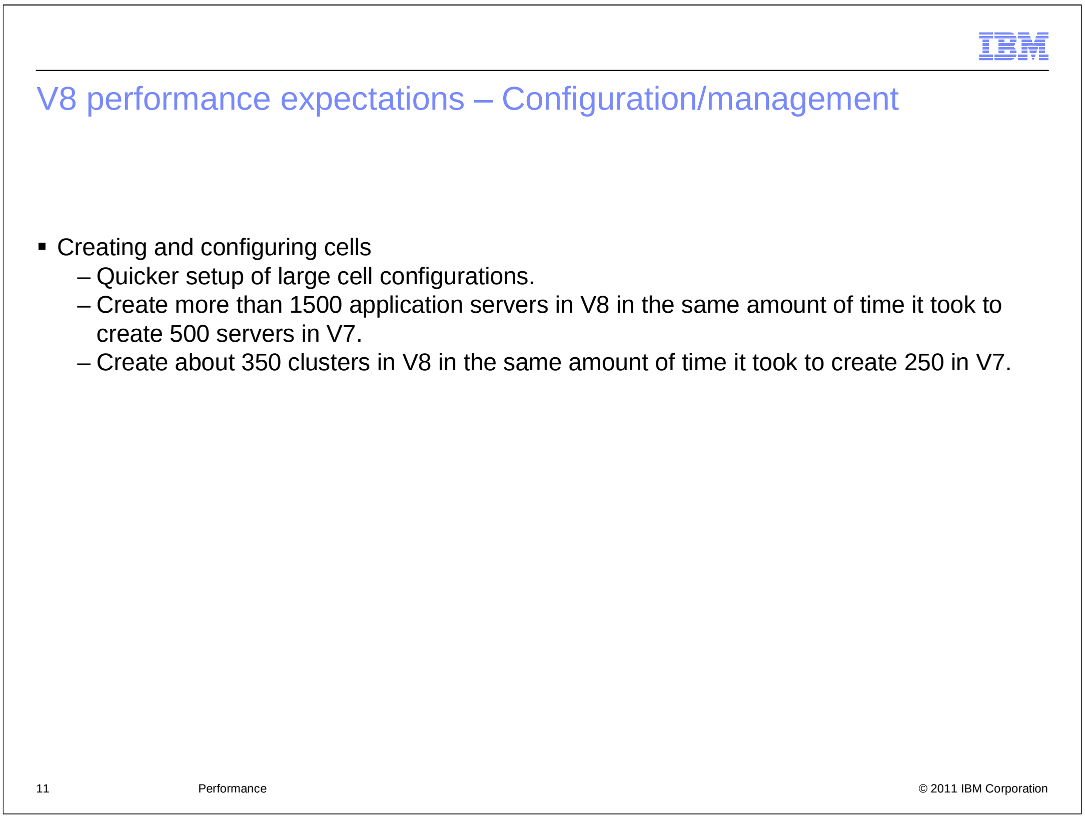


V8 performance expectations – Application deploy

- Application deployment
  - Better handling of large applications (>50 MB) can reduce deployment times by as much as 65% version 7 (pre-7.0.0.15)
  - Optimized annotation scanning reduces deployment times for EE5 applications

10 Performance © 2011 IBM Corporation

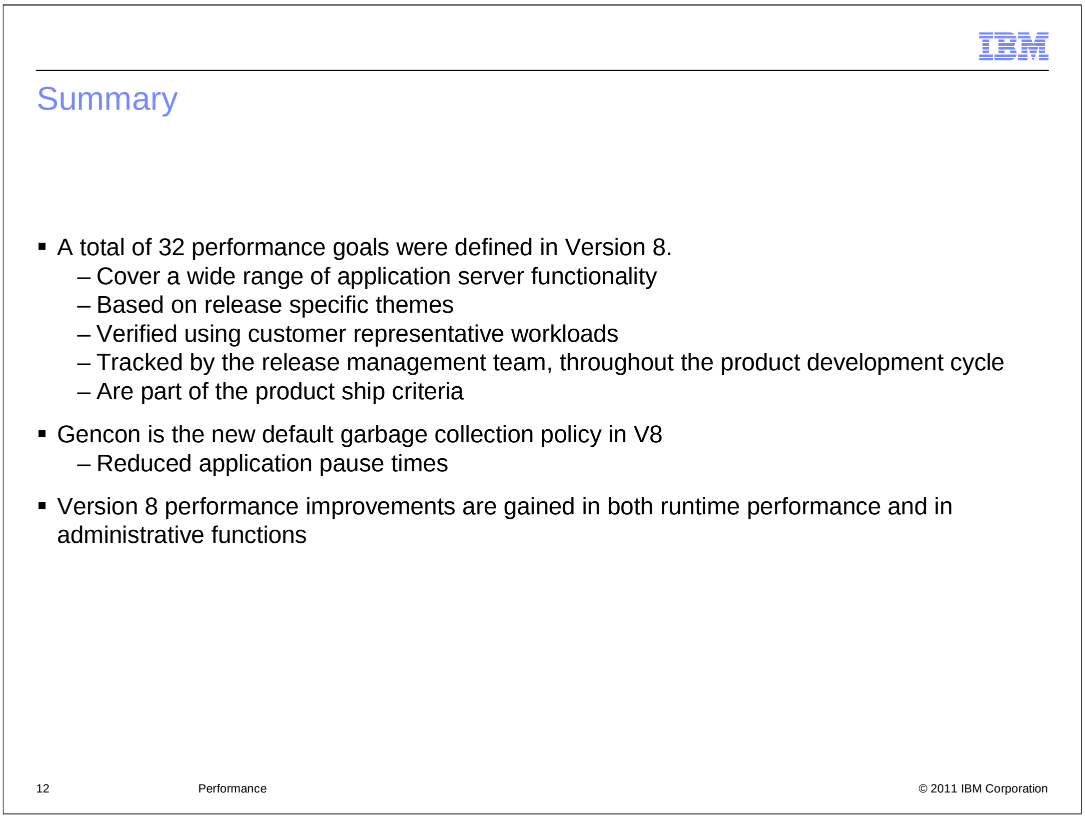
Recently huge improvements have been made in the area of application deploy. This slide shows the improvements that were shipped in 7.0.0.15 and additional improvements in V8 when deploying large Java EE 5 applications such as IBM WebSphere Commerce Version 7. Improvements in the area of annotation scanning has improved deployment times by 59% in version 7.0.0.15 and another 17% in version 8.



V8 performance expectations – Configuration/management

- Creating and configuring cells
  - Quicker setup of large cell configurations.
  - Create more than 1500 application servers in V8 in the same amount of time it took to create 500 servers in V7.
  - Create about 350 clusters in V8 in the same amount of time it took to create 250 in V7.

In V8, some significant performance improvements were made for configurations with a large number of servers and clusters. In V8 over 1500 servers can be created in the same amount of time it takes to create 500 servers in V7. This represents over a 3x improvement. Cluster creation has also improved by about 40%. 350 clusters can be created in the same amount of time as 250 in V7.



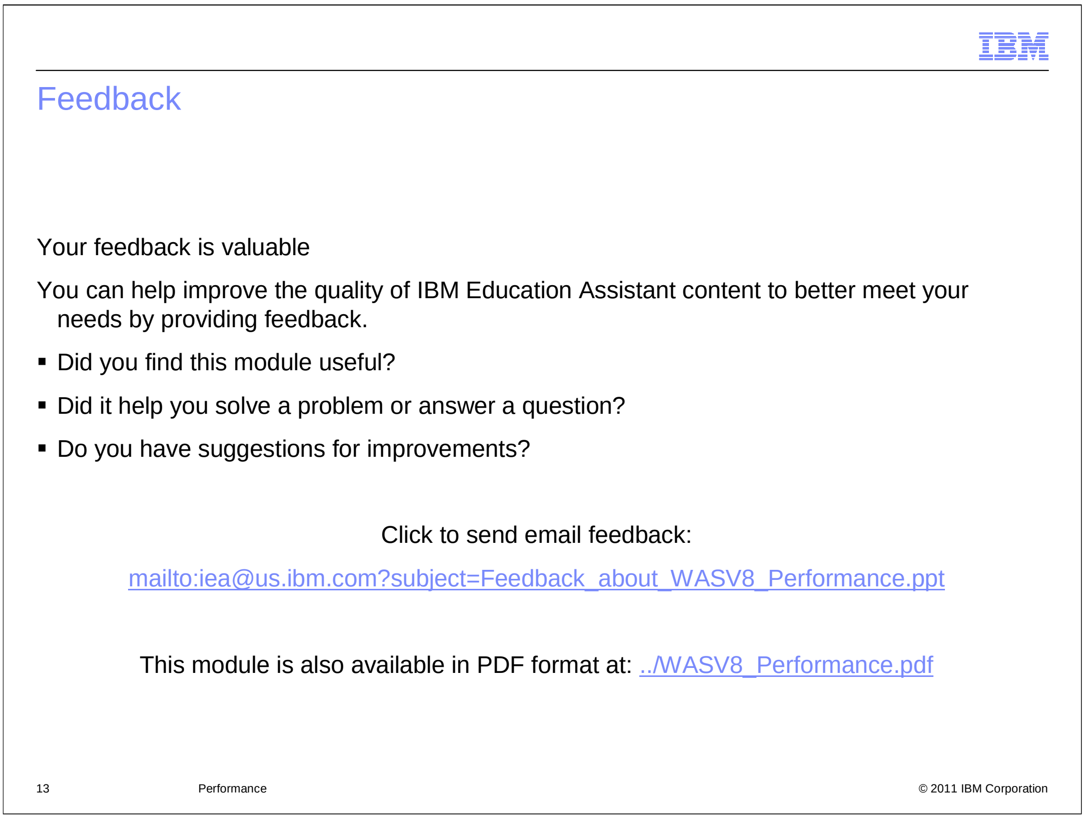
Summary

- A total of 32 performance goals were defined in Version 8.
  - Cover a wide range of application server functionality
  - Based on release specific themes
  - Verified using customer representative workloads
  - Tracked by the release management team, throughout the product development cycle
  - Are part of the product ship criteria
- Gencon is the new default garbage collection policy in V8
  - Reduced application pause times
- Version 8 performance improvements are gained in both runtime performance and in administrative functions

There were a total of 32 performance goals set in V8 that covered a wide range of application server functionality. Goals were chosen based on specific themes (content) for the release. Each goal is verified using a customer representative workload. Goals are tracked by the release management team throughout the development cycle and are part of the release ship criteria.

An important change in V8 that is related to performance is the change in the default garbage collection policy. In V7 the default was opt_thruput, in V8 it is Gencon. The main advantage to using Gencon is the more frequent but shorter garbage collection intervals which leads to shorter pause times for application requests.

V8 performance has improved as compared to V7 in both the mainline application runtime performance, startup and footprint, and administrative functions such as application deploy time and server/cluster creation times.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

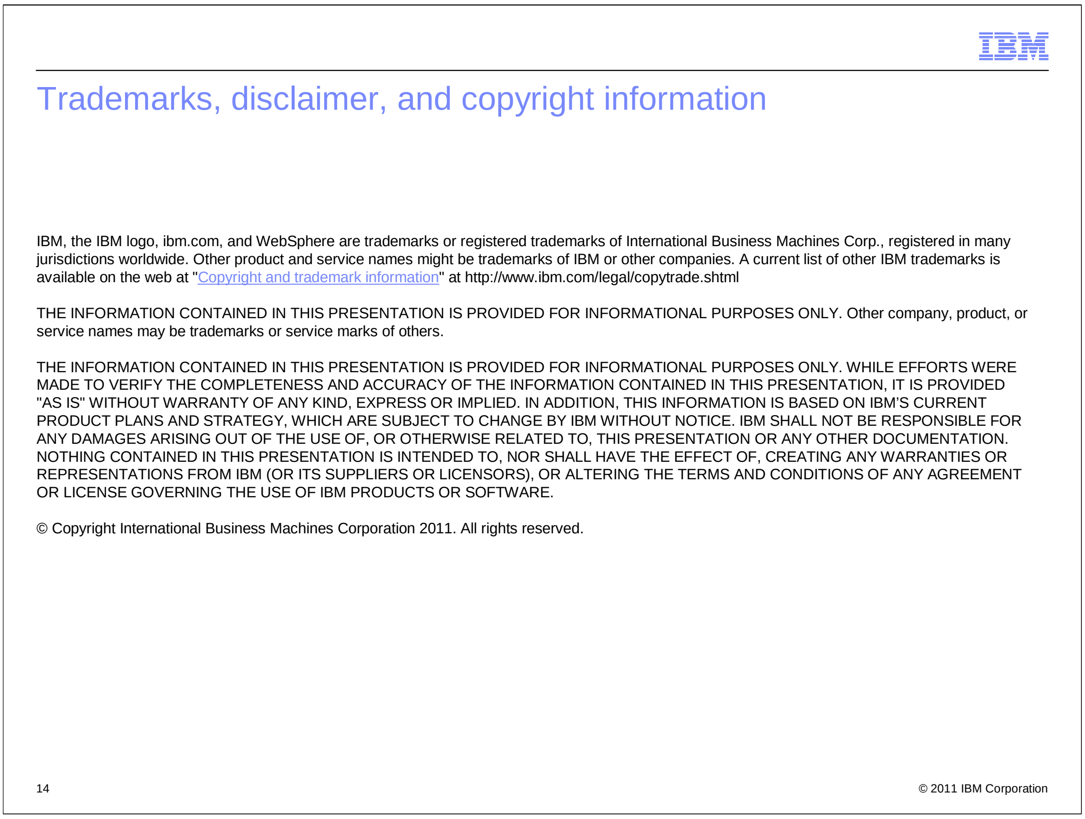
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASV8_Performance.ppt

This module is also available in PDF format at: ../WASV8_Performance.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.