Unit 20. Performance monitoring

What this unit is about

This unit describes performance monitoring methods and tools that are available through the administrative console.

What you should be able to do

After completing this unit, you should be able to:

- Describe performance monitoring and tuning methods
- Use the Tivoli Performance Viewer to monitor application server resources
- Use the performance servlet to generate performance data
- Configure the Request Metrics tool to generate performance data about the end-to-end request flow
- Use Performance Advisors to generate suggested tuning actions
- Enable the performance collectors from IBM Tivoli Composite Application Manager for WebSphere Application Server

How you will check your progress

- Checkpoint questions
- Lab exercises

References

WebSphere Application Server V8.5 Information Center, Monitoring, and Tuning Performance:

`http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp`

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Unit objectives

After completing this unit, you should be able to:

- Describe performance monitoring and tuning methods
- Use the Tivoli Performance Viewer to monitor application server resources
- Use the performance servlet to generate performance data
- Configure the Request Metrics tool to generate performance data about the end-to-end request flow
- Use Performance Advisors to generate suggested tuning actions
- Enable the performance collectors from IBM Tivoli Composite Application Manager for WebSphere Application Server

© Copyright IBM Corporation 2013

Figure 20-1. Unit objectives WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Topics

- Performance tuning and monitoring
- Request metrics
- Performance advisors
- IBM Tivoli Composite Application Manager for WebSphere Application Server

© Copyright IBM Corporation 2013

Figure 20-2. Topics WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

20.1.Performance tuning and monitoring

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

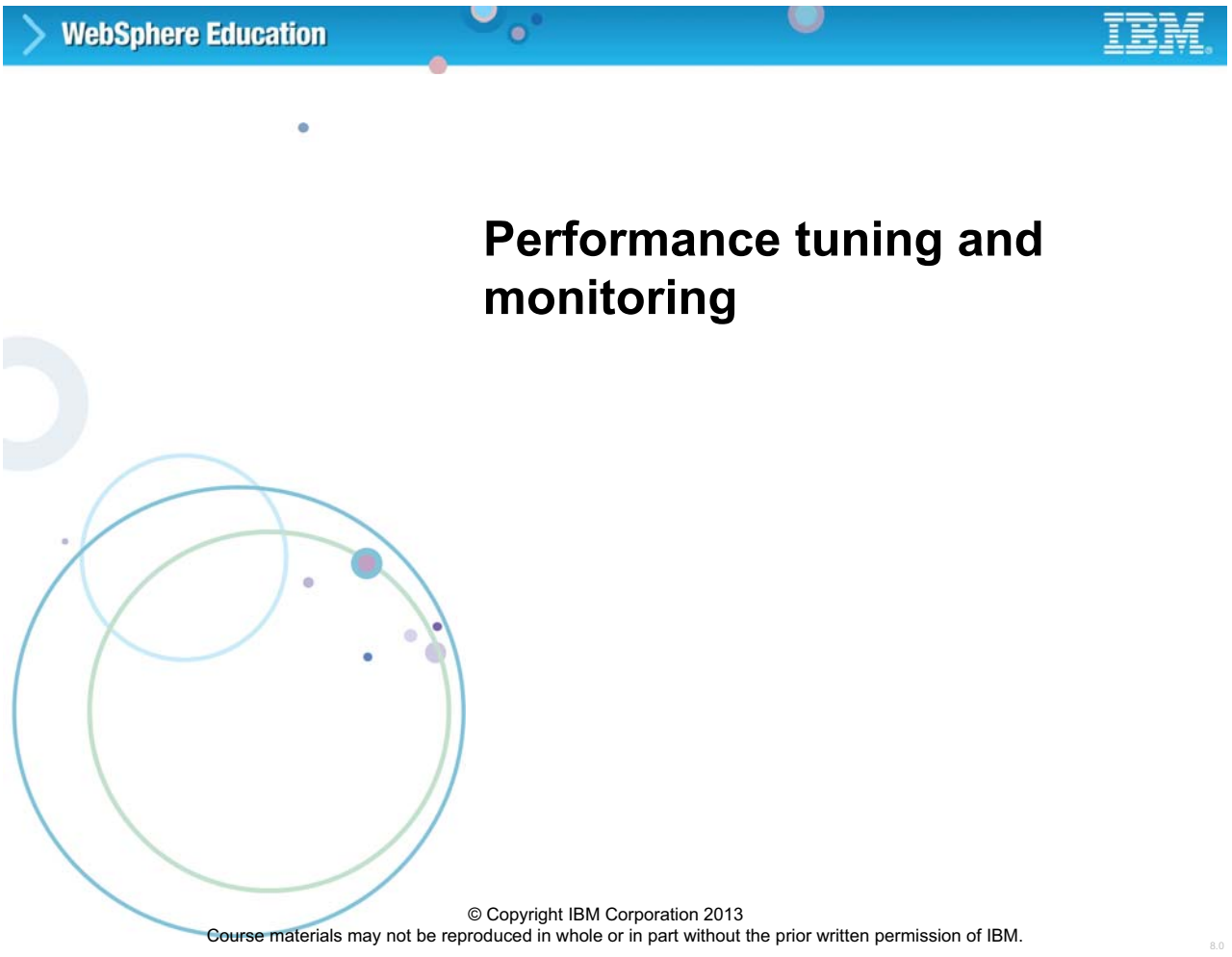


Figure 20-3. Performance tuning and monitoring WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

The need for performance monitoring and tuning

- How well a website performs while receiving heavy user traffic is an essential factor in the overall success of an organization
- Poor performance results in:
  - Escalated support costs
  - Loss of customer confidence
  - Loss of revenue
- Performance problems can be anywhere in the application server environment
  - Monitoring ensures that applications are running as expected and, if not, determines why and where the problem lies
- WebSphere Application Server can function with default settings but:
  - Improving throughput, and reducing server response times, requires more tuning

© Copyright IBM Corporation 2013

Figure 20-4. The need for performance monitoring and tuning WA855 / VA8551.0

Notes:

The goal of performance monitoring is to collect runtime statistics on your application and its environment to quantify their performance behavior. It allows you to determine whether your application meets its performance objectives and helps to identify any performance bottlenecks.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Tuning performance suggested practices

- Plan for performance
- Take advantage of performance functions (for example, use the dynamic cache service)
- Obtain performance advice from the advisors
- Tune the environment
- Troubleshoot performance problems

© Copyright IBM Corporation 2013

Figure 20-5. Tuning performance suggested practices WA855 / VA8551.0

Notes:

Tuning WebSphere Application Server is a critical part of getting the best performance from your website. But tuning WebSphere Application Server involves analyzing performance data and determining the optimal server configuration. This determination requires considerable knowledge about the various components in the application server and their performance characteristics. The performance advisors encapsulate this knowledge and analyze the performance data. The advisors provide configuration recommendations to improve the application server performance. Therefore, the performance advisors provide a starting point for tuning the application server. Keep in mind the following suggestions:

- Take advantage of performance functions.
- Obtain performance advice from the advisors.
- Tune the environment.
- Troubleshoot performance problems.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance terminology

- **Response time** measures an **individual** user's average wait for a request
- Response time includes:
  - Processing time
  - Transit time
  - Wait time in queues
- **Throughput** measures activities that are completed in a unit of time
  - Example: Website pages that are served per second
- **Bottleneck** defines a choke point in the system that is manifested as multiple threads that are waiting for some task to complete
- Bottlenecks result when users are queued waiting for a shared resource
  - Processor
  - Data source connections
  - Disk I/O
- **Load** is user activity against a website
  - Users arriving, logging in, sending requests
  - Requests per second, pages per hour

© Copyright IBM Corporation 2013

Figure 20-6. Performance terminology WA855 / VA8551.0

Notes:

Some other performance-related terms are path length, scalability, and capacity.

Path length refers to the number of steps an action takes. Reducing the path length speeds up a website or application. Path length reduction can be achieved by speeding up the steps and reducing the number of steps an activity takes.

Scalability defines how easily a site can expand. Sites must expand, sometimes with little warning, to support increased load. Load can come from many sources: new markets, normal growth, and extreme peaks in activity.

Capacity describes how much load the site can support. Discovering the website capacity is the result of performance and load testing.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Tuning parameter hot list

- Review hardware and software requirements
- Install the current refresh pack, fix pack, and the interim fixes
- Check hardware configuration and settings
- Tune the operating system
- Set the minimum and maximum Java virtual machine (JVM) heap sizes
- Use type 4 (pure Java) JDBC driver when feasible
- Tune WebSphere Application Server data sources and connection pools
- Enable the pass by reference option
- Tune related components, for example, the database
- Disable functions that are not required
- Review the application design

© Copyright IBM Corporation 2013

Figure 20-7. Tuning parameter hot list WA855 / VA8551.0

Notes:

This hot list contains recommendations that improve performance or scalability, or both, for many applications.

WebSphere Application Server provides several tunable parameters and options to match the application server environment to the requirements of your application.

- **Review the hardware and software requirements**.

  For correct functionality and performance, it is critical to satisfy the minimum hardware and software requirements. See the IBM WebSphere Application Server supported hardware, software, and APIs website, which details hardware and software requirements.

- **Install the most current refresh pack, fix pack, and suggested interim fixes**.

  The list of suggested updates is maintained on the support site.

- **Check hardware configuration and settings**.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Verify that network interconnections and hardware configuration are set up for peak performance.

- **Tune the operating systems**.

  Operating system configuration plays a key role in performance. For example, adjustments such as TCP/IP parameters might be necessary for your application.

- **Set the minimum and maximum Java virtual machine (JVM) heap sizes**.

  Many applications need a larger heap size than the default for best performance. It is also advisable to select an appropriate GC policy that is based on the characteristics of the application.

- **Use a type 4 (or pure Java) JDBC driver**.

  In general, the type 2 JDBC driver is suggested if the database exists on the same physical machine as the WebSphere instance. However, in the case where the database is in a different tier, the type 4 JDBC driver offers the fastest performance since it is pure Java and does not require native implementation.

- **Tune WebSphere Application Server JDBC data sources and associated connection pools**.

  The JDBC data source configuration might have a significant performance impact. For example, the connection pool size and prepared statement cache must be sized based on the number of concurrent requests that are processed and the design of the application.

- **Enable the pass by reference option**.

  Use applications that can take advantage of the pass by reference option to avoid the cost of copying parameters to the stack.

- **Ensure that the transaction log is assigned to a fast disk**.

  Some applications generate a high rate of writes to the WebSphere Application Server transaction log. Locating the transaction log on a fast disk or disk array can improve response time.

- **Tune related components; for example, database**.

  In many cases, some other component, for example a database, needs adjustments to achieve higher throughput for your entire configuration.

- **Disable functions that are not required**.

  For example, if your application does not use the web services addressing (WS-Addressing) support, disabling this function can improve performance. **Attention:** Use this property with care because applications might require WS-Addressing MAPs to function correctly. Setting this property also disables WebSphere Application Server support for the following specifications, which depend on the WS-Addressing support: Web Services Atomic Transactions, Web Services Business Agreement, and Web

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Services-Notification. To disable the support for WS-Addressing, see Enabling Web Services Addressing support for JAX-RPC applications.

- **Review your application design**.

  You can track many performance problems back to the application design. Review the design to determine whether it causes performance problems.

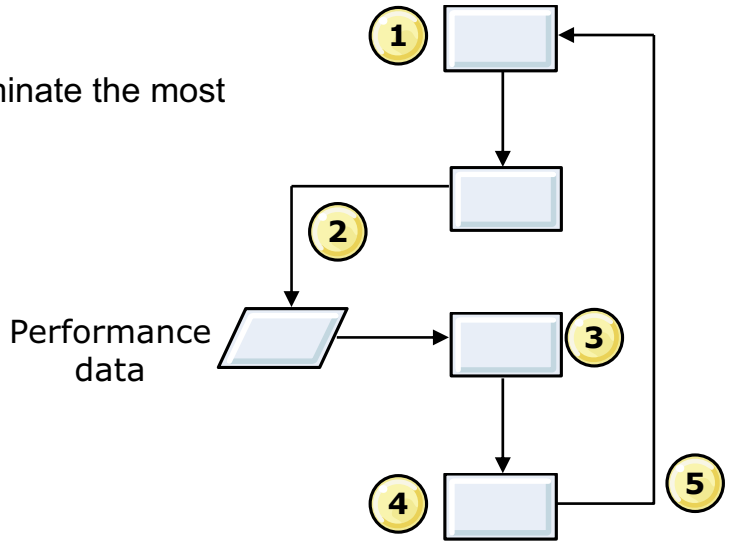
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

Solving performance problems

- An iterative process:
  1. Load test the system
  2. Monitor and collect performance data
  3. Identify bottlenecks
  4. Tune parameters to eliminate the most severe bottleneck
  5. Repeat



© Copyright IBM Corporation 2013

Figure 20-8. Solving performance problems WA855 / VA8551.0

Notes:

Your application and its runtime environment must also be tuned optimally. This process entails conducting many iterations of a monitor, tune, and test cycle. In short, monitoring, performance testing, and tuning are essential tasks for ensuring a well-performing, application-serving environment.

This process is often iterative because when one bottleneck is removed, some other part of the system now constrains the performance. For example, replacing slow hard disks with faster ones might shift the bottleneck to the processor of a system.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Measuring performance and collecting data

- Establish a benchmark:
  - A **benchmark** is a standard set of application functions to run
  - Use the benchmark to test the application under expected loads
  - Record throughput and response time under normal load and peak load
- Two types of performance data:
  - WebSphere Application Server Performance Monitoring Infrastructure (PMI) provides performance data that you can use to tune application server performance
  - With the Request Metrics tool, you can track individual transactions, recording the processing time in each of the major WebSphere Application Server components

© Copyright IBM Corporation 2013

Figure 20-9. Measuring performance and collecting data WA855 / VA8551.0

Notes:

Begin by choosing a *benchmark*, a standard set of operations to run. This benchmark exercises those application functions experiencing performance problems. Complex systems frequently need a warm-up period to cache objects and optimize code paths. System performance during the warm-up period is much slower than after the warm-up period. The benchmark must be able to generate work that warms up the system before recording the measurements that are used for performance analysis. Depending on the system complexity, a warm-up period can range from a few thousand transactions to longer than 30 minutes.

If the performance problem under investigation occurs only when many clients use the system, then the benchmark must also simulate multiple users. Another key requirement is that the benchmark must be able to produce repeatable results. If the results vary more than a few percent from one run to another, consider the possibility that the initial state of the system might not be the same for each run. It might also be that the measurements are made during the warm-up period, or that the system is running more workloads.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Several tools facilitate benchmark development. The tools range from tools that merely invoke a URL to script-based products that can interact with dynamic data that the application generates. IBM Rational has tools that can generate complex interactions with the system under test and simulate thousands of users. Producing a useful benchmark requires effort and must be part of the development process. Do not wait until an application goes into production to determine how to measure performance.

The benchmark records throughput and response time results in a form to allow graphing and other analysis techniques. The performance data that WebSphere Application Server Performance Monitoring Infrastructure (PMI) provides helps to monitor and tune the application server performance. Request metrics are another source of performance data that WebSphere Application Server provides. Request metrics allow a request to be timed at WebSphere Application Server component boundaries, enabling a determination of the time that is spent in each major component.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere performance tools (1 of 3)

- WebSphere provides integrated tools to monitor and tune system and application performance:
- Tivoli Performance Viewer
  - With Tivoli Performance Viewer, administrators can monitor the overall health of WebSphere Application Server
  - Accessed from within the administrative console
- Performance advisors
  - Analyze collected performance data and provide configuration recommendations to improve the application server performance
  - Output can be viewed in Tivoli Performance Viewer or in administrative console runtime messages

© Copyright IBM Corporation 2013

Figure 20-10. WebSphere performance tools (1 of 3) WA855 / VA8551.0

Notes:

These tools are explained in greater detail in the subsequent sections.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere performance tools (2 of 3)

- Request Metrics (tool)
  - With request metrics, you can track individual transactions, recording the processing time in each of the major WebSphere Application Server components
  - Output is viewed in standard logs or by using an Application Response Measurement (ARM)-based tool
- Performance servlet
  - Provides simple retrieval of performance data in XML format
  - Accessed through a browser
- IBM Tivoli Composite Application Manager for WebSphere Application Server
  - Installed separately
  - Provides other IBM Tivoli Composite Application Manager collectors for Java EE applications
  - Performance metrics are viewable as a Tivoli Performance Viewer performance module

© Copyright IBM Corporation 2013

Figure 20-11. WebSphere performance tools (2 of 3) WA855 / VA8551.0

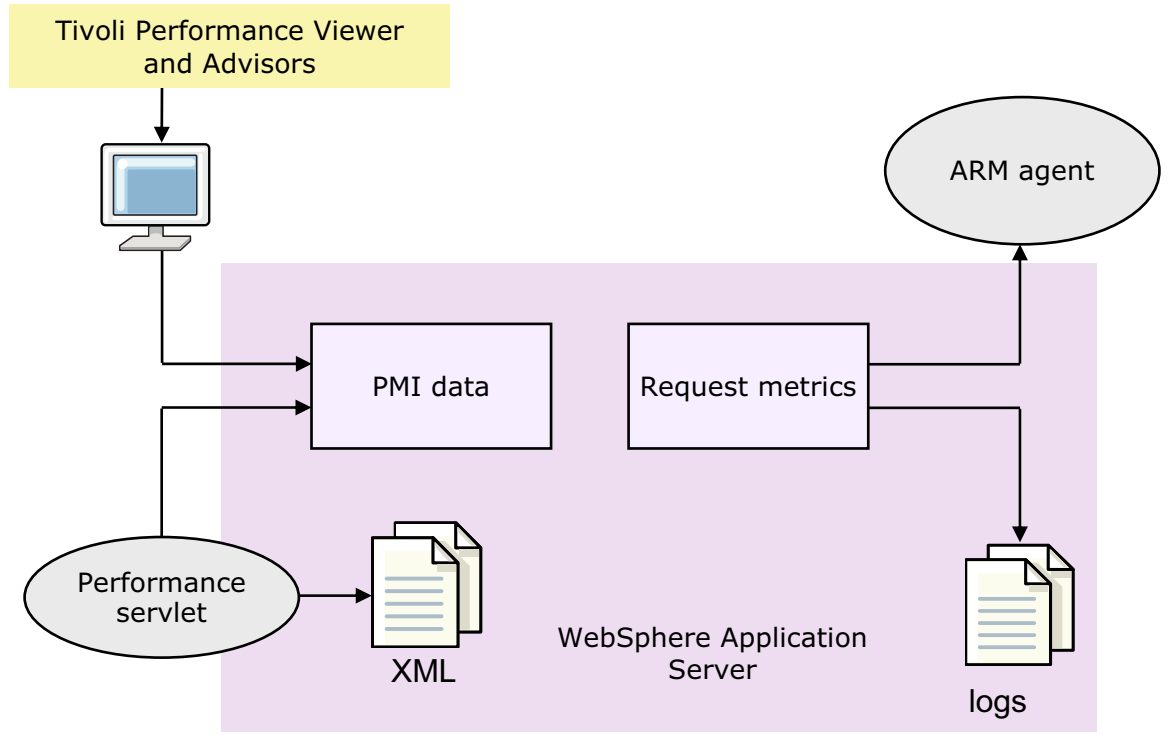
Notes:

These tools are explained in greater detail in the subsequent topics.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

WebSphere performance tools (3 of 3)



ARM= Application Response Measurement

© Copyright IBM Corporation 2013

Figure 20-12. WebSphere performance tools (3 of 3) WA855 / VA8551.0

Notes:

WebSphere provides integrated tools to monitor and tune system and application performance:

- Tivoli Performance Viewer
  - Gives administrators the ability to monitor the overall health of WebSphere Application Server
  - Accessed from within the administrative console
- Request metrics (tool)
  - Gives you the ability to track individual transactions, recording the processing time in each of the major WebSphere Application Server components
  - Output is viewed in standard logs or by using an Application Response Measurement (ARM)-based tool
- Performance advisors

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

- Analyze collected performance data and provide configuration recommendations to improve the application server performance
- Output is viewed in Tivoli Performance Viewer or in administrative console runtime messages

- Performance servlet
  - Provides simple retrieval of performance data in XML format
  - Accessed through a browser

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

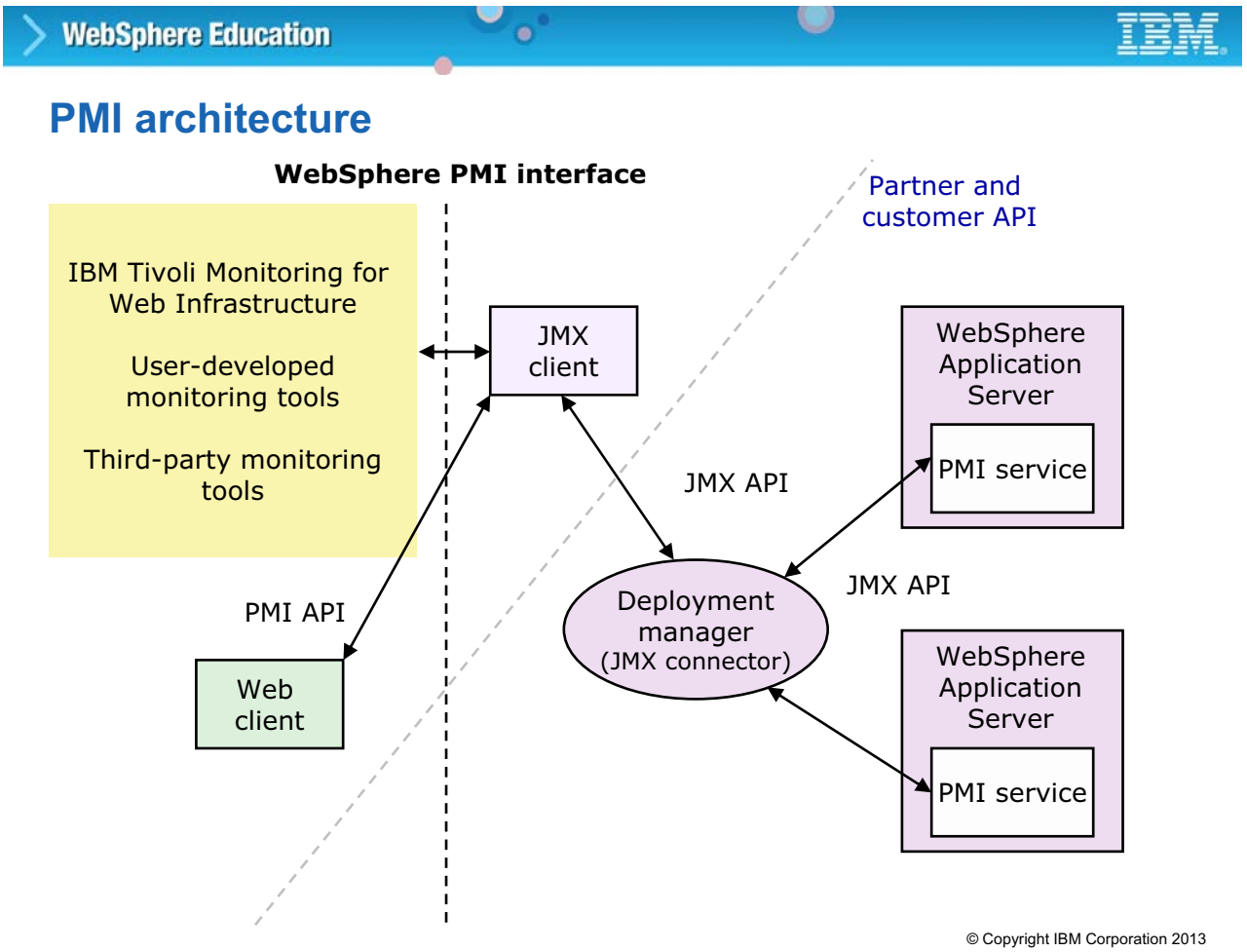


Figure 20-13. PMI architecture WA855 / VA8551.0

Notes:

The Performance Monitoring Infrastructure (PMI) uses a client/server architecture.

The figure shows the overall PMI architecture. On the right side, the server updates and keeps PMI data in memory. The left side displays a web client, a Java client, and a JMX client that retrieves the performance data. This data consists of counters such as servlet response time and data connection pool usage. The data points are then retrieved by using a web client, a Java client, or a Java Management Extensions (JMX) client. WebSphere Application Server contains Tivoli Performance Viewer, a Java client that displays and monitors performance data.

The server collects performance data from various WebSphere Application Server components. A client retrieves performance data from one or more servers and processes the data. WebSphere Application Server supports the Java Platform, Enterprise Edition (Java EE) Management Reference Implementation (JSR-77).

PMI counters are enabled, based on a monitoring or instrumentation level. The levels are None, Basic, Extended, All, and Custom. These levels are specified in the PMI module XML file. Enabling the module at a certain level includes all the counters at that level plus

© Copyright IBM Corp. 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

counters from levels below that level. So, enabling the module at the extended level enables all the counters at that level plus all the Basic level counters.

JSR-077 defines a set of statistics for Java EE components as part of the Statistic Provider interface. The PMI monitoring level of Basic includes all of the JSR-077 specified statistics. PMI is set to monitor at a Basic level by default.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Types of performance data

- System resources such as processor utilization
- WebSphere pools and queues, such as a database connection pool
- Customer application data, such as average servlet response time
- You can also view data for other products or customer applications that implement custom PMI by using the Tivoli Performance Viewer

© Copyright IBM Corporation 2013

Figure 20-14. Types of performance data WA855 / VA8551.0

Notes:

Tivoli Performance Viewer is used to help manage configuration settings by viewing the various graphs or by using the Tivoli Performance Advisor. For example, by looking at the summary chart for thread pools, you can determine whether the thread pool size must be increased or decreased by monitoring the percent usage. After configuration settings are changed based on the data that is provided, you can determine the effectiveness of the changes. To help with configuration settings, use the Tivoli Performance Advisor. The Advisor assesses various data while your application is running, and provides advice about configuration settings to improve performance.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

PMI data collection settings

- None
  - All statistics are disabled
- Basic
  - Statistics that are specified in Java EE specification, plus top statistics like processor usage and live HTTP sessions, are enabled
  - This set is enabled by default and provides basic performance data about runtime and application components **(up to 2% more processor usage)**
- Extended
  - Basic set, plus key statistics from various WebSphere Application Server components like WLM and dynamic caching, are enabled
  - This set provides detailed performance data about various runtime and application components **(up to 3% more processor usage)**
- All
  - All statistics are enabled **(up to 6% more processor usage)**
- Custom
  - Statistics are enabled or disabled individually

© Copyright IBM Corporation 2013

Figure 20-15. PMI data collection settings WA855 / VA8551.0

Notes:

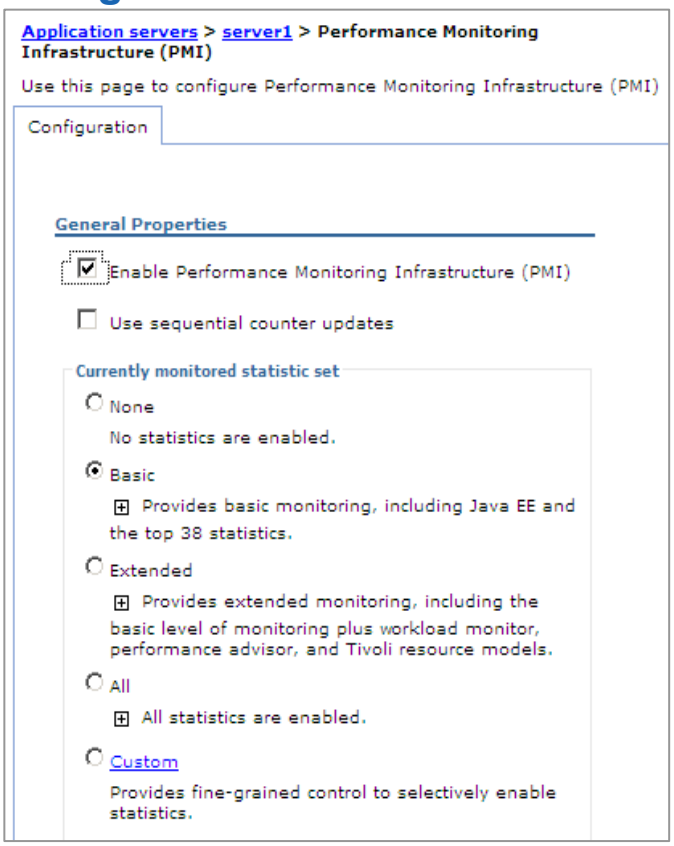
PMI uses statistics sets to specify the type and amount of performance data to collect.

PMI counters are enabled, based on a monitoring or instrumentation level. The levels are None, Basic, Extended, All, and Custom. These levels are specified in the PMI module XML file. Enabling the module at a certain level includes all the counters at that level plus counters from levels below that level. So, enabling the module at the extended level enables all the counters at that level plus all the Basic level counters as well.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Using the administrative console to enable PMI



- Click **Servers > Server Types > WebSphere Application Servers > *server_name***
- On the Configuration tab, under Performance, click **Performance Monitoring Infrastructure (PMI)**
- Select the **Enable Performance Monitoring Infrastructure (PMI)** check box
- Select the statistics set

© Copyright IBM Corporation 2013

Figure 20-16. Using the administrative console to enable PMI WA855 / VA8551.0

Notes:

Click **Servers > Server Types > WebSphere Application Servers > <*server_name*>**.

On the Configuration tab, under Performance, click **Performance Monitoring Infrastructure (PMI)**.

Select the **Enable Performance Monitoring Infrastructure (PMI)** check box.

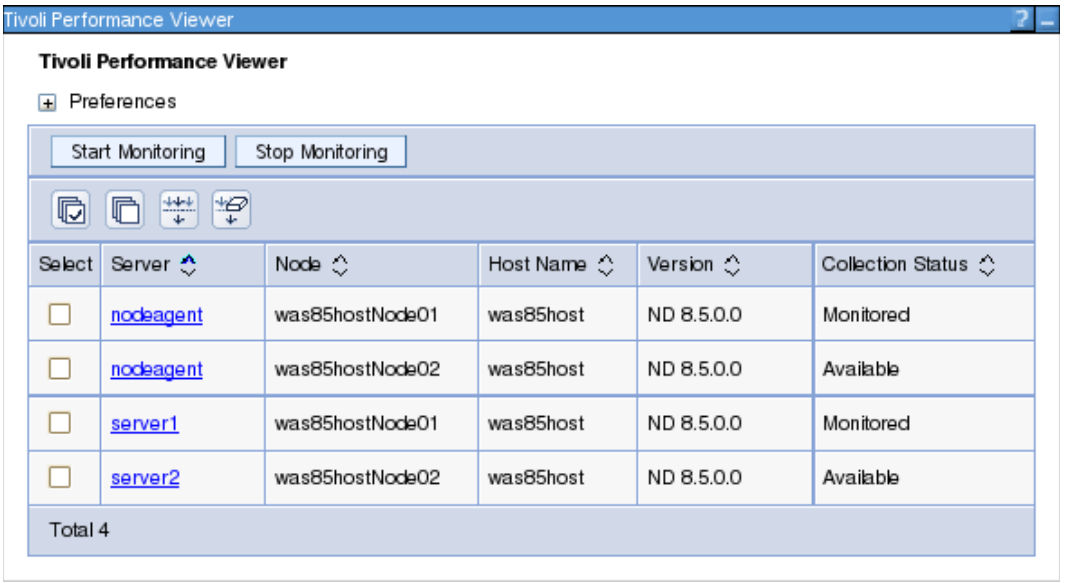
 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

Start monitoring

- After enabling PMI, select the server and click **Start Monitoring** on the Tivoli Performance Viewer page
  - In the administrative console, select **Monitoring and Tuning > Performance Viewer > Current activity**



© Copyright IBM Corporation 2013

Figure 20-17. Start monitoring WA855 / VA8551.0

Notes:

- **Deprecated feature**: Tivoli Performance Viewer displays graphics in either the Scalable Vector Graphics (SVG) format or as a static image in the JPG format. If you do not have the Adobe SVG browser plug-in, you are prompted to download and install it. If you select not to install the plug-in (by selecting Cancel), Tivoli Performance Viewer displays the static image.

  Installing the Adobe SVG plug-in is advantageous for several reasons. First, the SVG format provides interactive graphics that provide more information when you hover your mouse over a point, line, or legend item. The SVG format also allows you to click a point and see details for it. Second, using the SVG format provides a performance benefit because the work to display the SVG image is done on the client side. When viewing a static image, the application server must convert the SVG image into a static image, which is a processor-intensive and memory-intensive operation. If your browser is Internet Explorer 7, the Adobe SVG installation prompt might be inaccessible. To resolve the problem, you can reinstall Adobe SVG.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

- **For users who are migrating from version 7**: Beginning with version 8, the Tivoli Performance Viewer graph uses Dojo technology for plotting the performance activity rather than the Scalable Vector Graphics (SVG) format. The Dojo format provides a better user experience and is more processor and memory efficient for the application server. The SVG format is still supported but is deprecated in version 8 of this product. To use the SVG format and image format, set the JVM property to false; for example: `com.ibm.websphere.tpv.DojoGraph=false`

  If the property is set to false, Dojo is disabled, and Tivoli Performance Viewer uses the SVG format to display interactive graphics; or it uses the JPG format to display non-interactive graphics.

  When you specify to use the SVG format by setting `com.ibm.websphere.tpv.DojoGraph=false`, if you do not have the Adobe SVG browser plug-in, you are prompted to download and install it. If you select not to install the plug-in (by selecting Cancel), Tivoli Performance Viewer displays the static image. If your browser is Internet Explorer 7, the Adobe SVG installation prompt might be inaccessible. To resolve the problem, you can reinstall Adobe SVG. By default, this property is set to a value of true to use the Dojo format.

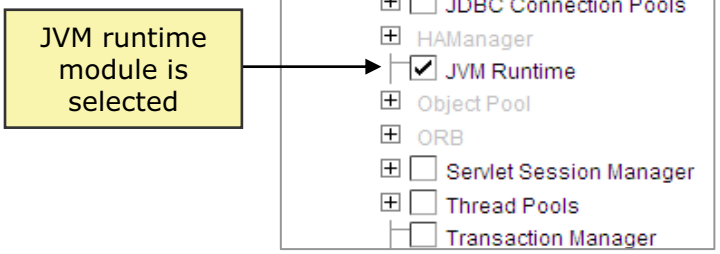
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Tivoli Performance Viewer (1 of 4)

- Select one or more performance modules to monitor from the navigation page
- Click **View Module(s)**
- The performance data is dynamically displayed in a chart and table
- Note: The disabled modules become active when you enable the Extended or All PMI statistics sets



© Copyright IBM Corporation 2013

Figure 20-18. Tivoli Performance Viewer (1 of 4) WA855 / VA8551.0

Notes:

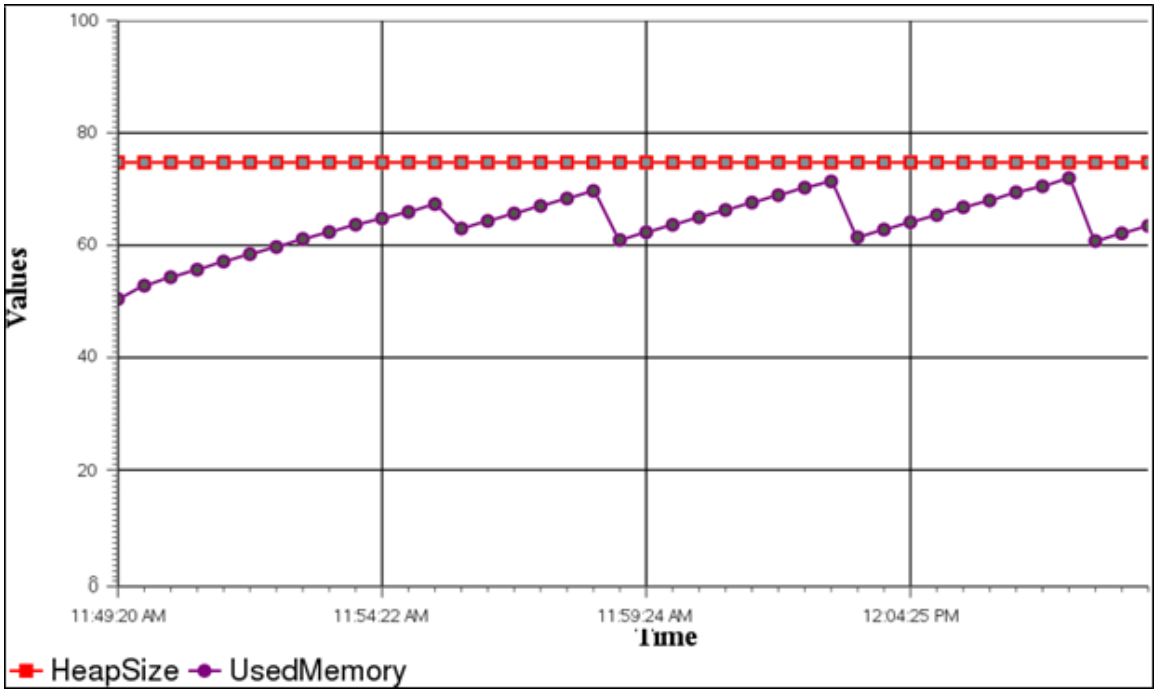
The screen capture on this slide shows the navigation tree in Tivoli Performance Viewer where you can select which components to monitor. In this case, the JVM Runtime module is selected.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Tivoli Performance Viewer (2 of 4)

- Chart view of JVM metrics



© Copyright IBM Corporation 2013

Figure 20-19. Tivoli Performance Viewer (2 of 4) WA855 / VA8551.0

Notes:

Statistics for the selected modules are displayed as a line graph. You can select which metrics you want to display in the graph, and you can optionally show the legend. In this example, only the heap size and used memory metrics are displayed for the JVM runtime. The sawtooth pattern of the used memory graph is typical of a steady state JVM. The periodic reductions in used memory correspond to JVM garbage collections, which return unused memory to the heap.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Tivoli Performance Viewer (3 of 4)

- Chart view controls
  - Reset To Zero
  - Clear Buffer
  - View Table
  - Show/Hide Legend

© Copyright IBM Corporation 2013

Figure 20-20. Tivoli Performance Viewer (3 of 4) WA855 / VA8551.0

Notes:

The **Reset to zero** button sets a new baseline by using the current counter readings at the instant the button is clicked. Future data points are plotted on the graph relative to their position at the time **Reset to zero** is clicked. Data points that are gathered before the time **Reset to zero** is clicked are not displayed, although they are still held in the Tivoli Performance Viewer buffer. If **Undo Reset to zero** is clicked again, Tivoli Performance Viewer displays all data that is recorded from the original baseline, not from the **Reset to zero** point.

Click **Clear Buffer** to remove the PMI data from a table or chart.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Tivoli Performance Viewer (4 of 4)

- Table view

Time	JVM Runtime HeapSize	JVM Runtime FreeMemory	JVM Runtime UsedMemory
12:29:29 PM	86528.00	30979.00	55548.00
12:28:59 PM	86528.00	32317.00	54210.00
12:28:29 PM	86528.00	18502.00	68025.00
12:27:59 PM	86528.00	19513.00	67014.00
12:27:29 PM	86528.00	20768.00	65759.00
12:26:59 PM	86528.00	22197.00	64330.00
12:26:29 PM	86528.00	23452.00	63075.00

© Copyright IBM Corporation 2013

Figure 20-21. Tivoli Performance Viewer (4 of 4) WA855 / VA8551.0

Notes:

Statistics can also be viewed in a table format, by clicking **View Table** in the Tivoli Performance Viewer.

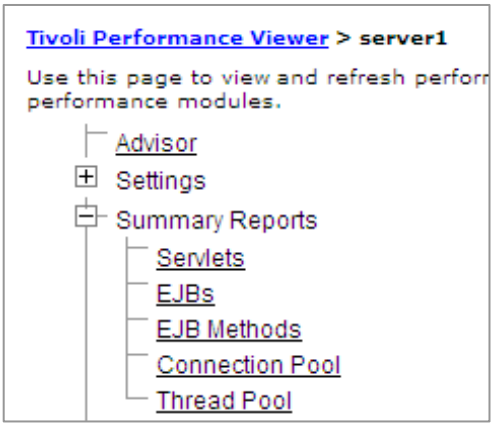
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Summary reports

- View a statistics report by selecting one of the summary reports
- Servlets
  - Lists all servlets that are running in the current application server
- EJBs
  - Lists all EJBs running in the server
  - Amount of time that is spent in their methods
  - Number of EJB invocations
  - Total time that is spent in each EJB
- EJB methods
  - Details about methods
- Connection pool
  - Lists all data source connections that are defined in the application server and show their usage over time
- Thread pool
  - Shows the usage of all thread pools in the application server over time



© Copyright IBM Corporation 2013

Figure 20-22. Summary reports WA855 / VA8551.0

Notes:

Servlets:

The servlet summary lists all servlets that are running in the current application server. Use the servlet summary view to quickly find the most time-intensive servlets and the applications that use them, and to determine which servlets are used most often. You can sort the summary table by any of the columns.

Tip: Sort by **Average Response Time** to find the slowest servlet or JSP page. Sort by **Total Requests** to find the servlet or JSP that is used the most. Sort by **Total Time** to find the most costly servlet or JSP.

Enterprise JavaBeans:

The Enterprise JavaBeans (EJB) summary lists the following information: all enterprise beans that are running in the server, the amount of time that is spent in their methods, the number of EJB invocations, and the total time that is spent in each enterprise bean.

```
total_time = number_of_invocations * time_in_methods
```


Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Sort the various columns to find the most expensive enterprise bean. Also, if the PMI counters are enabled for individual EJB methods, there is a check box next to the EJB name that you can select to see statistics for each of the methods.

Tip: Sort by **Average Response Time** to find the slowest enterprise bean. Sort by **Method Calls** to find the enterprise bean that is used the most. Sort by **Total Time** to find the most costly enterprise bean.

EJB methods:

The EJB method summary shows statistics for each EJB method. Use the EJB method summary to find the most costly methods of your enterprise beans.

Tip: Sort by **Average Response Time** to find the slowest EJB method. Sort by **Method Calls** to find the EJB method that is used the most. Sort by **Total Time** to find the most costly EJB method.

Connection pools:

The connection pool summary lists all data source connections that are defined in the application server and shows their usage over time.

When the application is experiencing normal to heavy usage, the pools that the application uses must be nearly fully used. Low utilization means that resources are being wasted by maintaining connections or threads that are never used. Consider the order in which work progresses through the various pools. If the resources near the end of the pipeline are underused, it might mean that resources near the front are constrained or that more resources than necessary are allocated near the end of the pipeline.

Thread pools:

The thread pool summary shows the usage of all thread pools in the application server over time.

When the application is experiencing normal to heavy usage, the pools that the application uses must be nearly fully used. Low utilization means that resources are being wasted by maintaining connections or threads that are never used. Consider the order in which work progresses through the various pools. If the resources near the end of the pipeline are underused, it might mean that resources near the front are constrained or that more resources than necessary are allocated near the end of the pipeline.

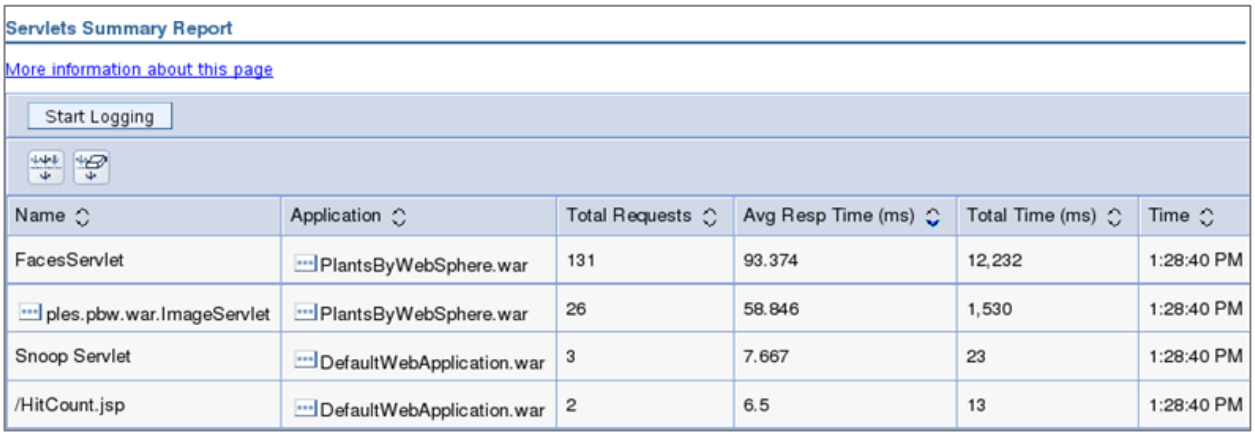
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Example: Servlet Summary Report

- Use the servlet summary to:
  - Find the servlets that use the most time and the applications that use them
  - Determine which servlets are called most often
- You can sort the summary table by any of the columns



Servlets Summary Report

More information about this page

Start Logging

Name	Application	Total Requests	Avg Resp Time (ms)	Total Time (ms)	Time
FacesServlet	PlantsByWebSphere.war	131	93.374	12,232	1:28:40 PM
ples.pbw.war.ImageServlet	PlantsByWebSphere.war	26	58.846	1,530	1:28:40 PM
Snoop Servlet	DefaultWebApplication.war	3	7.667	23	1:28:40 PM
/HitCount.jsp	DefaultWebApplication.war	2	6.5	13	1:28:40 PM

© Copyright IBM Corporation 2013

Figure 20-23. Example: Servlet Summary Report WA855 / VA8551.0

Notes:

In this screen capture, the servlets report is shown with the total requests column sorted. You can see which two servlets in the PlantsByWebSphere application are used most frequently.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Performance servlet overview

- Provides performance data output as an XML document, as the provided document type definition (DTD) describes
  - The DTD is located inside the `PerfServletApp.ear` file
- Deployed exactly as any other servlet:
  1. Deploy the servlet on a single application server instance within the domain
  2. After the servlet deploys, you can start it to retrieve performance data for the entire domain; start the performance servlet by accessing the following default URL:
     `http://<hostname>/wasPerfTool/servlet/perfservlet`

© Copyright IBM Corporation 2013

Figure 20-24. Performance servlet overview WA855 / VA8551.0

Notes:

The servlet provides a way to use an HTTP request to query the performance metrics for an entire WebSphere Application Server administrative domain. Because the servlet provides the performance data through HTTP, issues such as firewalls are trivial to resolve.

The PerfServlet provides the performance data output as an XML document, as described in the provided document type definition (DTD). In the XML structure, the leaves of the structure provide the actual observations of performance data and the paths to the leaves that provide the context.

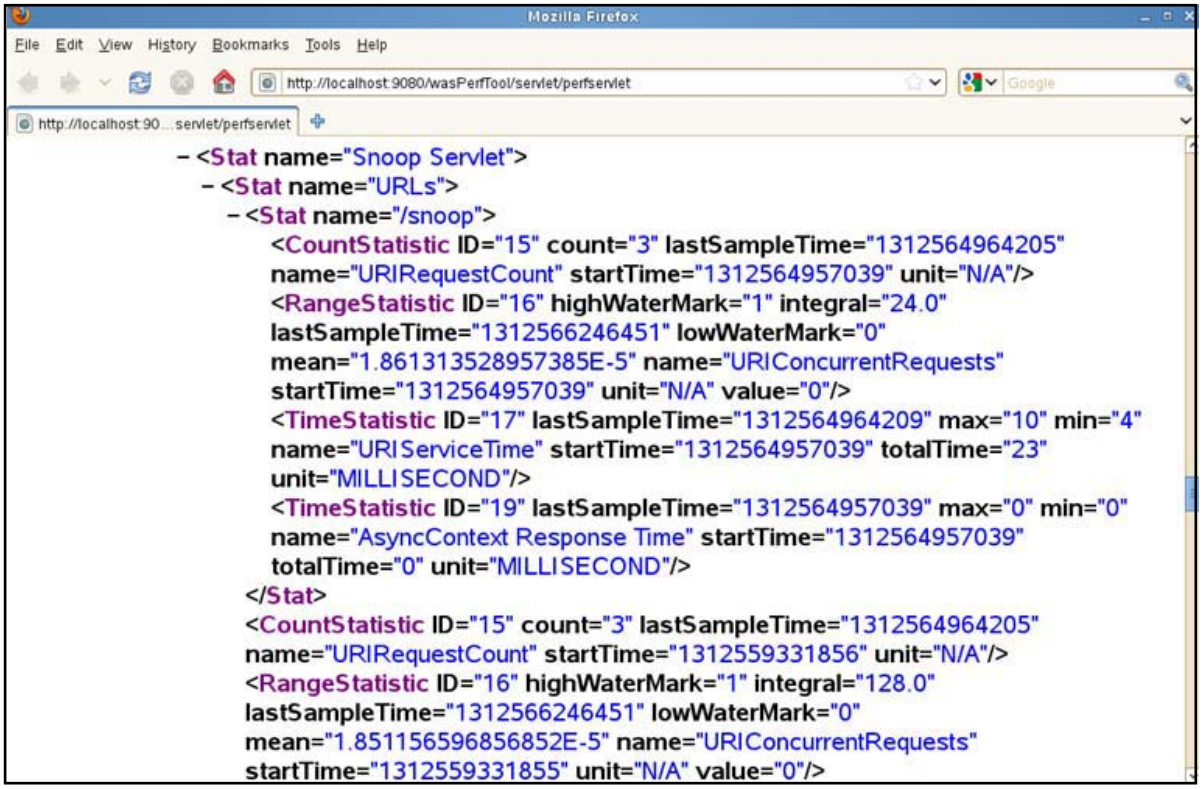
The performance servlet EAR file `PerfServletApp.ear` is in the *`WAS_HOME`*`/installableApps` directory, where *`WAS_HOME`* is the installation path for WebSphere Application Server.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

Performance servlet output



© Copyright IBM Corporation 2013

Figure 20-25. Performance servlet output WA855 / VA8551.0

Notes:

This screen capture shows sample output that the performance servlet generates, which is displayed in a browser. The section that is shown here represents a request to the Snoop Servlet.

Tip: The PerfServlet is a sample monitoring tool that uses WebSphere Application Server administration and monitoring interfaces to extract and display performance data. Using the PerfServlet is not intended for real-time performance monitoring in production environments or for use in large topologies. For these environments, you might use the Tivoli Performance Viewer or IBM Tivoli Composite Application Manager for WebSphere Application Server.

Specific tips for the PerfServlet include the following practices:

- **PerfServlet resource usage**: The PerfServlet is not designed to run concurrently. Being a single threaded servlet, it would collect the data sequentially from available servers. This single threaded operation can cause higher response times when the PerfServlet is used in larger deployments.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

- **PerfServlet in large deployments**: By default, when the PerfServlet is first initialized, it retrieves the list of nodes and servers within the cell in which it is deployed. Because collecting this data requires system processing time, the PerfServlet holds this information as a cached list. To force the servlet to refresh its configuration, you can use the option "`refreshconfig=true`". However, using this option is not suggested unless required because this option adds extra resource usage to the PerfServlet processing. Use option, node, and server, if you are looking for performance data of a specific server.
- **PerfServlet response time**: How responsive the PerfServlet is depends on the following factors: numbers of application servers that exist in the cell and number of resources that are configured in the cell (including applications).
- **PerfServlet alternative**: If you are looking for an alternative to using the PerfServlet to capture data programmatically, see the Perf MBean programming interfaces documentation. The documentation can be found under the **Reference > Programming Interfaces > MBean** interfaces section of the WebSphere Application Server Information Center.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

20.2.Request metrics

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Request metrics

© Copyright IBM Corporation 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

8.0

Figure 20-26. Request metrics WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Request metrics (RM) overview

- Tracing facility that allows you to measure the amount of time a request spends in each component that is traversed during its execution
- Captured information includes:
  - Elapsed time in the web server
  - Response time of invoked components in the web and EJB containers
  - Response time of related JDBC calls
- Writes trace records to `SystemOut.log` or sends metrics to an Application Response Measurement (ARM) agent

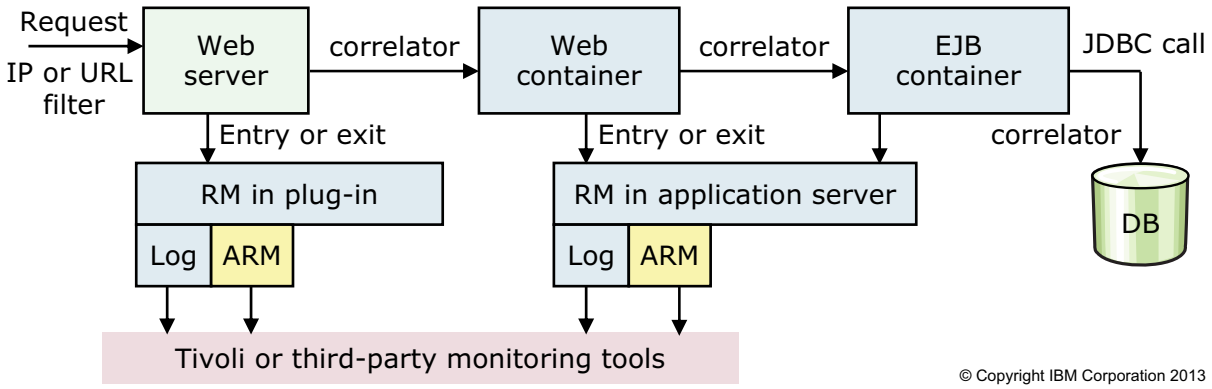


Figure 20-27. Request metrics (RM) overview WA855 / VA8551.0

Notes:

Request metrics allow you to monitor the transaction flow and analyze the response time of the components that are involved in processing it. This analysis can help you target performance problem areas and debug resource constraint problems. For example, it can help determine whether a transaction spends most of its time in the web server plug-in, the web container, the Enterprise JavaBeans (EJB) container, or the back-end database. The response time that is collected for each level includes the time that is spent at that level and the time that is spent in the lower levels. For example, if the total response time for the servlet is 130 milliseconds, and it includes 38 milliseconds from the enterprise beans and JDBC calls, then 92 ms can be attributed to the servlet process.

An ARM agent is not included with WebSphere Application Server, but third-party tools can provide it.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

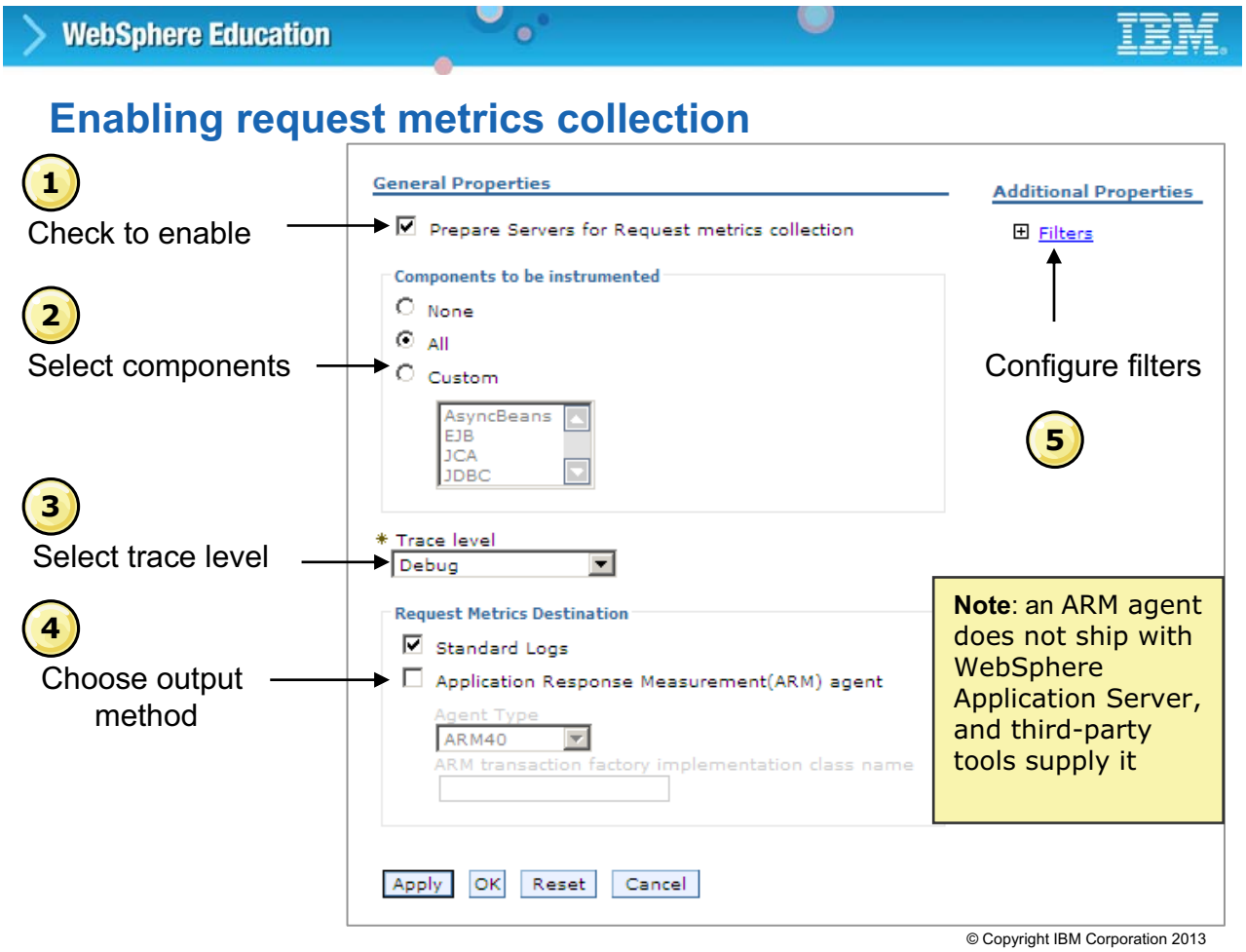


Figure 20-28. Enabling request metrics collection WA855 / VA8551.0

Notes:

In the administrative console, select **Monitoring and Tuning > Request Metrics** and select the check box to **Prepare Servers for Request metrics collection**.

Trace level specifies how much trace data to accumulate for a particular transaction. **Trace level** and **Components to be instrumented** work together to control whether a request is instrumented or not. The trace level can be set to one of the following values:

- **None**: No instrumentation.
- **Hops**: Generates instrumentation information about process boundaries only. When this setting is selected, you see the data at the application server level, not the level of individual components such as enterprise beans or servlets.
- **Performance_debug**: Generates the data at Hops level and the first level of the intra-process servlet and Enterprise JavaBeans (EJB) call (for example, when an inbound servlet forwards to a servlet and an inbound EJB calls another EJB). Other intra-process calls like naming and service integration bus (SIB) are not enabled at this level.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

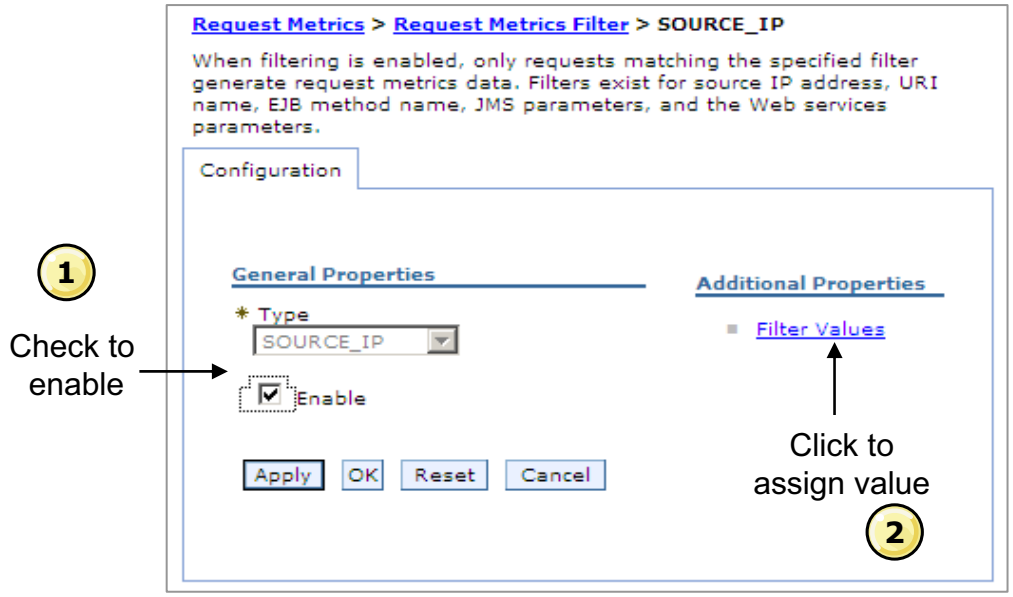
- **Debug**: Provides detailed instrumentation data, including response times for all intra-process calls. Note: Requests to servlet filters are only instrumented at this level.
- **Standard logs: E**nables the request metrics logging feature. Select this check box to trigger the generation of request metrics logs in the SystemOut.log file. Note: Since enabling the request metrics logging feature increases processor usage, it is suggested to use this feature together with filters so that only selected requests are instrumented.
- **Application Response Measurement (ARM) agent:** Allows request metrics to call an underlying Application Response Measurement (ARM) agent. Before enabling ARM, you must install an ARM agent and configure it to the appropriate class path and path, following the instructions of the ARM provider.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Isolating performance for specific types of requests

- Click **Monitoring and Tuning > Request Metrics > Filters**
- Select a filter type (for example, SOURCE_IP)
- Assign filter values



© Copyright IBM Corporation 2013

Figure 20-29. Isolating performance for specific types of requests WA855 / VA8551.0

Notes:

The request metrics filters are enabled according to your configuration. For example, if you enabled source IP, only requests whose source IP matches the one specified in the filter are instrumented.

Note: Filters are only checked for edge transactions. An edge transaction is the transaction that first enters an instrumented system. For example, if a servlet calls an Enterprise JavaBeans component, the servlet is the edge transaction. The servlet must not be instrumented at the web server plug-in, and the URI and SOURCE_IP filters must be checked for the servlet request. However, when the request comes to the EJB container, the EJB filter is not checked because it is no longer an edge transaction.

You must regenerate the web server plug-in configuration file after modifying the request metrics configuration.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

Example request metrics data

- Request metrics data from a `SystemOut.log` file

```
[8/5/11 16:12:31:338 EDT] 00000029 PmiRmArmWrapp I   PMRM0003I:
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32878,event=1 type=JDBC
detail=java.sql.PreparedStatement.executeQuery() elapsed=0

[8/5/11 16:12:31:346 EDT] 00000029 PmiRmArmWrapp I   PMRM0003I:
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32879,event=1 type=JDBC
detail=javax.resource.spi.XAResource.end(Xid, int) elapsed=0

[8/5/11 16:12:31:350 EDT] 00000029 PmiRmArmWrapp I   PMRM0003I:
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32880,event=1 type=JDBC
detail=javax.resource.spi.XAResource.commit(Xid, boolean) elapsed=0

[8/5/11 16:12:31:366 EDT] 00000029 PmiRmArmWrapp I   PMRM0003I:
parent:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32874,event=1 -
current:ver=1,ip=127.0.0.1,time=1312575082923,pid=4269,reqid=32881,event=1 type=JDBC
detail=javax.resource.spi.ManagedConnection.cleanup() elapsed=0
```

© Copyright IBM Corporation 2013

Figure 20-30. Example request metrics data WA855 / VA8551.0

Notes:

The example of request metrics data that is shown on this slide detail the use of the prepared statement cache to make an SQL call. You can trace the steps that are involved and timings for this database transaction.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

20.3.Performance advisors

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Performance advisors

© Copyright IBM Corporation 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 20-31. Performance advisors WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance advisors overview

- WebSphere provides two separate advisors:
  - Performance and Diagnostic Advisor – disabled by default
  - Tivoli Performance Viewer Advisor
- Both provide configuration advice that is based on collected PMI data on a per server basis
  - Advisors do not compare counters among different application servers
- Provides advice that is based on basic rules for tuning WebSphere Application Server
  - Rules are IBM-defined and nonconfigurable
- Advisors do not automatically tune WebSphere based on advice
  - Administrator must manually apply recommendations
  - Suggested settings must be checked against baseline performance to verify improvement: tune, test, monitor

© Copyright IBM Corporation 2013

Figure 20-32. Performance advisors overview WA855 / VA8551.0

Notes:

The advisors provide advice on the following application server resources: thread pools, persisted session sizes, cache sizes, and JVM heap size.

For example, consider the data source statement cache. It optimizes the processing of *prepared statements* and *callable statements* by caching those statements that are not used in an active connection. (Both statements are SQL statements that essentially run repeatable tasks without the costs of repeated compilation.) If the cache is full, an old entry in the cache is discarded to make room for the new one. The best performance is generally obtained when the cache is large enough to hold all of the statements that are used in the application. The PMI counter, prepared statement cache discards, indicates the number of statements that are discarded from the cache.

The performance advisors check this counter and provide recommendations to minimize the cache discards. Another example is thread or connection pooling. The idea behind pooling is to use an existing thread or connection from the pool instead of creating an instance for each request. Because each thread or connection in the pool consumes memory and increases the context-switching cost, the pool size is an important

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

configuration parameter. A pool that is too large can hurt performance as much as a pool that is too small. The performance advisors use PMI information about current pool usage, minimum or maximum pool size, and the application server processor utilization to suggest efficient values for the pool sizes.

The advisors can also issue diagnostic advice to help in problem determination and health monitoring. For example, if your application requires more memory than is available, the diagnostic adviser tells you to increase the size of the heap for the application server.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance and Diagnostic Advisor (1 of 5)

- Performance advice:
  - Object Request Broker (ORB) service thread pools
  - Web container thread pools
  - Connection pool size
  - Persisted session size and time
  - Prepared statement cache size
  - Session cache size
  - Memory leak detection

- Diagnostic advice:
  - Connection factory diagnostic messages
  - Data source diagnostic messages

- Connection usage diagnostic messages
  - Detection of connection use by multiple threads
  - Detection of connection use across components

© Copyright IBM Corporation 2013

Figure 20-33. Performance and Diagnostic Advisor (1 of 5) WA855 / VA8551.0

Notes:

The Performance and Diagnostic Advisor runs in the Java virtual machine (JVM) process of the application server; therefore, the performance cost is minimal.

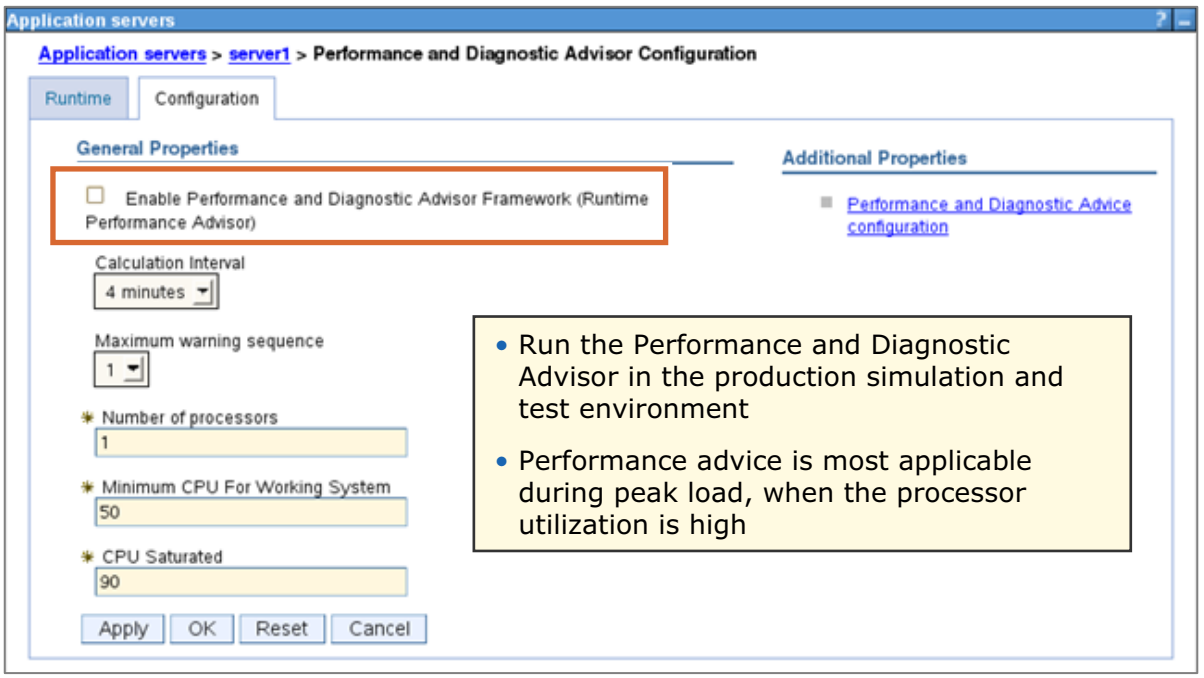
To access the Performance and Diagnostic Advisor Configuration, click **Servers > Server Types > WebSphere application servers > *server_name* > Performance and Diagnostic Advisor Configuration.**

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance and Diagnostic Advisor (2 of 5)

- Click **Servers > Server Types > WebSphere application servers > *server_name* > Performance and Diagnostic Advisor**



© Copyright IBM Corporation 2013

Figure 20-34. Performance and Diagnostic Advisor (2 of 5) WA855 / VA8551.0

Notes:

The Performance and Diagnostic Advisor analyzes PMI data and receives notifications about performance and diagnostic information from components. Use this page to specify settings for the Performance and Diagnostic Advisor. Performance issues can be related to memory leaks in the system. Use the Memory Dump Diagnostic for Java tool, a separate memory leak analysis utility, for detecting memory leaks.

The Performance and Diagnostic Advisor Framework is disabled by default. Each time that you enable it for an application server, you see the warning message:

Run the Performance and Diagnostic Advisor in the Production Simulation and Test environment. Performance advice is most applicable during peak load, when the processor utilization is high.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

Performance and Diagnostic Advisor (3 of 5)

- Advisor configuration panel (on both configuration and runtime tabs)
- Select advice and click Start or Stop

© Copyright IBM Corporation 2013

Figure 20-35. Performance and Diagnostic Advisor (3 of 5) WA855 / VA8551.0

Notes:

Advice type categorizes the primary indent of a piece of Advice.

Use Advice type for grouping, and then enable or disable sets of advice that is based on your performance goal. Advice has the following types:

- Performance: Performance advice provides tuning recommendations, or identifies problems with your configuration from a performance perspective.
- Diagnostic: Diagnostic advice provides automated logic and analysis that relates to problem identification and analysis. These types of advice are issued when the application server encounters unexpected circumstances.

Performance impact generalizes the negative effect on performance that an alert might incur.

The performance impact of a particular piece of advice is highly dependent upon the scenario that is run and upon the conditions that are met. The performance categorization of alerts is based on worst case scenario measurements. The performance categorizations are:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

- **Low:** Advice has minimal negative effect on performance. Advice might be run in test and production environments. Cumulative negative effect on performance is within run-to-run variance when all advice of this type is enabled.
- **Medium:** Advice has measurable but low negative effect on performance. Advice might be run within test environments, and might be run within production environments if deemed necessary. Cumulative negative effect on performance is less than 4% when all advice of this type is enabled.
- **High:** Advice impact is high or unknown. Advice might be run during problem determination tests and functional tests. It is not run in production simulation or production environments unless deemed necessary. Cumulative negative effect on performance might be significant when all advice of this type is enabled.

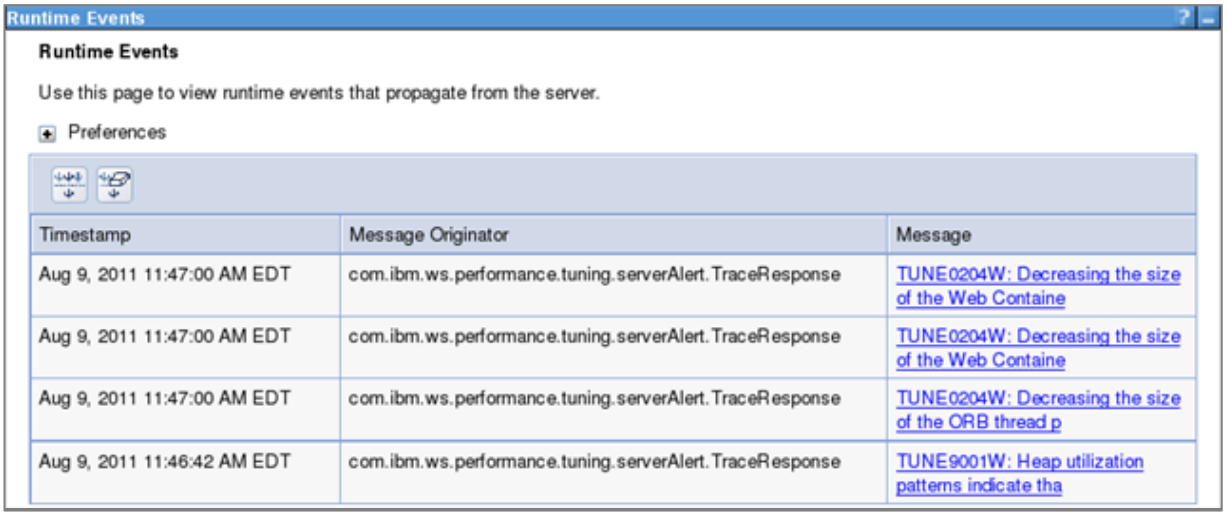
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance and Diagnostic Advisor (4 of 5)

- Tuning advice can be viewed in Runtime Events
- Click any TUNE message link for details



© Copyright IBM Corporation 2013

Figure 20-36. Performance and Diagnostic Advisor (4 of 5) WA855 / VA8551.0

Notes:

Tuning advice is provided as messages written to the runtime events. The TUNExxxx messages are typically at the Warning level.

Examples:

TUNE0201W: The rate of discards from the prepared statement cache is high. Increase the size of the prepared statement cache for the data source.

TUNE0207W: Utilization of the connection pool is high. Performance might be improved by increasing the maxPoolSize for data source {DS_name}. Try setting the minimum size to {value}, and the maximum size to {value}.

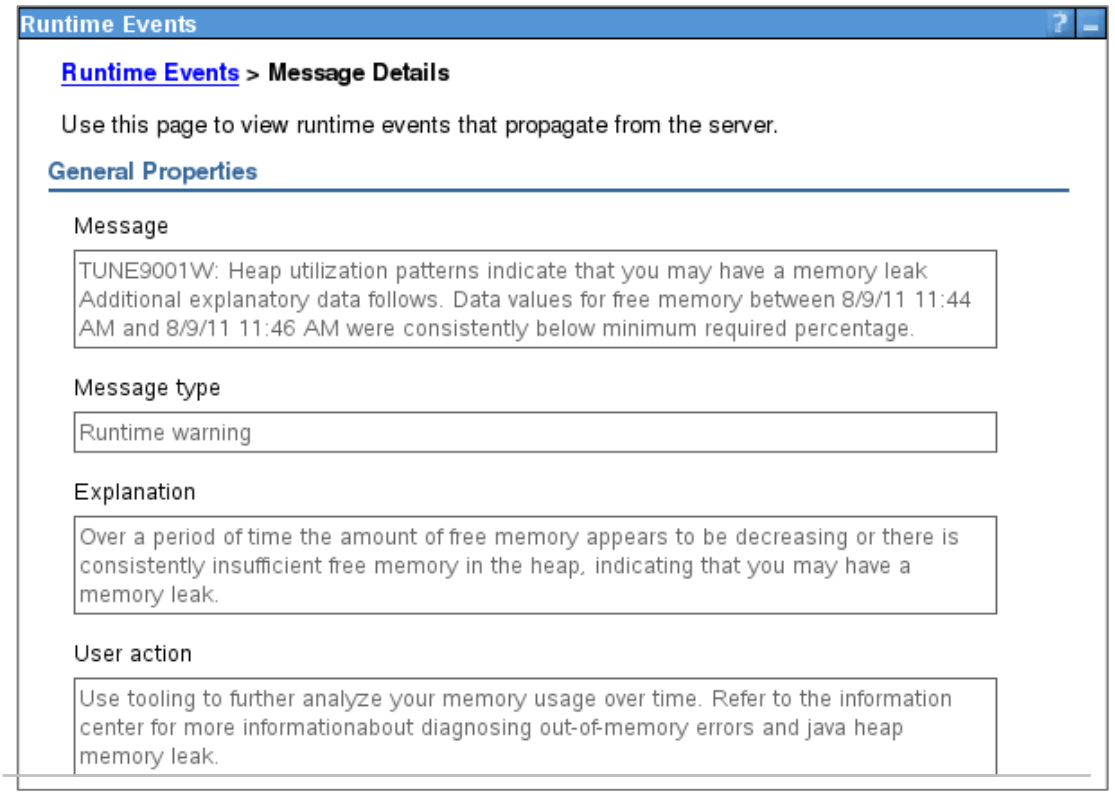
TUNE0220W: The Java virtual machine is spending a considerable amount of time in garbage collection. Consider increasing the heap size.

A complete list is available in the WebSphere Application Server V8 Information Center under **Reference > Messages > TUNE**.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance and Diagnostic Advisor (5 of 5)



© Copyright IBM Corporation 2013

Figure 20-37. Performance and Diagnostic Advisor (5 of 5) WA855 / VA8551.0

Notes:

The message is:

`TUNE9001W: Heap utilization patterns indicate that you might have a memory leak. Additional explanatory data follows. Data values for free memory between 8/9/11 11:44 AM and 8/9/11 11:46 AM were consistently less than the minimum required percentage.`

Explanation: Over time the amount of free memory seems to be decreasing or there is consistently insufficient free memory in the heap, indicating that you might have a memory leak.

User action: Use tools to further analyze your memory usage over time. For more information about diagnosing out-of-memory errors and Java heap memory leaks, see the information center.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Tivoli Performance Viewer advisor

- Performance advice:
  - ORB service thread pools
  - Web container thread pools
  - Connection pool size
  - Persisted session size and time
  - Prepared statement cache size
  - Session cache size
  - Dynamic cache size
  - Java virtual machine (JVM) heap size
  - DB2 Performance Configuration wizard

© Copyright IBM Corporation 2013

Figure 20-38. Tivoli Performance Viewer advisor WA855 / VA8551.0

Notes:

The performance advisor in Tivoli Performance Viewer provides advice on using collected Performance Monitoring Infrastructure (PMI) data to help tune systems for optimal performance and provide recommendations on inefficient settings. Obtain the advice by selecting the performance advisor in Tivoli Performance Viewer.

In a Network Deployment environment, the performance advisor in Tivoli Performance Viewer runs within the JVM of the node agent and can provide advice on resources that are more expensive to monitor and analyze. In a stand-alone application server environment, the performance advisor in Tivoli Performance Viewer runs within the application server JVM. The Tivoli Performance Viewer advisor requires that you enable performance modules, counters, or both.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Examples of performance advice

- Data sources
  - Situation: The prepared statement discard rate is too high, and heap space is available
  - Advice provided: Increase statement cache size
- Thread pools (ORB, web container, data source)
  - Situation: The number of connections is low (at the minimum)
  - Advice provided: Decrease pool size
  - Situation: All data source connections are heavily used, and heap space is available
  - Advice provided: Increase maximum pool size
  - Situation: The size of the pool is fluctuating a lot (high variance), possibly indicating batch processing, and wasted resources
  - Advice provided: Decrease pool size
- JVM heap size
  - Situation: Heap size is too small (less than 256 MB)
  - Advice provided: Increase the heap size to a value greater than 256 MB

© Copyright IBM Corporation 2013

Figure 20-39. Examples of performance advice WA855 / VA8551.0

Notes:

More examples of advice that the performance advisors would give in certain situations include:

Unbounded thread pools

- Situation: Threads added to an unbounded pool are not pooled
- Advice: If the average number of threads is higher than the pool size, then the pool must be increased to allow better pooling

Sessions

- Situation: Read/write time or size is too large
- Advice: Warn of application problem
- Situation: The number of live sessions is greater than the session cache, and memory is available
- Advice: Increase session cache

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

- Situation: Requests are turned down because there is no room for new sessions
- Advice: There are either too many active sessions, or the cache size is too small

DB2 Performance Configuration wizard

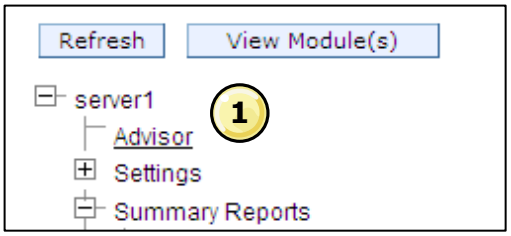
- Situation: A DB2 database is detected in the configuration
- Advice: Use the DB2 Performance wizard to tune the indicated database

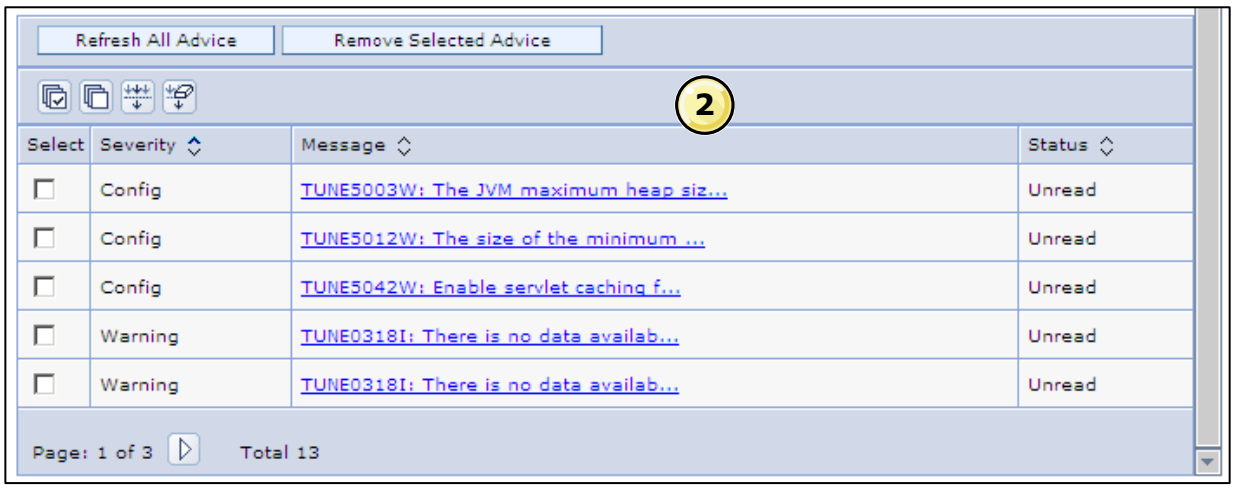
Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Viewing performance advice

- In Tivoli Performance Viewer, click the Advisor link
- From the list of messages, click a link to see more detail
  - Messages can be sorted by severity





© Copyright IBM Corporation 2013

Figure 20-40. Viewing performance advice WA855 / VA8551.0

Notes:

To view advice messages in Tivoli Performance Viewer, click the Advisor link.

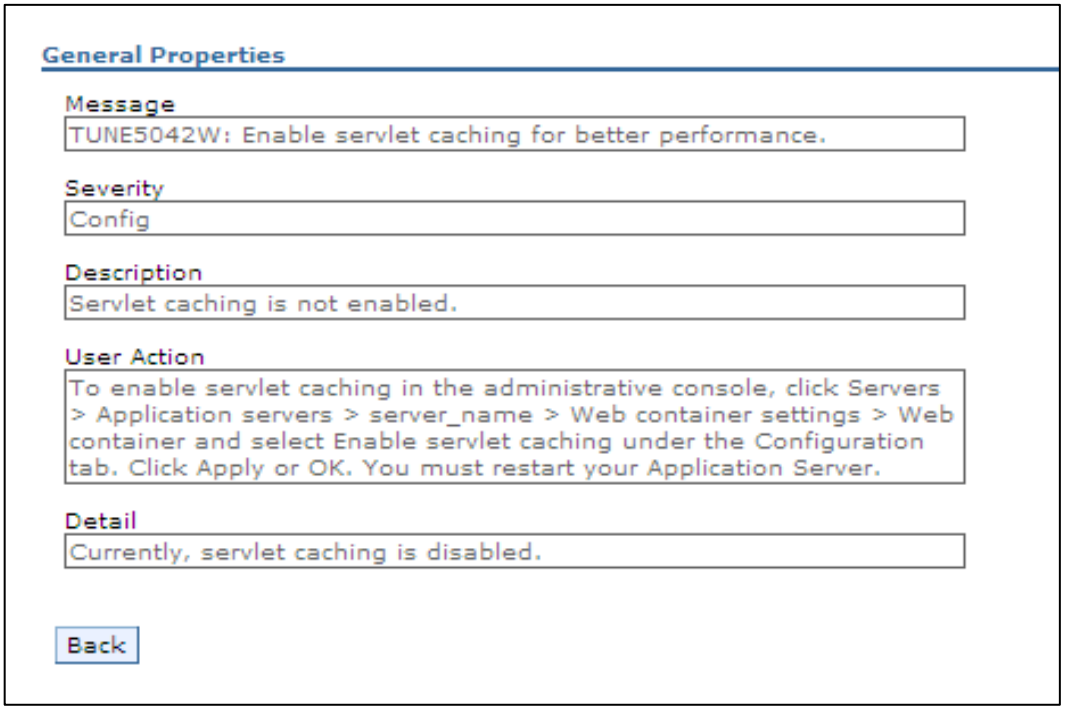
From the list of messages, click a link to see more detail.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Performance advice detail



© Copyright IBM Corporation 2013

Figure 20-41. Performance advice detail WA855 / VA8551.0

Notes:

In this example, the message suggests enabling servlet caching for better performance. Servlet caching is a web container setting that is disabled by default. The User Action section in the advice details provides instructions for enabling servlet caching.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Performance advisor suggested practices

- Use only during stable production load tests
  - Application must remain stable during production tests
  - Any exceptions and deadlock issues must be resolved before running
  - The test load must be consistent
  - Varied load might lead to contradictory advice
- Enable after production load tests reach peak load levels
  - Exclude ramp-up and ramp-down times from monitoring
  - Increasing or decreasing loads might lead to contradictory advice
  - Certain types of advice are only generated when processor is being stressed (processor use > 50%)
- Important: tune your application before you tune WebSphere

© Copyright IBM Corporation 2013

Figure 20-42. Performance advisor suggested practices WA855 / VA8551.0

Notes:

When using the performance advisors, processor utilization must rise above 50% before advice is generated. Typically when running your production level load, you push the processor usage to 80–100% before turning on one of the performance advisors.

Consider the following when using a performance advisor for tuning:

If the load changes on the system under test, contradictory advice is generated. This behavior is because the collected PMI data shows a different type of environment, causing the advice to shift. To avoid this situation, always run the advisors while simulating the load WebSphere experiences during deployment (peak load).

If the pool size minimum and maximum values are the same, the performance advisor rules are much more likely to give contradictory advice when load fluctuates.

The amount of processor usage determines the amount of system activity. The advisors do not consider disk activity, network activity, memory usage, or other factors to get a more realistic view of system load.

Recommendations are only generated when processor load reaches 50% and higher.

 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Performance advisors from different application servers might give contradictory advice on the same node resources. This behavior is because the application servers take into account *only* how they are individually employing the resource. In this situation, if the advice from the different advisors varies greatly, consider the generated advice and decide what changes to make. However, if all advisors are giving the same recommendations, then you must seriously consider the suggested changes.

If the performance advisor suggests setting a pool size to X, you must set the minimum value to X/2 and the maximum value to X.

If the performance advisor suggests setting the *prepared statement cache* value to a certain setting, check the amount of memory that is available before setting. The advisors do not take into account the amount of actual physical memory available on the system.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

20.4. IBM Tivoli Composite Application Manager for WebSphere Application Server

© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

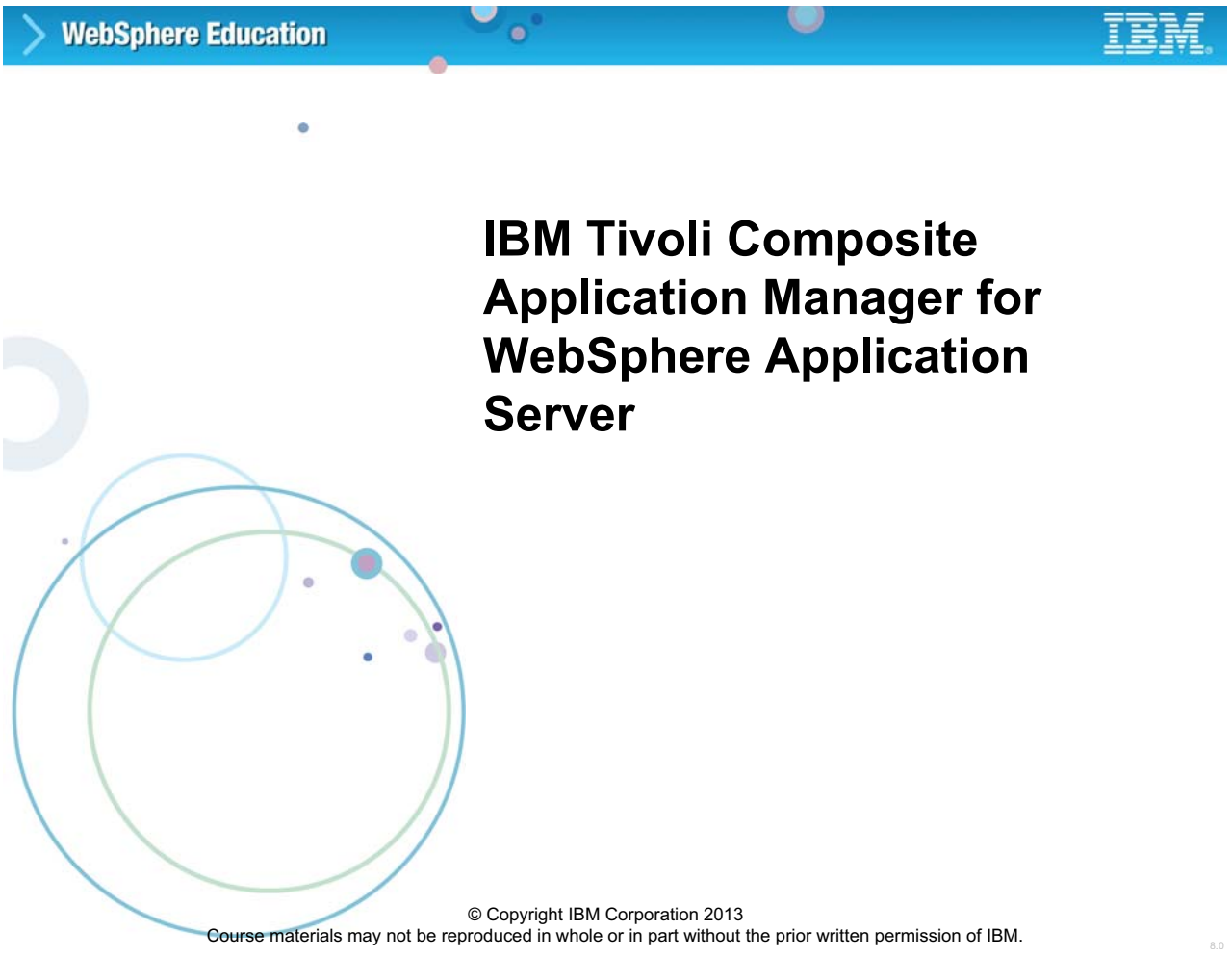


Figure 20-43. IBM Tivoli Composite Application Manager for WebSphere Application Server WA855 / VA8551.0

Notes:

© Copyright IBM Corp. 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

IBM Tivoli Composite Application Manager

- IBM Tivoli Composite Application Manager provides a suite of products for managing and monitoring applications
  - IBM Tivoli Composite Application Manager for CICS Transactions
  - IBM Tivoli Composite Application Manager for IMS Transactions
  - IBM Tivoli Composite Application Manager for J2EE
  - IBM Tivoli Composite Application Manager for Application Diagnostics
- With IBM Tivoli Composite Application Manager for Application Diagnostics, users can view the health of web applications and servers
  - Drill down to diagnostic information for specific application requests to identify the root cause of problems
- IBM Tivoli Composite Application Manager provides several data collectors for different servers, including
  - WebSphere Application Server
  - WebSphere Application Server Community Edition
  - WebSphere Process Server
  - WebSphere ESB Server
  - WebSphere Portal Server

© Copyright IBM Corporation 2013

Figure 20-44. IBM Tivoli Composite Application Manager WA855 / VA8551.0

Notes:

IBM Tivoli Composite Application Manager for Application Diagnostics gives users the ability to view the health of web applications and servers; then drill down to diagnostic information for specific application requests to identify the root cause of problems. Some of the features of IBM Tivoli Composite Application Manager for Application Diagnostics include:

- Analyze application performance through trending or historical analysis
- Provides key performance metrics to IBM Tivoli Composite Application Manager for Transactions to sense and isolate potential problems
- View all Java EE transactions that are “in-flight” to uncover the root cause of bottlenecks and do detailed memory analysis
- Correlates and profiles transactions that span multiple subsystems such as IBM CICS and IMS
- Software consistency checker compares key system and JVM metrics on working and non-working systems to help isolate differences that might be causing problems

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

- Exchange information with IBM Rational Performance Tester so developers understand the performance of applications in test or production

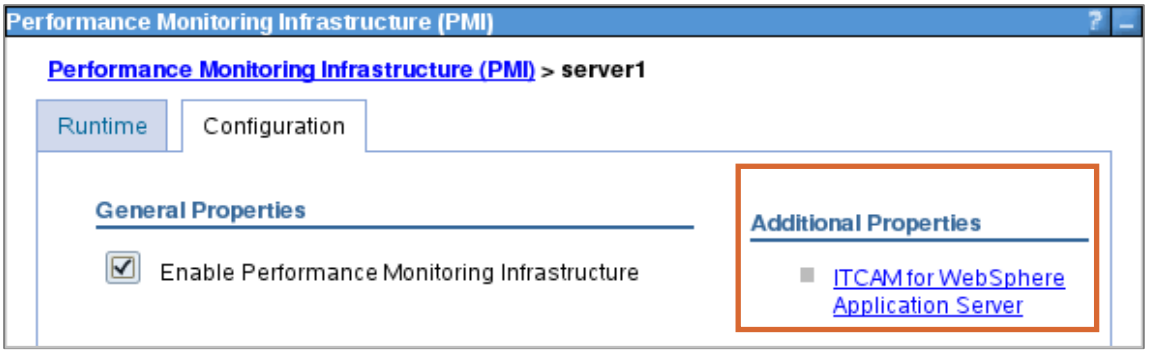
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

IBM Tivoli Composite Application Manager for WebSphere Application Server

- Data collector available in WebSphere Application Server V8.5 as an extension offering (optional download and installation)
- IBM Tivoli Composite Application Manager for WebSphere Application Server is a separate installation
  - Installed by using the IBM Installation Manager
  - Configure one or more servers for data collection
- The IBM Tivoli Composite Application Manager for WebSphere Application Server link shows up on the PMI configuration page



© Copyright IBM Corporation 2013

Figure 20-45. IBM Tivoli Composite Application Manager for WebSphere Application Server WA855 / VA8551.0

Notes:

IBM Tivoli Composite Application Manager is enhanced in version 8.0 and can be installed together with the application server. This integrated monitoring tool allows you to view the health of web applications and servers, and drill down to diagnostic information for specific application requests to identify the root cause of problems.

IBM Tivoli Composite Application Manager for WebSphere Application Server can be configured per server by selecting **Monitoring and Tuning > Performance Monitoring Infrastructure > server_name**.

The server must be configured with the IBM Tivoli Composite Application Manager GUI before you can see the IBM Tivoli Composite Application Manager for WebSphere Application Server link under Additional Properties on the PMI configuration tab of the server.

Use this page to enable or disable the IBM Tivoli Composite Application Manager for WebSphere Application Server Data Collector. Changes take effect after the server is restarted.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

It is a simple upgrade from IBM Tivoli Composite Application Manager for WebSphere Application Server to IBM Tivoli Composite Application Manager for Application Diagnostics (no “rip and replace”). After the upgrade, IBM Tivoli Composite Application Manager data is still visible in Tivoli Performance Viewer as well.

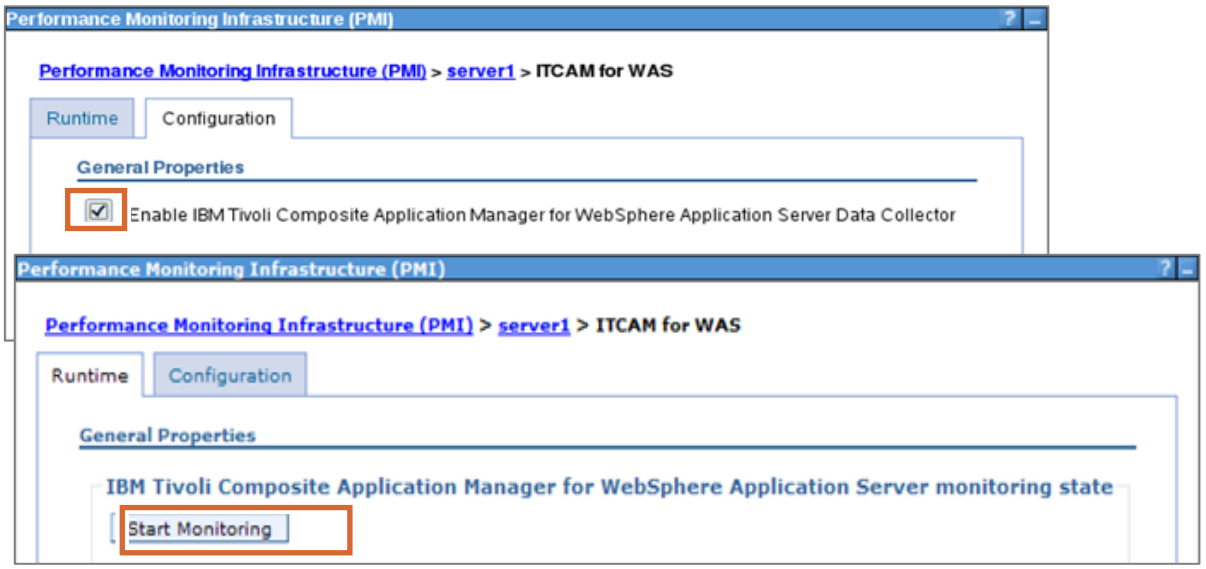
© Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

WebSphere Education

IBM Tivoli Composite Application Manager for WebSphere Application Server

- Select **Enable IBM Tivoli Composite Application Manager for WebSphere Application Server Data Collector** on the **Configuration** tab
- On the Runtime tab, click **Start Monitoring**



© Copyright IBM Corporation 2013

Figure 20-46. IBM Tivoli Composite Application Manager for WebSphere Application Server WA855 / VA8551.0

Notes:

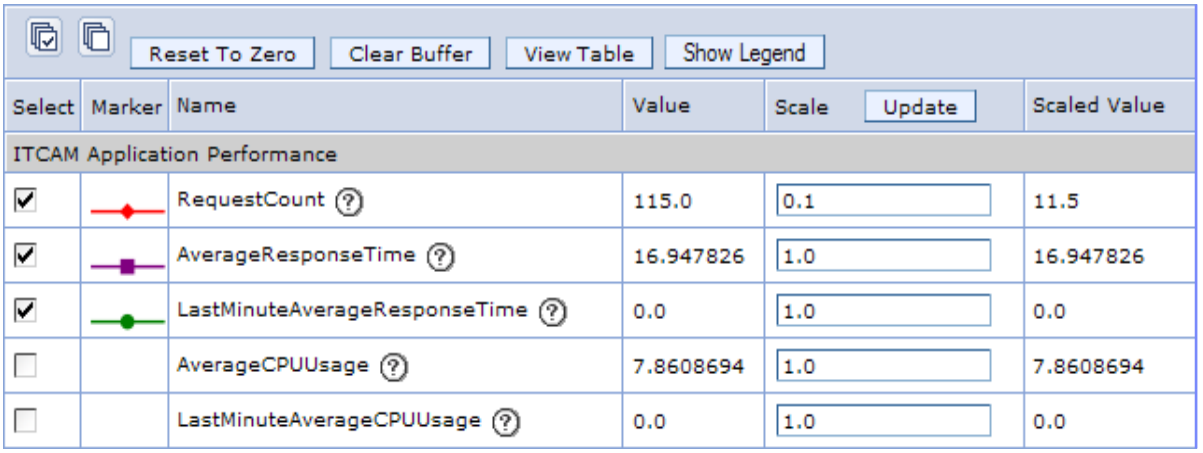
To see the IBM Tivoli Composite Application Manager modules and metrics in the Tivoli Performance Viewer, you are required to enable it by selecting the check box on the **Configuration** tab. The next step is to click **Start Monitoring** on the runtime tab.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

IBM Tivoli Composite Application Manager metrics in Tivoli Performance Viewer

- View metrics in Tivoli Performance Viewer
  - Select the **ITCAM Application Performance** module
  - Select the application



© Copyright IBM Corporation 2013

Figure 20-47. IBM Tivoli Composite Application Manager metrics in Tivoli Performance Viewer WA855 / VA8551.0

Notes:

IBM Tivoli Composite Application Manager for WebSphere Application Server provides more request-based response time and processor metrics.

Customer application code is not instrumented in any way.

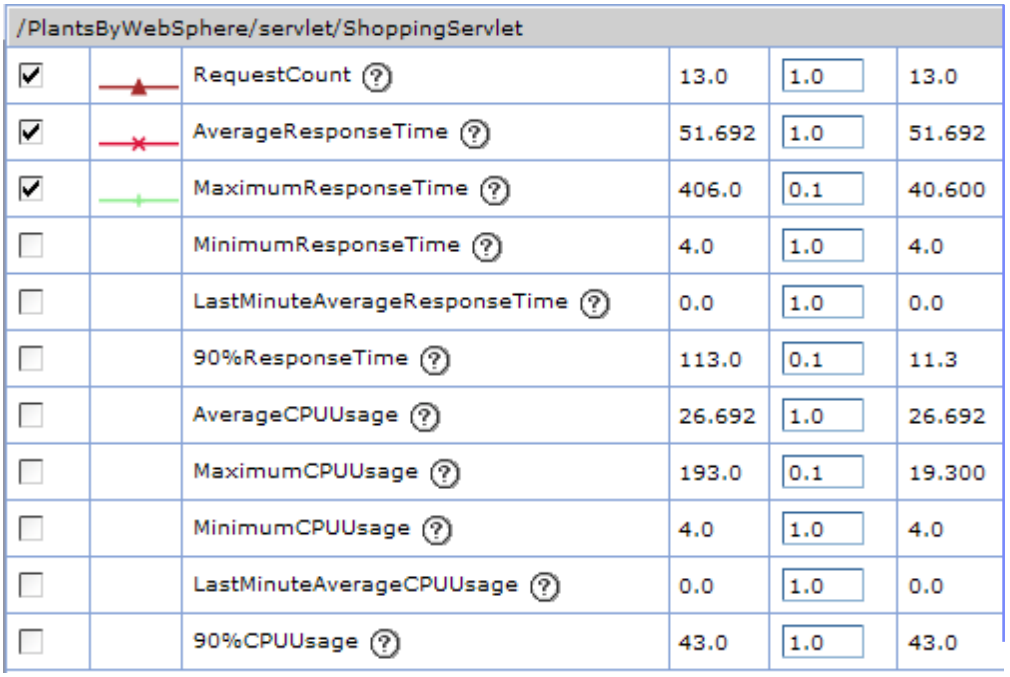
 © Copyright IBM Corp. 2013

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

IBM Tivoli Composite Application Manager application metrics in Tivoli Performance Viewer

- Additional metrics for the ShoppingServlet



/PlantsByWebSphere/servlet/ShoppingServlet					
☑		RequestCount ?	13.0	1.0	13.0
☑		AverageResponseTime ?	51.692	1.0	51.692
☑		MaximumResponseTime ?	406.0	0.1	40.600
☐		MinimumResponseTime ?	4.0	1.0	4.0
☐		LastMinuteAverageResponseTime ?	0.0	1.0	0.0
☐		90%ResponseTime ?	113.0	0.1	11.3
☐		AverageCPUUsage ?	26.692	1.0	26.692
☐		MaximumCPUUsage ?	193.0	0.1	19.300
☐		MinimumCPUUsage ?	4.0	1.0	4.0
☐		LastMinuteAverageCPUUsage ?	0.0	1.0	0.0
☐		90%CPUUsage ?	43.0	1.0	43.0

© Copyright IBM Corporation 2013

Figure 20-48. IBM Tivoli Composite Application Manager application metrics in Tivoli Performance Viewer WA855 / VA8551.0

Notes:

Several performance metrics are collected and displayed for each component of an application. This screen capture shows the metrics for the PlantsByWebSphere shopping servlet after a load test. Clicking or hovering over the question mark (?) for each metric displays a description of the metric.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Describe performance monitoring and tuning methods
- Use the Tivoli Performance Viewer to monitor application server resources
- Use the performance servlet to generate performance data
- Configure the Request Metrics tool to generate performance data about the end-to-end request flow
- Use Performance Advisors to generate suggested tuning actions
- Enable the performance collectors from IBM Tivoli Composite Application Manager for WebSphere Application Server

© Copyright IBM Corporation 2013

Figure 20-49. Unit summary WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education

Checkpoint questions

1. What are the two performance data collection technologies in WebSphere?

2. Which WebSphere performance tool allows you to monitor overall system health?

3. True or False: The Performance Monitoring Infrastructure is enabled by default.

4. True or False: The Tivoli Performance Viewer Advisor tool generates tuning advice and automatically applies it to the environment.

© Copyright IBM Corporation 2013

Figure 20-50. Checkpoint questions WA855 / VA8551.0

Notes:

Write your answers here:

1.

2.

3.

4.

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

WebSphere Education IBM

Checkpoint answers

1. What are the two performance data collection technologies in WebSphere?
   - The Performance Monitoring Infrastructure (PMI) and request metrics provide the data collection mechanisms in WebSphere.
2. Which WebSphere performance tool allows you to monitor overall system health?
   - The Tivoli Performance Viewer allows you to monitor overall system health.
3. True or False: The Performance Monitoring Infrastructure is enabled by default.
   - True. PMI is enabled by default.
4. True or False: The Tivoli Performance Viewer Advisor tool generates tuning advice and automatically applies it to the environment.
   - False. The performance advisor tools do not automatically tune the environment. You must tune manually and test the effect of the changes.

© Copyright IBM Corporation 2013

Figure 20-51. Checkpoint answers WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Exercise 16

Using the performance monitoring tools

© Copyright IBM Corporation 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

8.0

Figure 20-52. Exercise 16 WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Exercise objectives

After completing this exercise, you should be able to:

- Enable various levels of Performance Monitoring Infrastructure (PMI) statistics for an application server
- Monitor an application server by using Tivoli Performance Viewer
- Configure user settings for Tivoli Performance Viewer
- Examine summary reports and performance modules in Tivoli Performance Viewer
- View performance messages from the Tivoli Performance Viewer Advisor
- Enable and configure the Request Metrics tool
- View Request Metrics messages in the standard logs of an application server
- Configure IBM Tivoli Composite Application Manager for WebSphere Application Server collector for an application server
- View IBM Tivoli Composite Application Manager application performance statistics by using Tivoli Performance Viewer

© Copyright IBM Corporation 2013

Figure 20-53. Exercise objectives WA855 / VA8551.0

Notes:

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.