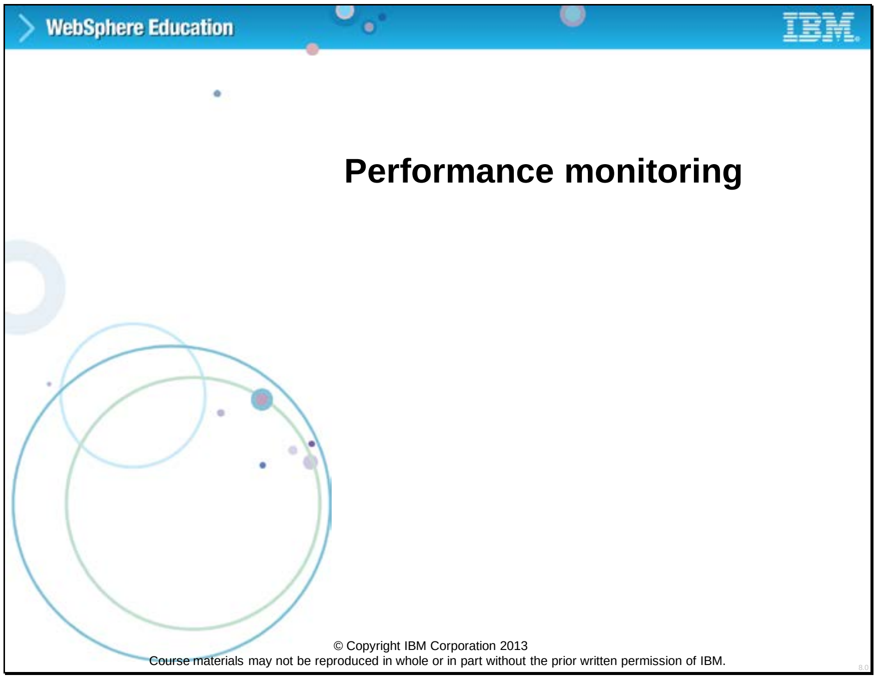
Slide 1

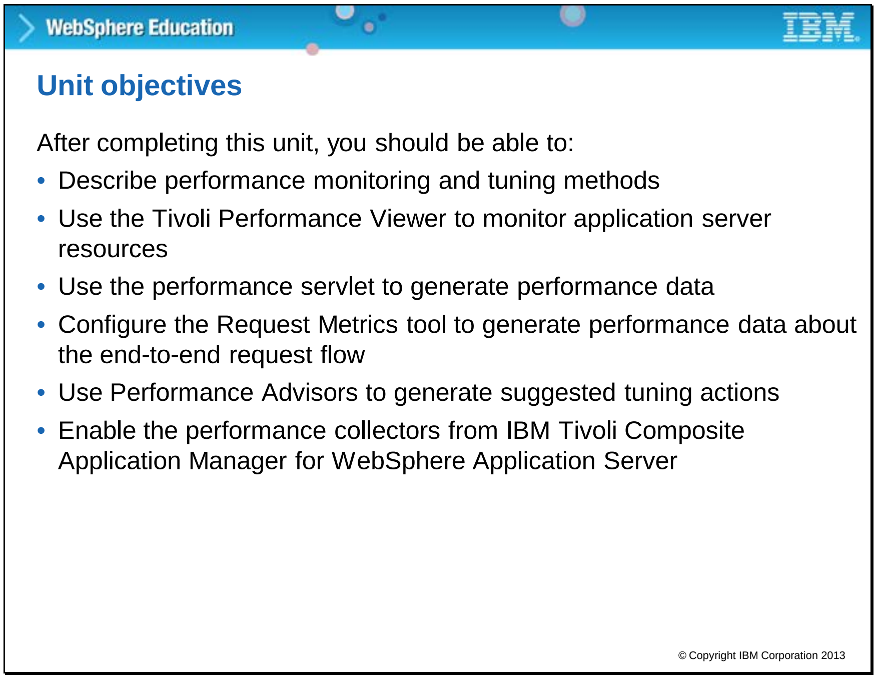


Performance monitoring

Unit 20: Performance monitoring

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

Slide 2



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

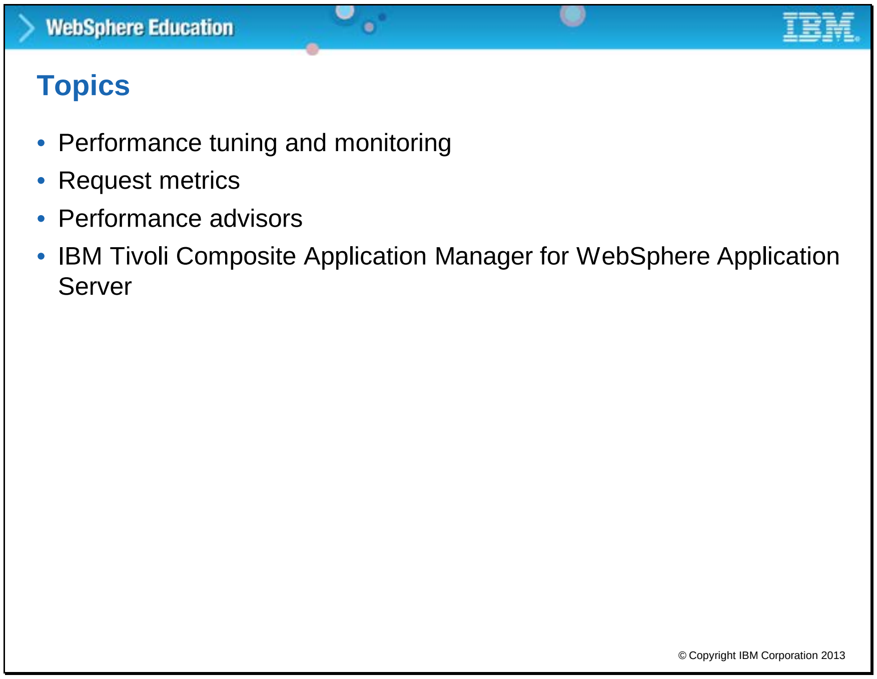
- Describe performance monitoring and tuning methods
- Use the Tivoli Performance Viewer to monitor application server resources
- Use the performance servlet to generate performance data
- Configure the Request Metrics tool to generate performance data about the end-to-end request flow
- Use Performance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

Title: 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 tuning actions
- Enable the performance collectors from IBM Tivoli Composite Application Manager for WebSphere Application Server

Slide 3



WebSphere Education

IBM

Topics

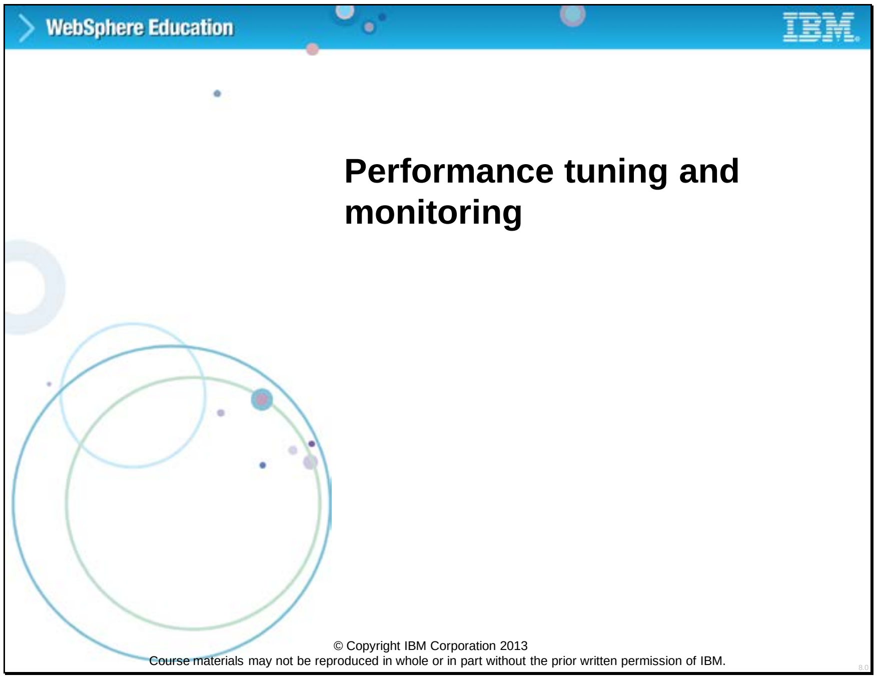
- Performance tuning and monitoring
- Request metrics
- Performance advisors
- IBM Tivoli Composite Application Manager for WebSphere Application Server

Title: Topics

This unit describes the following topics.

- Performance tuning and monitoring
- Request metrics
- Performance advisors
- ITCAM for WebSphere Application Server

Slide 4



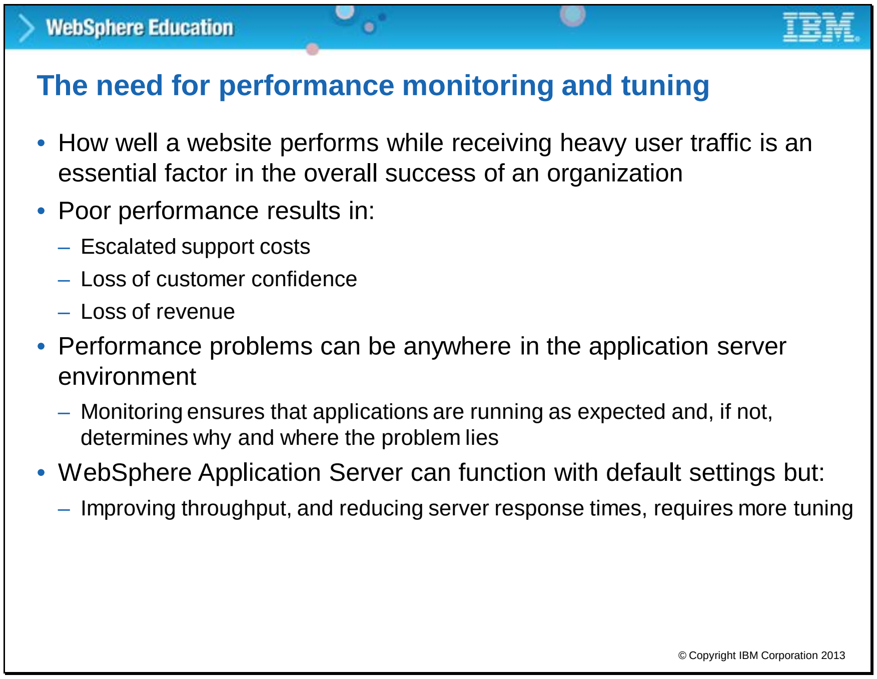
WebSphere Education

Performance tuning and monitoring

Topic: Performance tuning and monitoring

This topic covers some methods for tuning and monitoring application server performance.

Slide 5



WebSphere Education

IBM

The need for performance monitoring and tuning

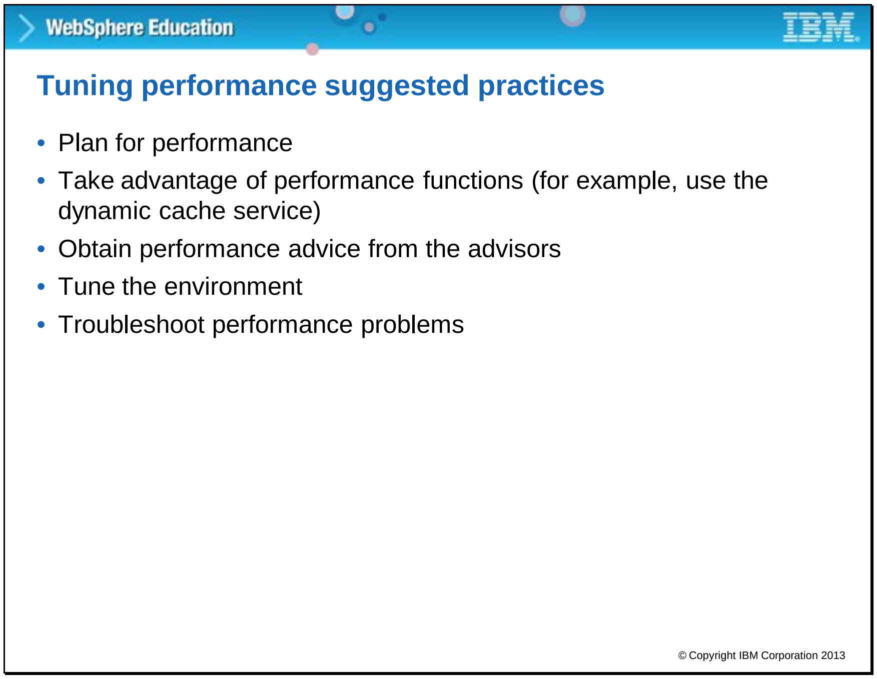
- How well a website performs while receiving heavy user traffic is an essential factor in the overall success of an organization
- Poor performance results in:
  - Escalated support costs
  - Loss of customer confidence
  - Loss of revenue
- Performance problems can be anywhere in the application server 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

Title: The need for performance monitoring and tuning

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. It is important to monitor system performance because poor performance can result in higher support costs, loss of customer confidence, and loss of revenue. Monitoring ensures that applications are running as expected, and if not, the cause of performance problems can be investigated. WebSphere Application Server can function well with default settings, but some applications can require further tuning for optimal performance.

Slide 6



WebSphere Education

Tuning performance suggested practices

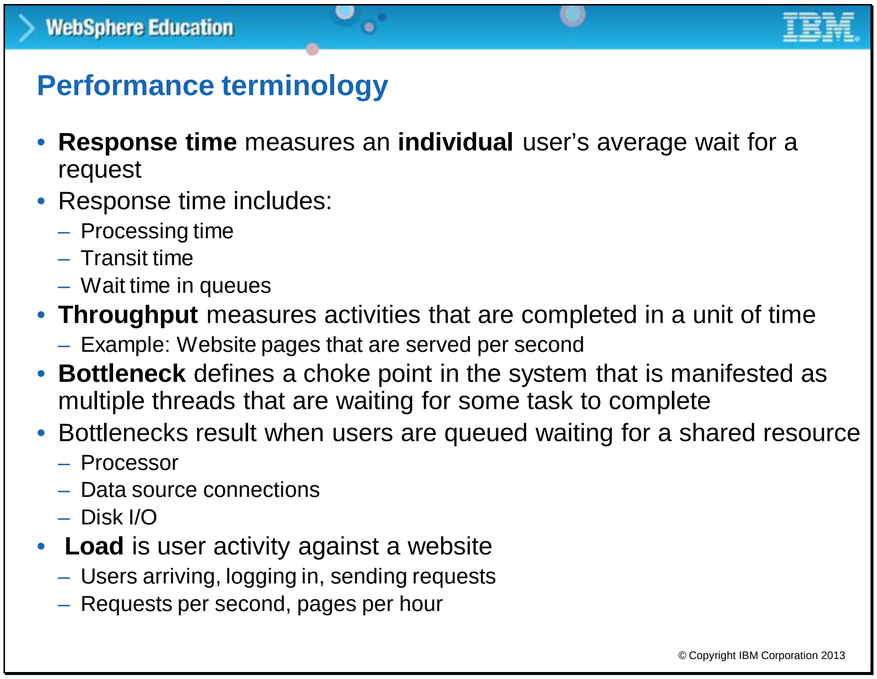
- Plan for performance
- Take advantage of performance functions (for example, use the dynamic cache service)
- Obtain performance advice from the advisors
- Tune the environment
- Troubleshoot performance problems

Title: Tuning performance best practices

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.

Slide 7



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

Title: Performance terminology

This slide displays a list of terms that are commonly associated with performance. Creating a glossary of performance terms is essential for good performance monitoring. The glossary is used to ensure that everyone who is involved understands the meaning of each term, decreasing the chance of inaccurate performance statistics.

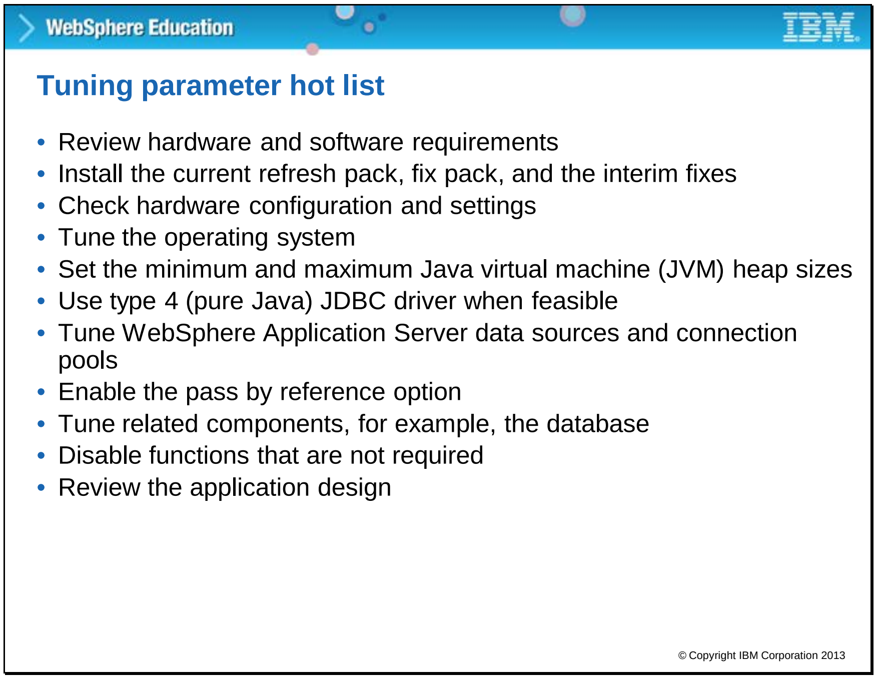
Response time measures average wait for a request for an individual user. Response times include the cumulative time that is spent in processing the request, time that is spent in transit between systems, and the wait time that is spent in queues.

Throughput measures activities that are completed in a unit of time. For example, the number of 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. A bottleneck results when user requests are queued waiting for a shared resource such as processors, data source connections, and disk I/O.

Load is user activity against a website. Some examples are the number of users that arrive, log in, and send requests.

Slide 8



WebSphere Education
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

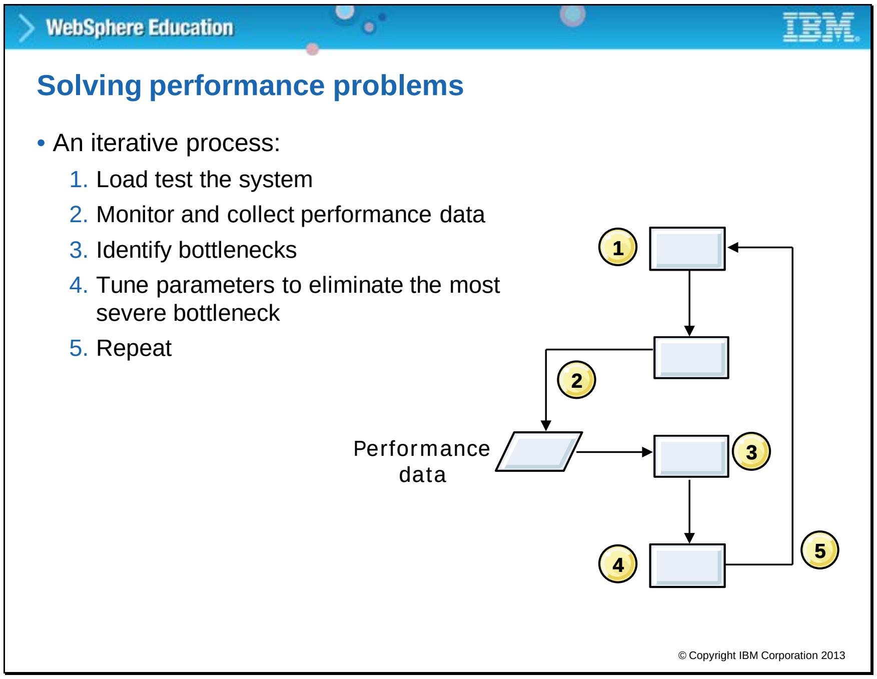
Title: Tuning parameter hot list

This hot list contains recommendations that can improve performance, 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. The details of each of these items are documented in the information center.

The last item is especially important. Review your application design. You can track many performance problems back to the application design. Review the design to determine whether it is the source of performance problems.

Slide 9



Solving performance problems

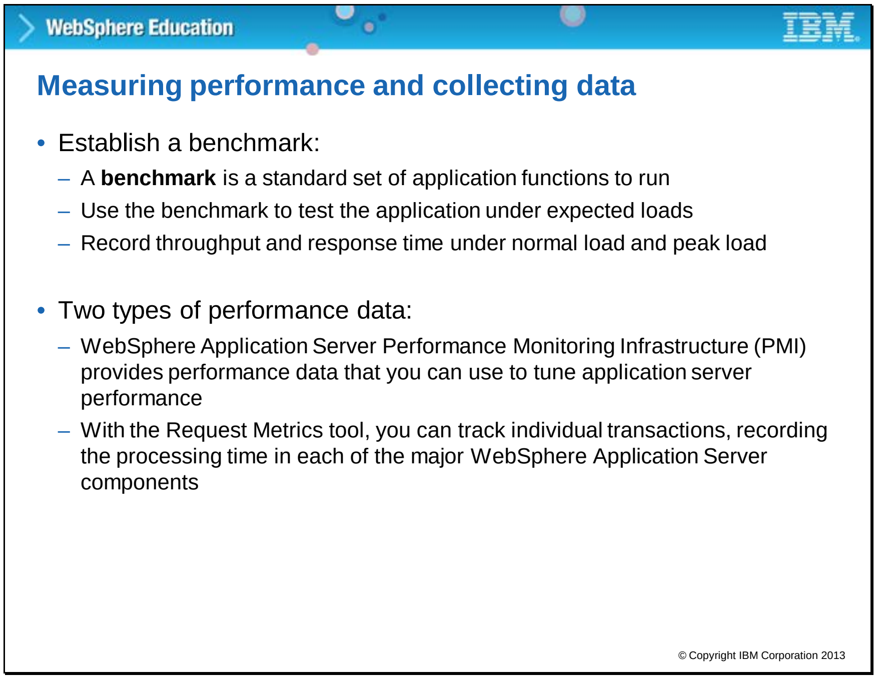
- An iterative process:
  1. Load test the system
  2. Monitor and collect performance data
  3. Identify bottlenecks
  4. Tune parameters to eliminate the most severe bottleneck
  5. Repeat

Title: Solving a performance problem

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.

Slide 10



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

Title: Measuring performance and collecting data

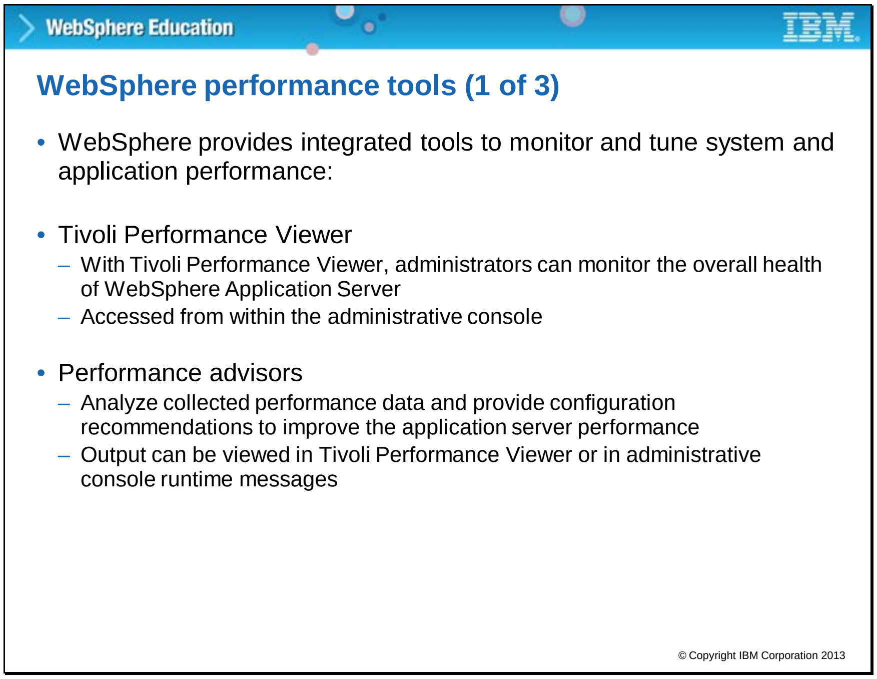
To measure performance, begin by choosing a benchmark. A benchmark is a standard set of operations to run. This benchmark exercises those application functions experiencing performance problems. Complex systems frequently need a warm-up period to cache objects and optimize code paths. System performance 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 several workloads.

Several tools facilitate benchmark development. The tools range from tools that merely call 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.

Slide 11



WebSphere Education
IBM

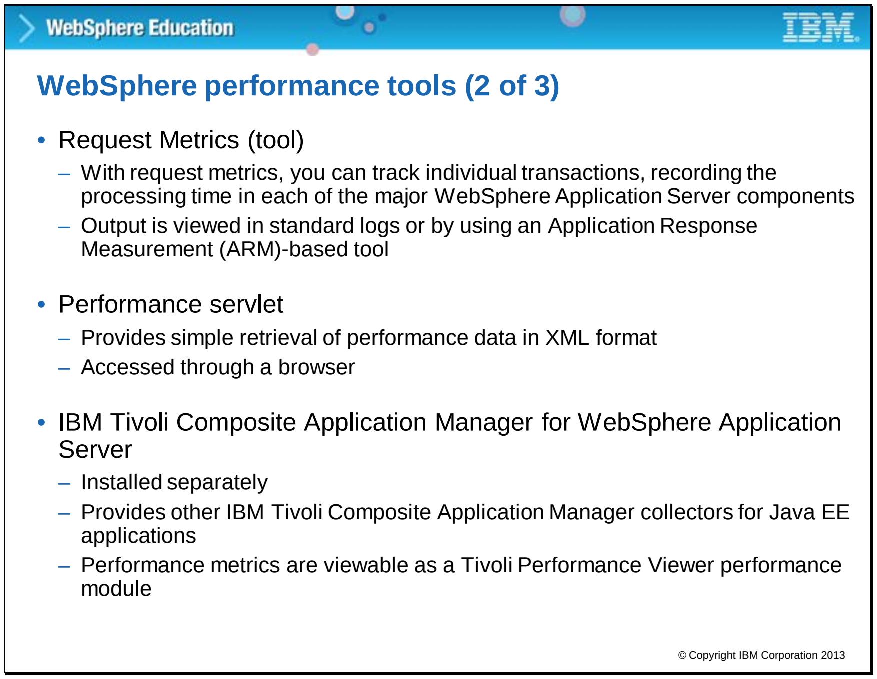
WebSphere performance tools (1 of 3)

- 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

Title: WebSphere performance tools (1 of 3)

WebSphere Application Server provides the following tools, or facilities, to monitor and tune system performance: The Tivoli Performance Viewer or TPV, request metrics, performance advisors, and the performance servlet, all of which are described in more detail in this unit.

Slide 12



WebSphere Education

IBM

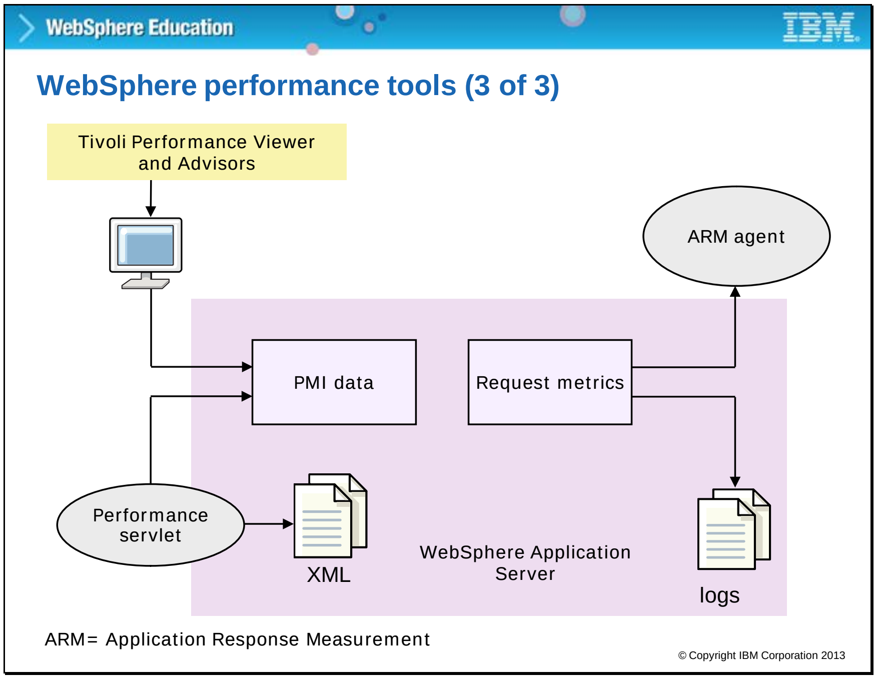
WebSphere performance tools (2 of 3)

- 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

Title: WebSphere performance tools (2 of 3)

PMI data can be viewed by using tools that are built into the administrative console, or can be viewed in XML form by using the performance servlet. Request metrics are written to the logs, and can also be viewed by using an ARM agent. IBM Tivoli Composite Application Manager (ITCAM) for WebSphere is introduced in version 7. ITCAM for WebSphere gives you some additional monitoring capabilities such as CPU usage per application.

Slide 13



WebSphere performance tools (3 of 3)

Tivoli Performance Viewer and Advisors
ARM agent
PMI data
Request metrics
Performance servlet
XML
WebSphere Application Server
logs

ARM= Application Response Measurement

© Copyright IBM Corporation 2013

Title: WebSphere performance tools (3 of 3)

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

The **Tivoli Performance Viewer,** which is accessed from within the administrative console, gives administrators the ability to monitor the overall health of WebSphere Application Server.

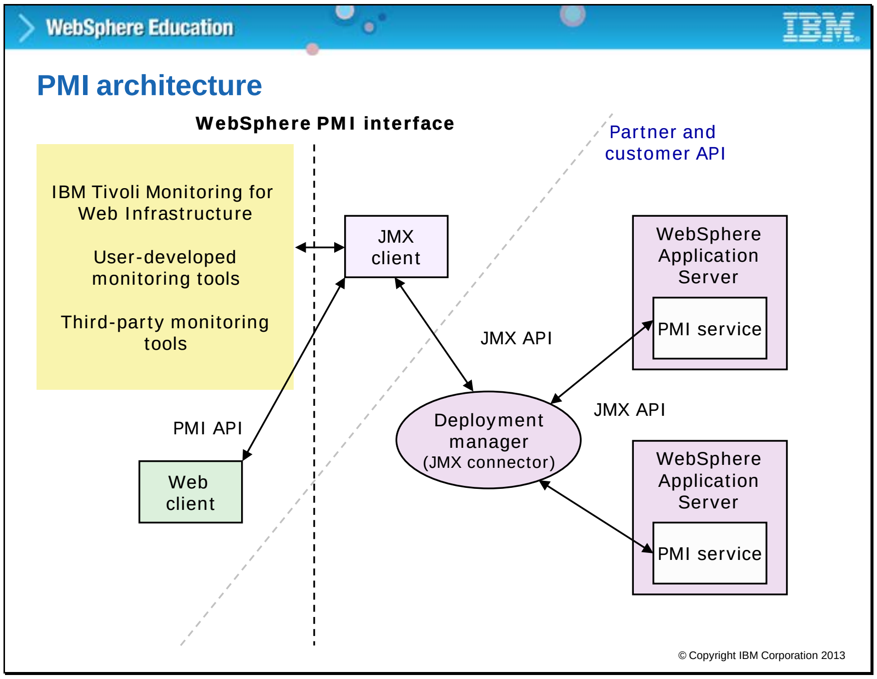
The **Request metric (tool) gives** you the ability to track individual transactions, and record the processing time in each of the major WebSphere Application Server components.

Output is viewed in standard logs or by using an Application Response Measurement (ARM)-based tool.

Performance advisors 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.

The **Performance servlet** provides simple retrieval of performance data in XML format, which is accessed through a browser.

Slide 14



WebSphere Education
IBM
PMI architecture
WebSphere PMI interface
Partner and customer API
IBM Tivoli Monitoring for Web Infrastructure
User-developed monitoring tools
Third-party monitoring tools
JMX client
WebSphere Application Server
PMI service
JMX API
JMX API
PMI API
Deployment manager (JMX connector)
Web client
WebSphere Application Server
PMI service
© Copyright IBM Corporation 2013

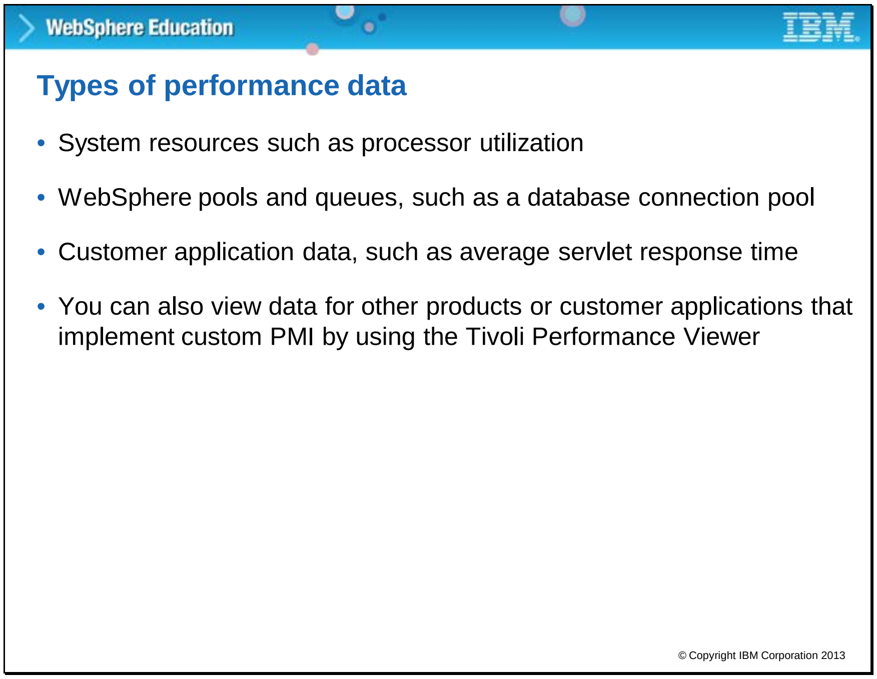
Title: PMI architecture

The Performance Monitoring Infrastructure (PMI) uses 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, which 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 EE Management Reference Implementation (JSR-77).

Slide 15



WebSphere Education
IBM

Types of performance data

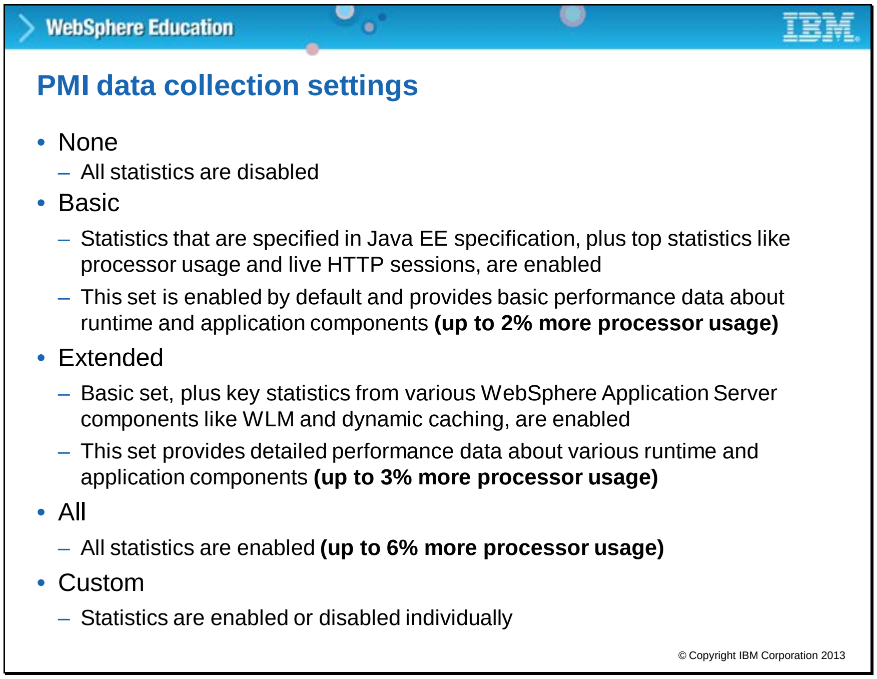
- System resources such as processor utilization
- WebSphere pools and queues, such as a database connection pool
- Customer application data, such as average servlet response time
- You can also view data for other products or customer applications that implement custom PMI by using the Tivoli Performance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

© Copyright IBM Corporation 2013

Title: Types of performance data

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.

Slide 16



WebSphere Education
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

Title: PMI data collection settings

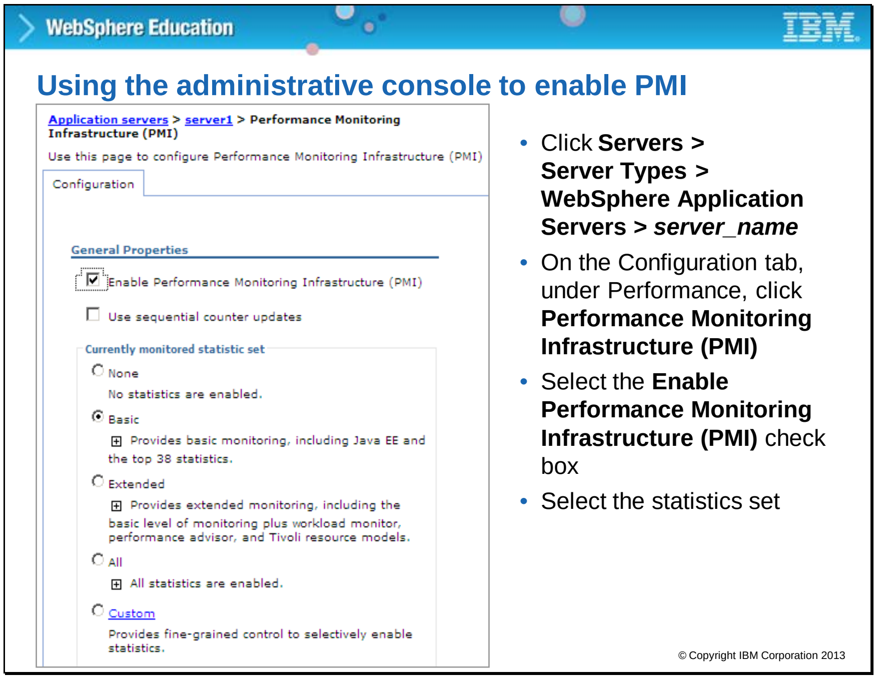
PMI uses statistics sets to specify the type and amount of performance data to collect.

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.

PMI is set to monitor at a Basic level by default.

Slide 17



Using the administrative console to enable PMI

Application servers > server1 > Performance Monitoring Infrastructure (PMI)

Use this page to configure Performance Monitoring Infrastructure (PMI)

Configuration

General Properties

☑ Enable Performance Monitoring Infrastructure (PMI)

□ Use sequential counter updates

Currently monitored statistic set

○ None
No statistics are enabled.

● Basic
⊞ Provides basic monitoring, including Java EE and the top 38 statistics.

○ Extended
⊞ Provides extended monitoring, including the basic level of monitoring plus workload monitor, performance advisor, and Tivoli resource models.

○ All
⊞ All statistics are enabled.

○ Custom
Provides fine-grained control to selectively enable statistics.

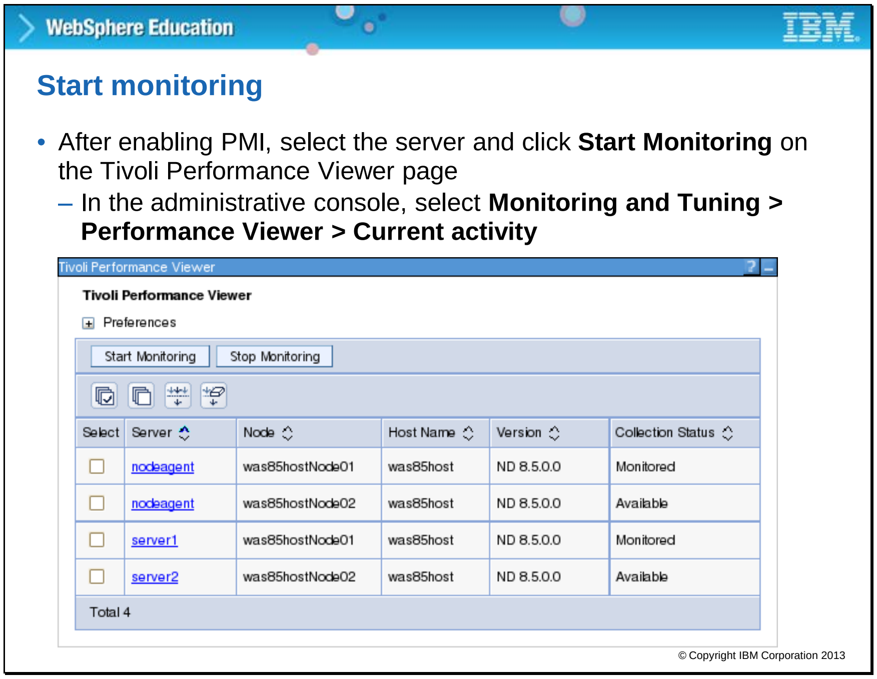
- Click **Servers > Server Types > WebSphere Application Servers >** ***server_name***
- On the Configuration tab, under Performance, click **Performance Monitoring Infrastructure (PMI)**
- Select the **Enable Performance Monitoring Infrastructure (PMI)** check box
- Select the statistics set

Title: Using the administrative console to enable PMI

For a particular server, navigate to its **Performance Monitoring** Infrastructure configuration panel.

Select the **Enable Performance Monitoring Infrastructure (PMI)** check box and select a statistics set.

Slide 18



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**

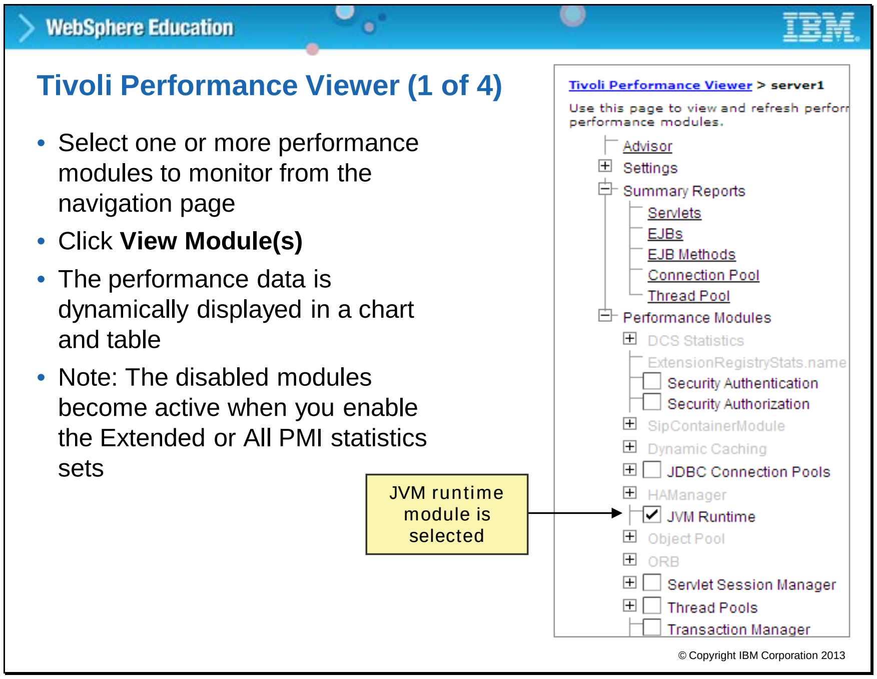
Tivoli Performance Viewer

Select	Server	Node	Host Name	Version	Collection Status
□	nodeagent	was85hostNode01	was85host	ND 8.5.0.0	Monitored
□	nodeagent	was85hostNode02	was85host	ND 8.5.0.0	Available
□	server1	was85hostNode01	was85host	ND 8.5.0.0	Monitored
□	server2	was85hostNode02	was85host	ND 8.5.0.0	Available
Total 4					

Title: Start monitoring

After enabling PMI for the server, in the console, select **Monitoring and Tuning > Tivoli performance Viewer > *server_name* > Start Monitoring.** Then, go back to **Monitoring and Tuning > Tivoli performance Viewer > current activity**.

Slide 19



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

JVM runtime module is selected

Tivoli Performance Viewer > **server1**

Use this page to view and refresh perforr performance modules.

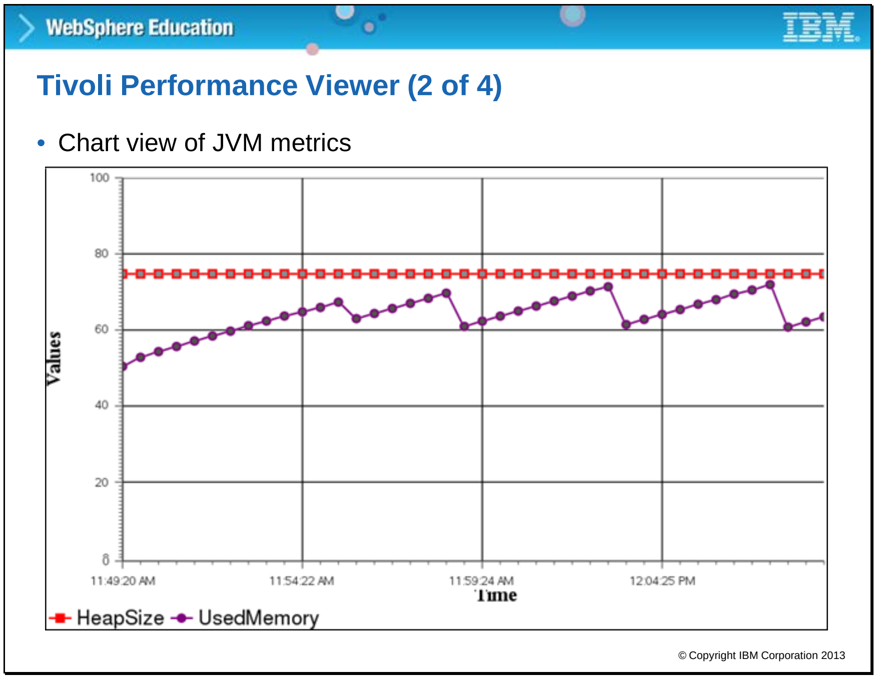
- Advisor
- Settings
- Summary Reports
  - Servlets
  - EJBs
  - EJB Methods
  - Connection Pool
  - Thread Pool
- Performance Modules
  - DCS Statistics
  - ExtensionRegistryStats.name
  - □ Security Authentication
  - □ Security Authorization
  - SipContainerModule
  - Dynamic Caching
  - □ JDBC Connection Pools
  - HAManager
  - ☑ JVM Runtime
  - Object Pool
  - ORB
  - □ Servlet Session Manager
  - □ Thread Pools
  - □ Transaction Manager

© Copyright IBM Corporation 2013

Title: Tivoli Performance Viewer (1 of 4)

In the Tivoli Performance Viewer, you can view graphs and reports that display PMI statistics. This area of the screen shows the navigation tree in Tivoli Performance Viewer where you can select which components to monitor. In this example, the JVM runtime module is selected.

Slide 20



WebSphere Education

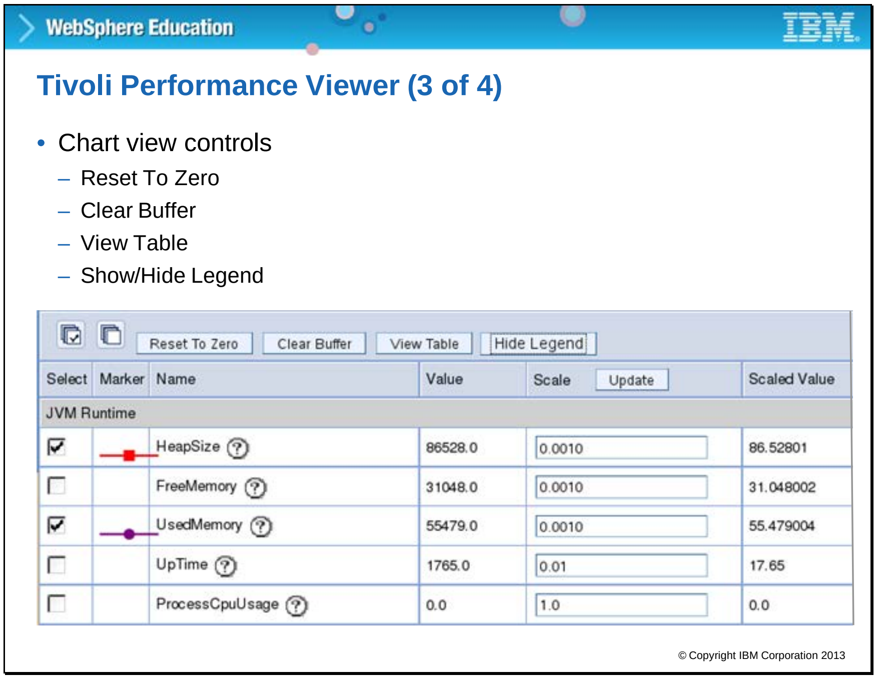
Tivoli Performance Viewer (2 of 4)

- Chart view of JVM metrics

Title: Tivoli Performance Viewer (2 of 4)

Statistics for the selected modules are displayed as a line graph. You can select which metrics you want 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 run time. The saw tooth 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.

Slide 21



Tivoli Performance Viewer (3 of 4)

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

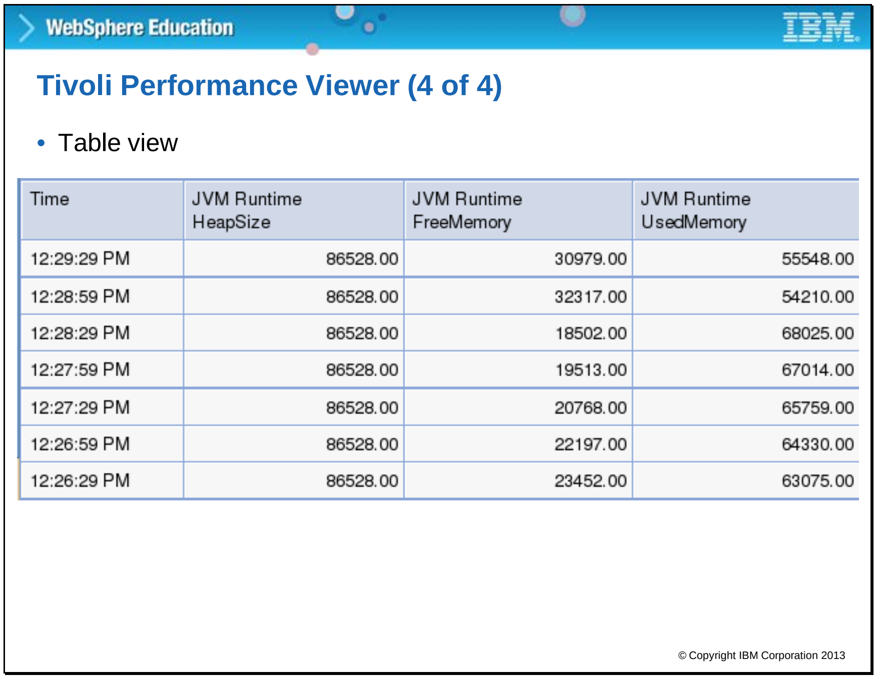
Select	Marker	Name	Value	Scale	Scaled Value
JVM Runtime					
☑	—■—	HeapSize ?	86528.0	0.0010	86.52801
☐		FreeMemory ?	31048.0	0.0010	31.048002
☑	—●—	UsedMemory ?	55479.0	0.0010	55.479004
☐		UpTime ?	1765.0	0.01	17.65
☐		ProcessCpuUsage ?	0.0	1.0	0.0

Title: Tivoli Performance Viewer (3 of 4)

The **Reset to zero** button sets a new baseline by using the current counter readings when 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 the **Clear Buffer** button to remove the PMI data from a table or chart.

Slide 22



WebSphere Education

Tivoli Performance Viewer (4 of 4)

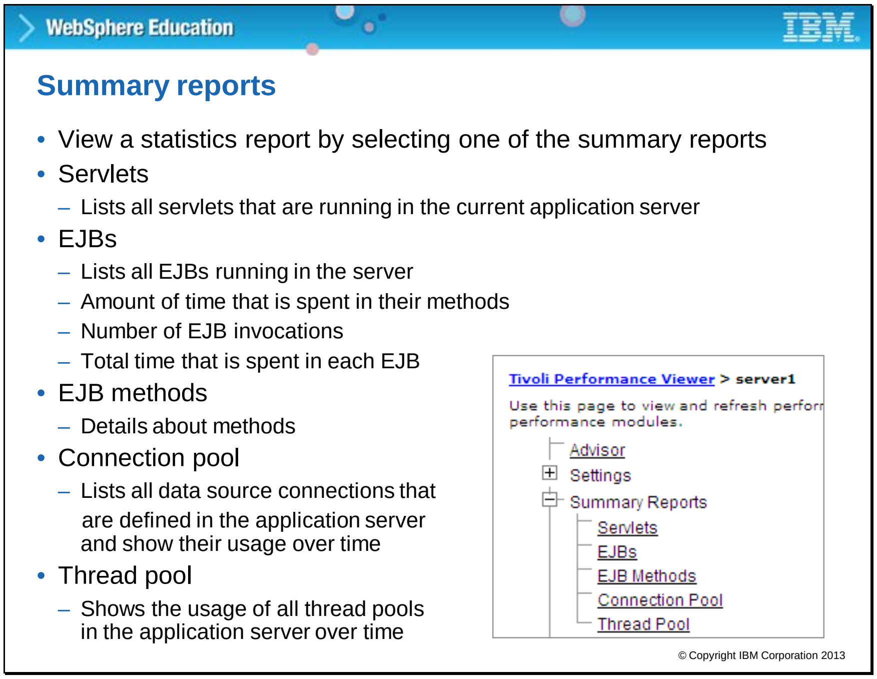
- Table view

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

Title: Tivoli Performance Viewer (4 of 4)

Statistics can also be viewed in a table format, by clicking the **View Table** button in the Tivoli Performance Viewer. This screen is an example of what the table view looks like for the JVM runtime module statistics.

Slide 23



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

Tivoli Performance Viewer > server1

Use this page to view and refresh perfor
performance modules.

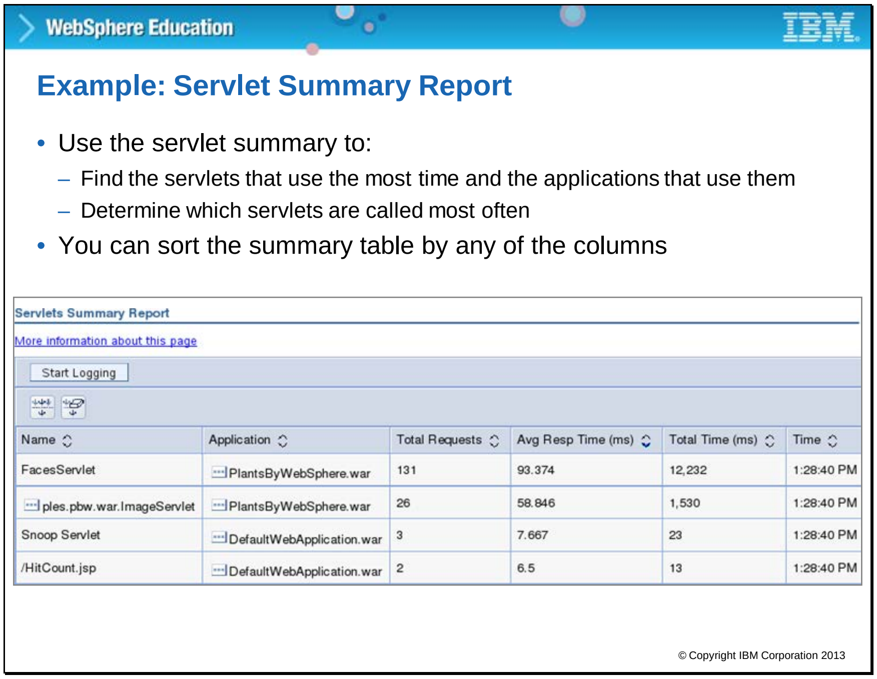
- Advisor
- ⊞ Settings
- ⊟ Summary Reports
  - Servlets
  - EJBs
  - EJB Methods
  - Connection Pool
  - Thread Pool

Title: Summary reports

Another option for viewing PMI data is selecting one of the summary reports available. These summary reports collect a number of statistics for a broader group of related components into a report form. Summary reports are available for:

- Servlets
- EJBs
- EJB methods
- Connection pools
- Thread pools

Slide 24



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

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

Title: Example: Servlet Summary Report

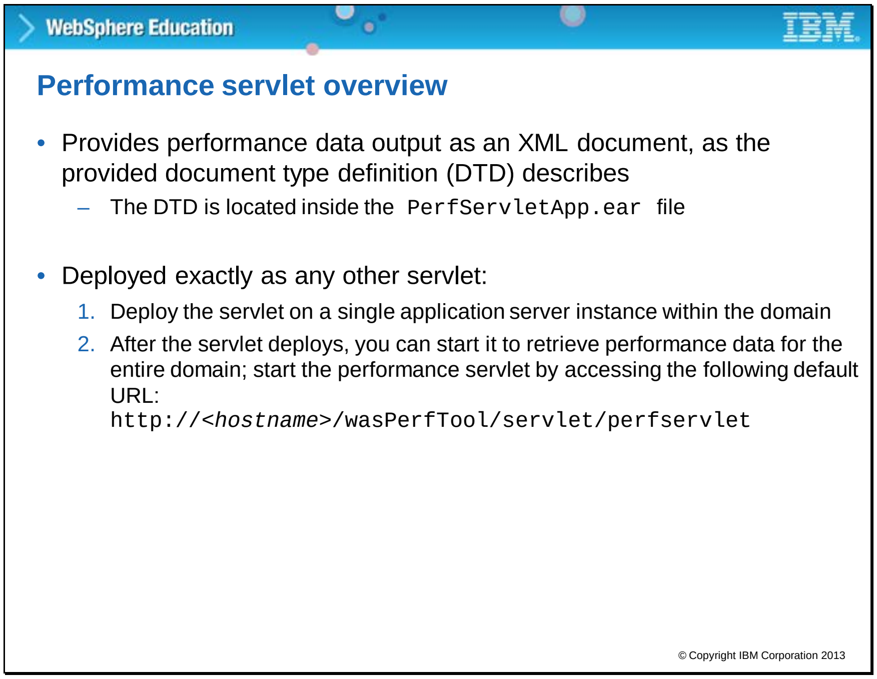
This screen shows an example of the servlet summary report. The servlet summary lists all servlets that are running in the current application server. Use the servlet summary view to quickly find the most time-intensive servlets and the applications that use them, and to determine which servlets are called most often.

You can sort the summary table by any of the columns. Some tips include:

- Sort by Avg Response Time to find the slowest servlet or JSP page.
- Sort by Total Requests to find the servlet or JSP used the most.
- Sort by Total Time to find the most costly servlet or JSP.

In this screen, 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.

Slide 25



WebSphere Education

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`

Title: Performance servlet overview

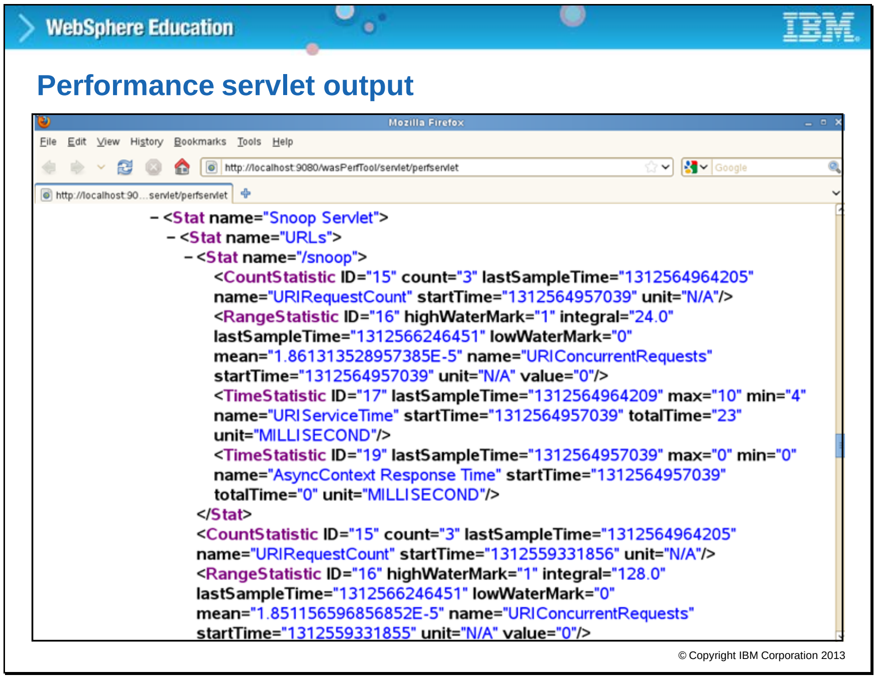
The performance servlet provides a way to use an HTTP request to query the performance metrics for an entire WebSphere Application Server administrative domain. Because the servlet provides the performance data through HTTP, issues such as firewalls, are trivial to resolve.

The performance servlet 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 uses the JMX Perf MBean interface to retrieve the PMI data and outputs an XML document that uses the Java EE Performance Data Framework to describe the statistics.

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.

Slide 26



Performance servlet output

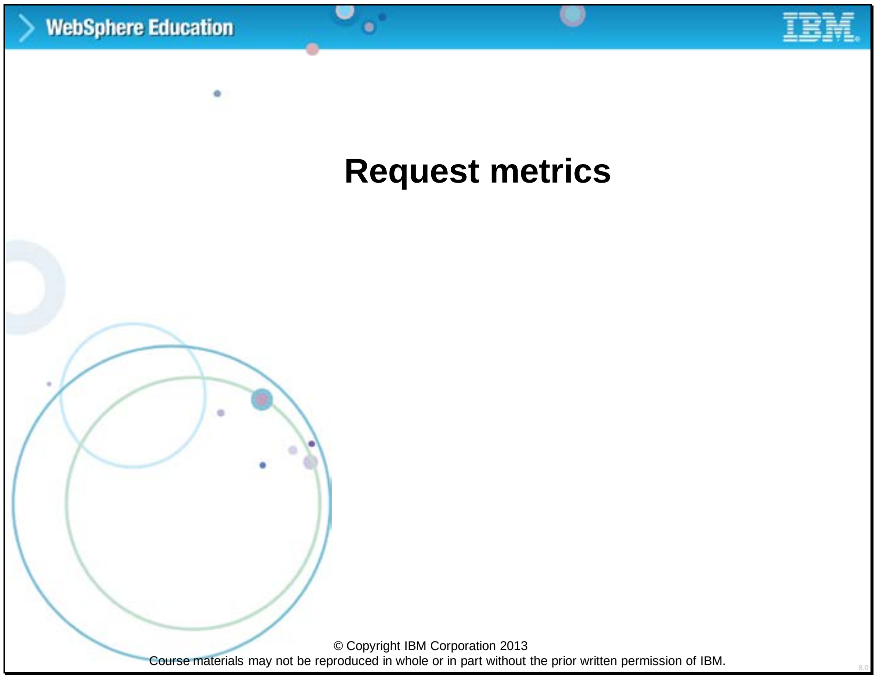
Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9080/wasPerfTool/servlet/perfservlet
Google
http://localhost:90....servlet/perfservlet

```
    - <Stat name="Snoop Servlet">
      - <Stat name="URLs">
        - <Stat name="/snoop">
            <CountStatistic ID="15" count="3" lastSampleTime="1312564964205"
            name="URIRequestCount" startTime="1312564957039" unit="N/A"/>
            <RangeStatistic ID="16" highWaterMark="1" integral="24.0"
            lastSampleTime="1312566246451" lowWaterMark="0"
            mean="1.861313528957385E-5" name="URIConcurrentRequests"
            startTime="1312564957039" unit="N/A" value="0"/>
            <TimeStatistic ID="17" lastSampleTime="1312564964209" max="10" min="4"
            name="URIServiceTime" startTime="1312564957039" totalTime="23"
            unit="MILLISECOND"/>
            <TimeStatistic ID="19" lastSampleTime="1312564957039" max="0" min="0"
            name="AsyncContext Response Time" startTime="1312564957039"
            totalTime="0" unit="MILLISECOND"/>
          </Stat>
          <CountStatistic ID="15" count="3" lastSampleTime="1312564964205"
          name="URIRequestCount" startTime="1312559331856" unit="N/A"/>
          <RangeStatistic ID="16" highWaterMark="1" integral="128.0"
          lastSampleTime="1312566246451" lowWaterMark="0"
          mean="1.851156596856852E-5" name="URIConcurrentRequests"
          startTime="1312559331855" unit="N/A" value="0"/>
```

Title: Performance servlet output

This slide shows you the output of a performance servlet request. The snapshot shows the performance statistics about Snoop servlet.

Slide 27



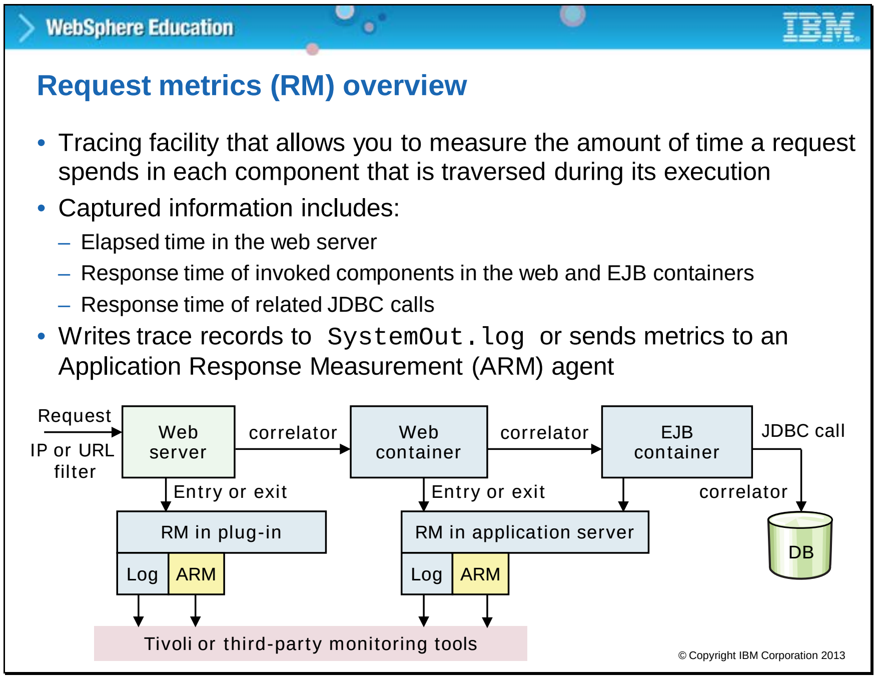
WebSphere Education

Request metrics

Topic: Request metrics

Request metrics are covered in this topic.

Slide 28



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

Request

IP or URL filter

Web server

correlator

Web container

correlator

EJB container

JDBC call

Entry or exit

Entry or exit

correlator

RM in plug-in

Log

ARM

RM in application server

Log

ARM

DB

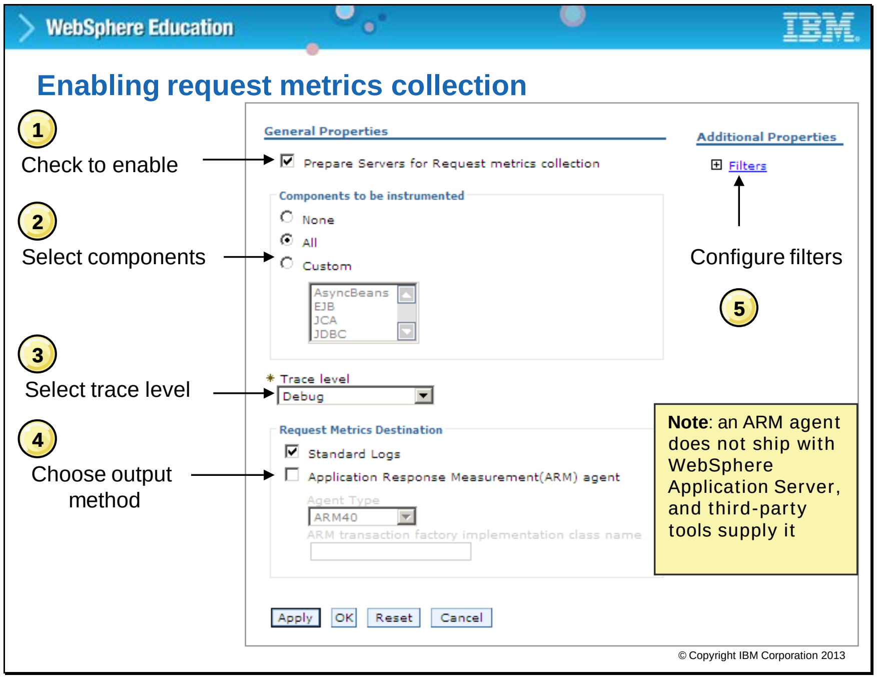
Tivoli or third-party monitoring tools

Title: Request metrics (RM) overview

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 milliseconds can be attributed to the servlet process.

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

Slide 29



Enabling request metrics collection

1 Check to enable

2 Select components

3 Select trace level

4 Choose output method

5 Configure filters

General Properties

☑ Prepare Servers for Request metrics collection

Components to be instrumented

○ None
● All
○ Custom

AsyncBeans
EJB
JCA
JDBC

* Trace level: Debug

Request Metrics Destination

☑ Standard Logs
☐ Application Response Measurement(ARM) agent

Agent Type: ARM40

ARM transaction factory implementation class name

Note: an ARM agent does not ship with WebSphere Application Server, and third-party tools supply it

© Copyright IBM Corporation 2013

Title: Enabling request metrics collection

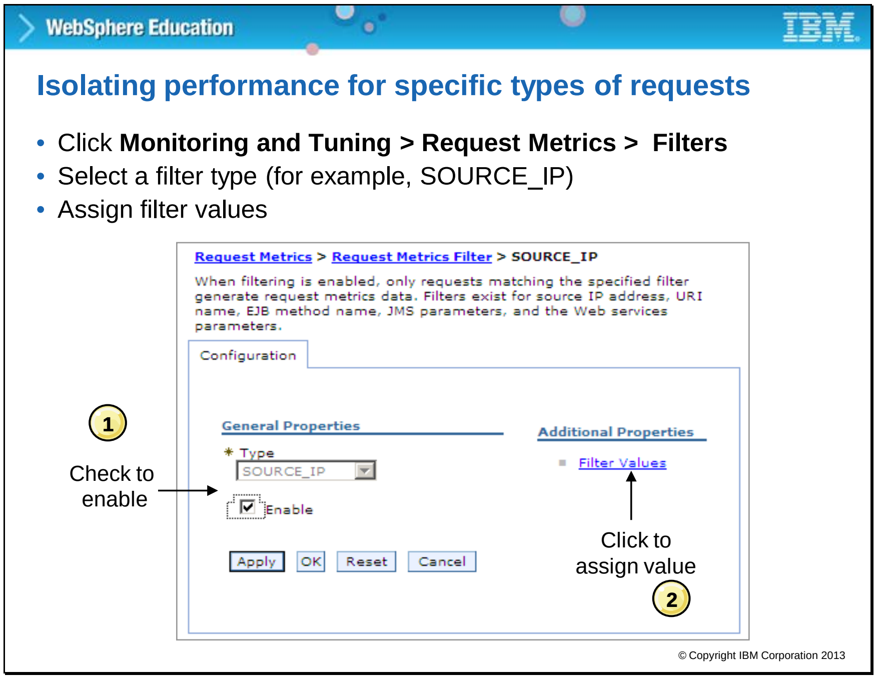
In the administrative console, select **Monitoring and Tuning > Request Metrics** and select the check box to **Prepare Servers for Request metrics collection**.

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.
- **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:** Enables the request metrics feature for logging. Select this check box to trigger the generation of request metrics logs in the SystemOut.log file. Enabling the request metrics feature for logging increases processor usage; therefore 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.

Slide 30



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

Request Metrics > Request Metrics Filter > SOURCE_IP

When filtering is enabled, only requests matching the specified filter generate request metrics data. Filters exist for source IP address, URI name, EJB method name, JMS parameters, and the Web services parameters.

© Copyright IBM Corporation 2013

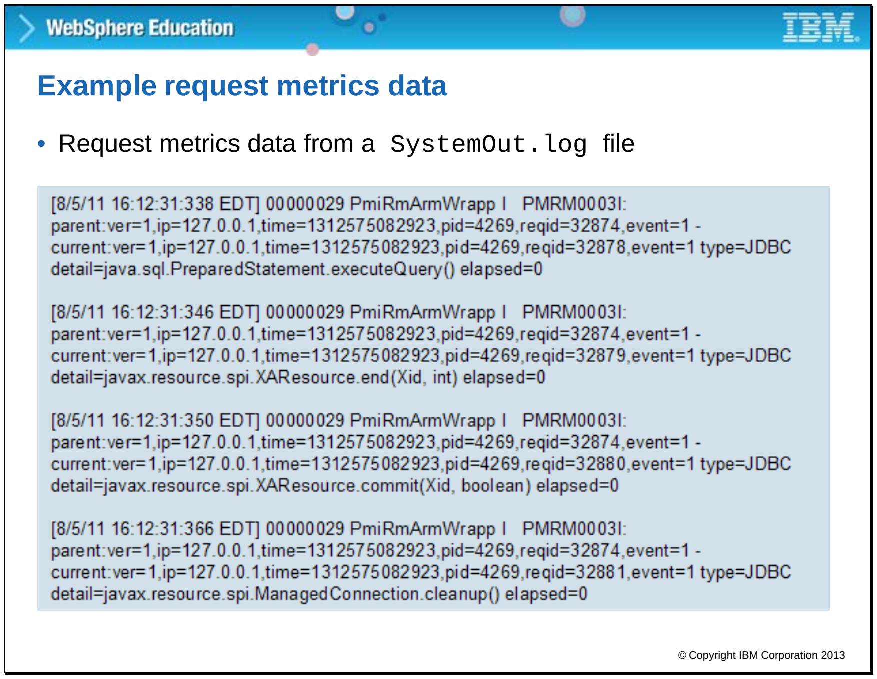
Title: Isolating performance for specific types of requests

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 EJB 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.

Slide 31



WebSphere Education
IBM

Example request metrics data

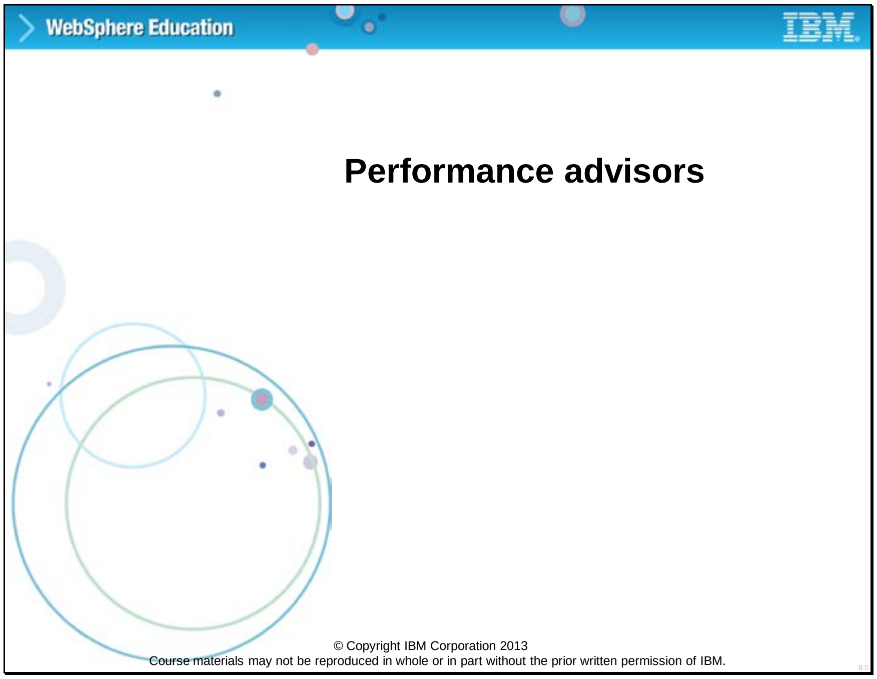
- Request metrics data from a `SystemOut.log` file

```
[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
```

Title: Example of request metrics data

The example of request metrics data that is shown on this slide, shows detail of an SQL call that is being made by using the prepared statement cache. You can trace the steps that are involved, and the timings for this database transaction.

Slide 32



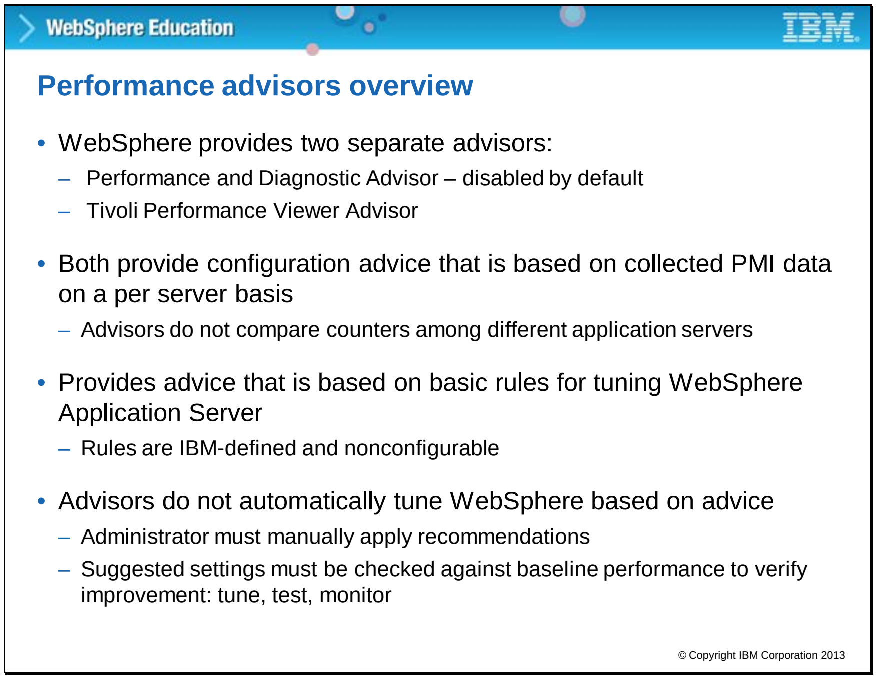
WebSphere Education

Performance advisors

Topic: Performance advisors

This topic describes performance advisors.

Slide 33



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

Title: Performance advisors overview

The advisors provide advice on the following application server resources:

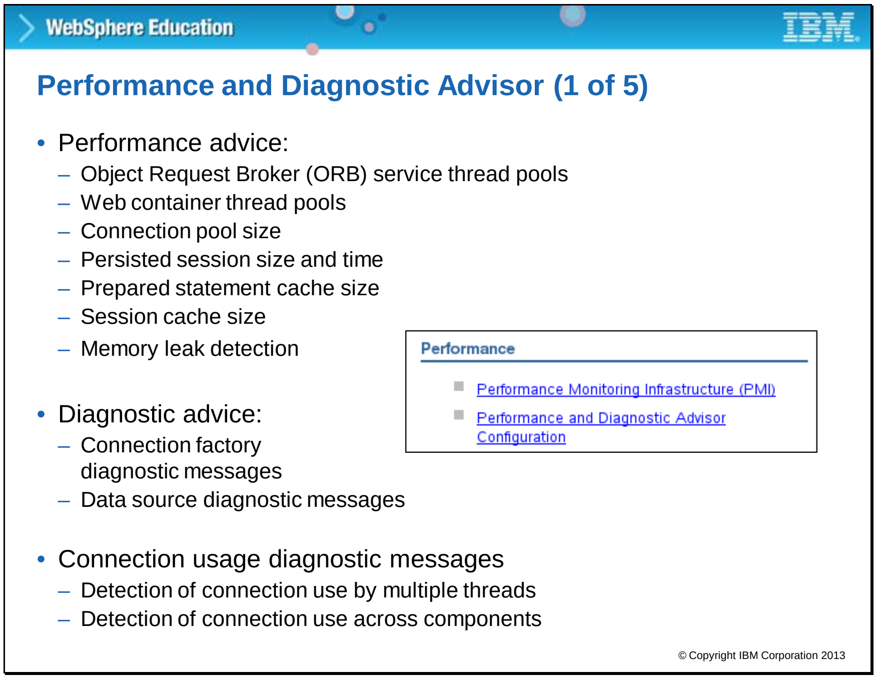
- Thread pools
- Persisted HTTP session sizes
- Cache sizes
- JVM heap size

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 called “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.

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.

Slide 34



WebSphere Education

IBM

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

Performance

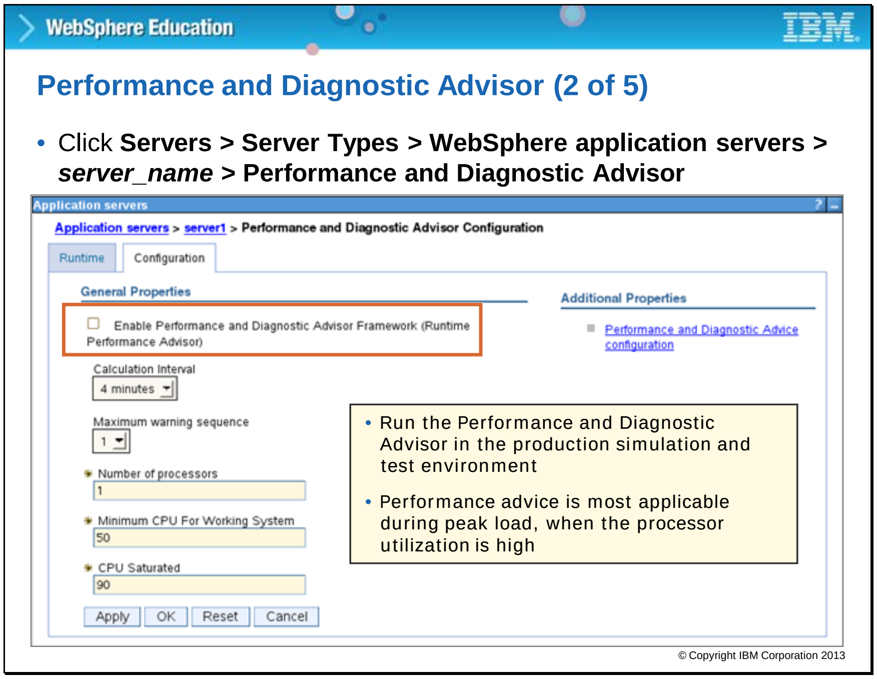
- Performance Monitoring Infrastructure (PMI)
- Performance and Diagnostic Advisor Configuration

Title: Performance and Diagnostic Advisor (1 of 5)

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.**

Slide 35



Performance and Diagnostic Advisor (2 of 5)

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

Application servers

Application servers > server1 > Performance and Diagnostic Advisor Configuration

Runtime | Configuration

General Properties

□ Enable Performance and Diagnostic Advisor Framework (Runtime Performance Advisor)

Calculation Interval: 4 minutes

Maximum warning sequence: 1

* Number of processors: 1

* Minimum CPU For Working System: 50

* CPU Saturated: 90

Apply | OK | Reset | Cancel

Additional Properties

- Performance and Diagnostic Advice configuration

- 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 Corporation 2013

Title: Performance and Diagnostic Advisor (2 of 5)

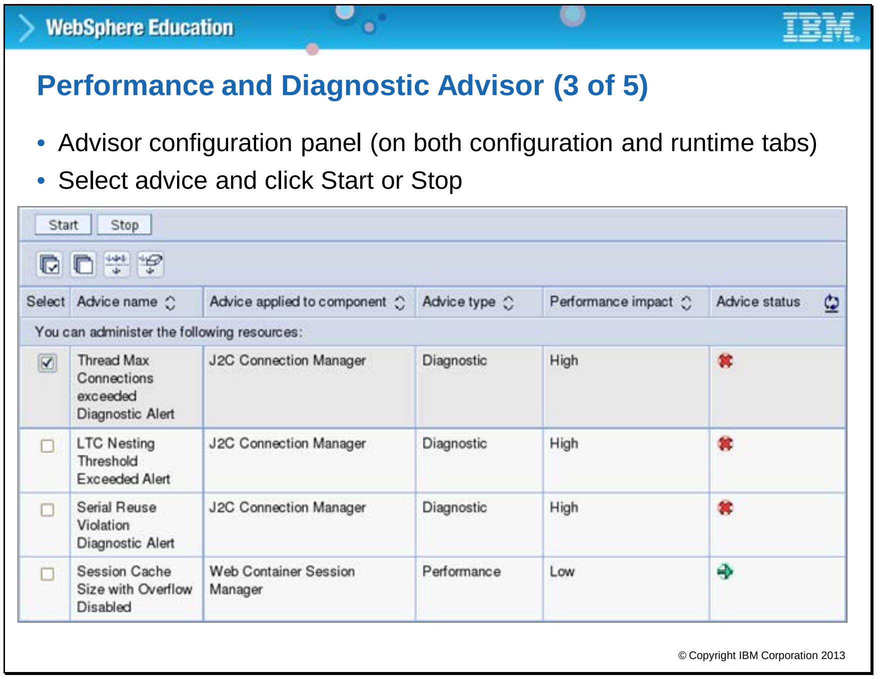
The Performance and Diagnostic Advisor analyzes PMI data and receives notifications about performance and diagnostic information from components. Use this page to specify settings for the Performance and Diagnostic Advisor. Performance issues can be related to memory leaks in the system.

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 use is high.

Slide 36



Performance and Diagnostic Advisor (3 of 5)

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

Select	Advice name	Advice applied to component	Advice type	Performance impact	Advice status
You can administer the following resources:					
☑	Thread Max Connections exceeded Diagnostic Alert	J2C Connection Manager	Diagnostic	High	✖
□	LTC Nesting Threshold Exceeded Alert	J2C Connection Manager	Diagnostic	High	✖
□	Serial Reuse Violation Diagnostic Alert	J2C Connection Manager	Diagnostic	High	✖
□	Session Cache Size with Overflow Disabled	Web Container Session Manager	Performance	Low	➔

Title: Performance and Diagnostic Advisor (3 of 5)

Advice type categorizes the primary intent of a piece of advice.

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

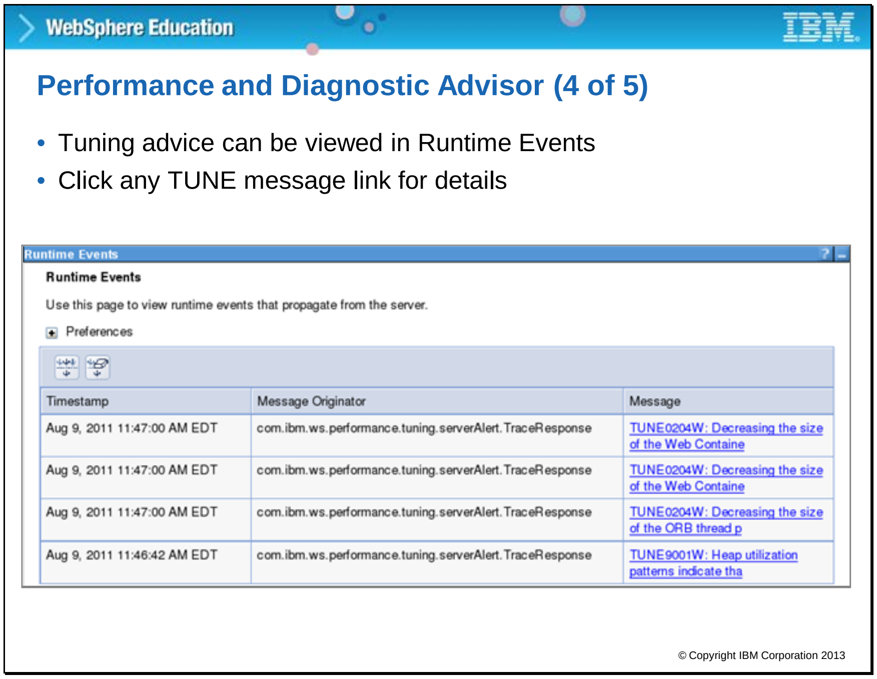
Performance: Performance advice provides tuning recommendations, or identifies problems with your configuration from a performance perspective.

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:

- Low
- Medium
- High

Slide 37



Performance and Diagnostic Advisor (4 of 5)

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

Runtime Events

Runtime Events

Use this page to view runtime events that propagate from the server.

Preferences

Timestamp	Message Originator	Message
Aug 9, 2011 11:47:00 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE0204W: Decreasing the size of the Web Containe
Aug 9, 2011 11:47:00 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE0204W: Decreasing the size of the Web Containe
Aug 9, 2011 11:47:00 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE0204W: Decreasing the size of the ORB thread p
Aug 9, 2011 11:46:42 AM EDT	com.ibm.ws.performance.tuning.serverAlert.TraceResponse	TUNE9001W: Heap utilization patterns indicate tha

Title: Performance and Diagnostic Advisor (4 of 5)

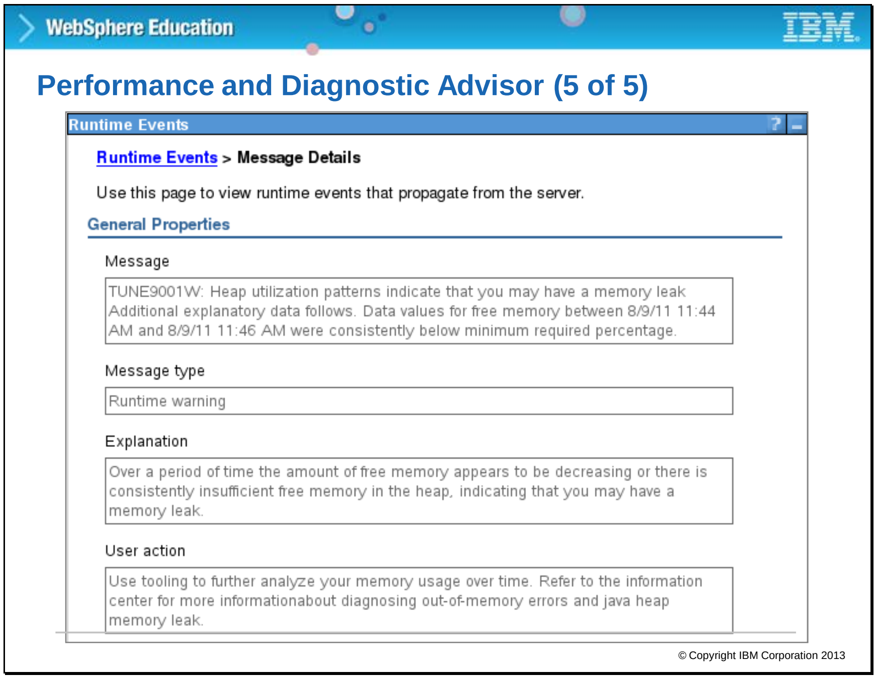
Tuning advice is provided as messages written to the runtime events page. The TUNExxxx messages are typically at the Warning level.

For examples,

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

A complete list of TUNE messages is available in the WebSphere Application Server V8.5 information center.

Slide 38



Performance and Diagnostic Advisor (5 of 5)

Runtime Events > **Message Details**

Use this page to view runtime events that propagate from the server.

General Properties

Message

TUNE9001W: Heap utilization patterns indicate that you may 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 below minimum required percentage.

Message type

Runtime warning

Explanation

Over a period of time the amount of free memory appears to be decreasing or there is consistently insufficient free memory in the heap, indicating that you may have a memory leak.

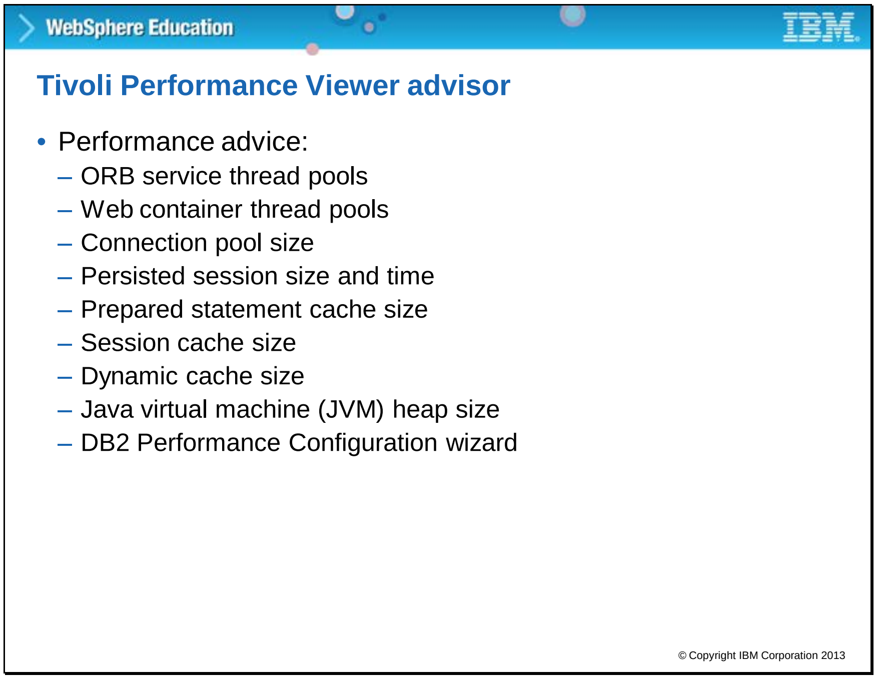
User action

Use tooling to further analyze your memory usage over time. Refer to the information center for more informationabout diagnosing out-of-memory errors and java heap memory leak.

Title: Performance and Diagnostic Advisor (5 of 5)

When you select a tune message from the runtime events panel, it displays the detailed information. Notice that you get the message, an explanation, and an action that the user can take. The advisor does not itself make the recommended change; the administrator must apply the advice.

Slide 39



WebSphere Education

Tivoli Performance Viewer advisor

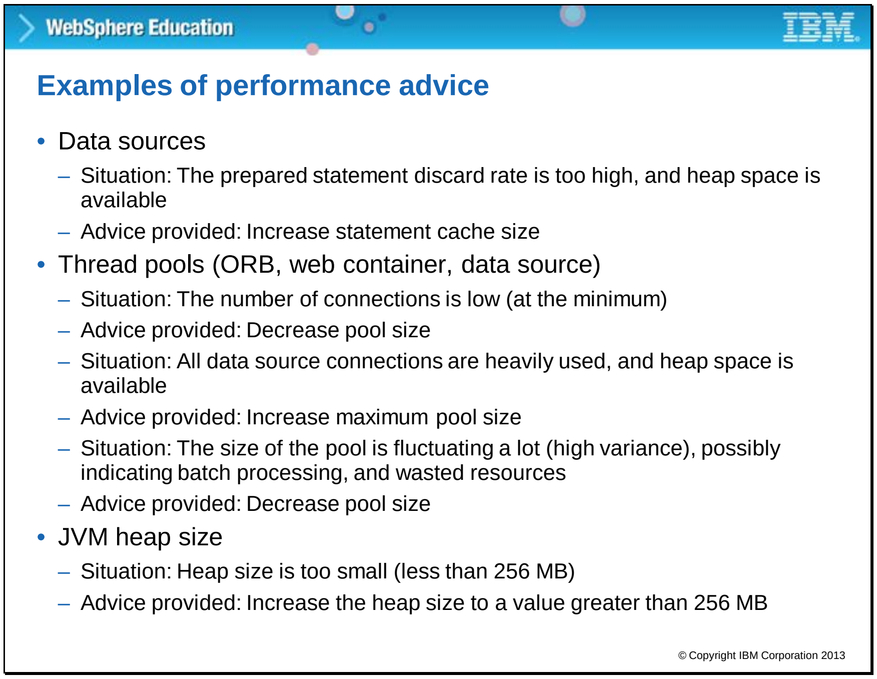
- Performance advice:
  - ORB service thread pools
  - Web container thread pools
  - Connection pool size
  - Persisted session size and time
  - Prepared statement cache size
  - Session cache size
  - Dynamic cache size
  - Java virtual machine (JVM) heap size
  - DB2 Performance Configuration wizard

Title: Tivoli Performance Viewer advisor

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

Slide 40



WebSphere Education
IBM

Examples of performance advice

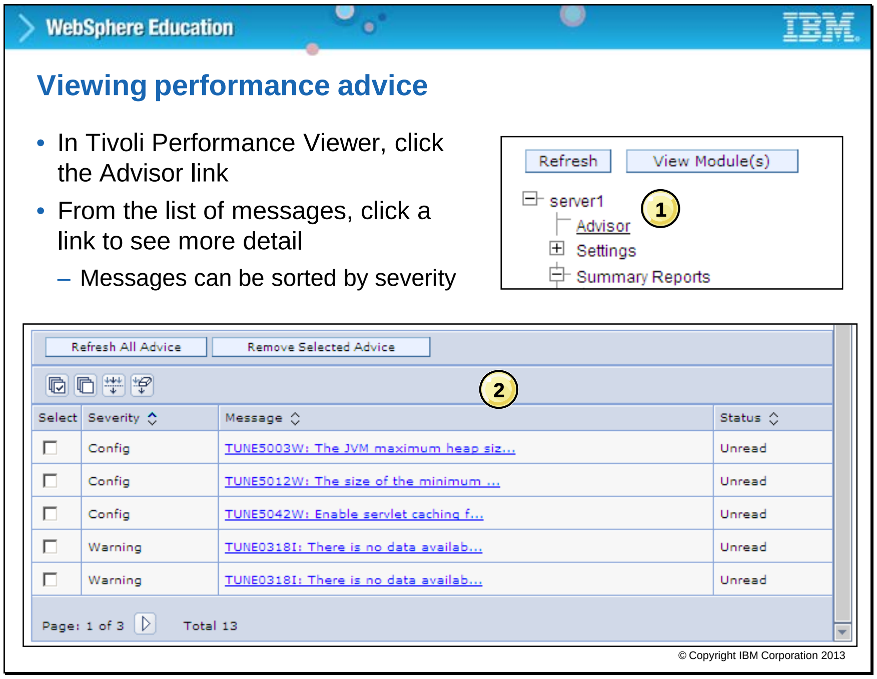
- Data sources
  - Situation: The prepared statement discard rate is too high, and heap space is available
  - Advice provided: Increase statement cache size
- Thread pools (ORB, web container, data source)
  - Situation: The number of connections is low (at the minimum)
  - Advice provided: Decrease 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

Title: Examples of performance advice

Here are some examples of performance advice.

The data source example is using PMI statistics about the rate of prepared statements that are discarded from the cache, which is high. The setting implies that some connections are creating new prepared statements instead of retrieving ones from the cache, which is not a good practice as the application server is creating and removing prepared statements. Also, the PMI statistics are showing that there is enough available memory in the heap. The advice is to increase the size of the prepared statement cache.

Slide 41



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

Select	Severity	Message	Status
□	Config	TUNE5003W: The JVM maximum heap siz...	Unread
□	Config	TUNE5012W: The size of the minimum ...	Unread
□	Config	TUNE5042W: Enable servlet caching f...	Unread
□	Warning	TUNE0318I: There is no data availab...	Unread
□	Warning	TUNE0318I: There is no data availab...	Unread

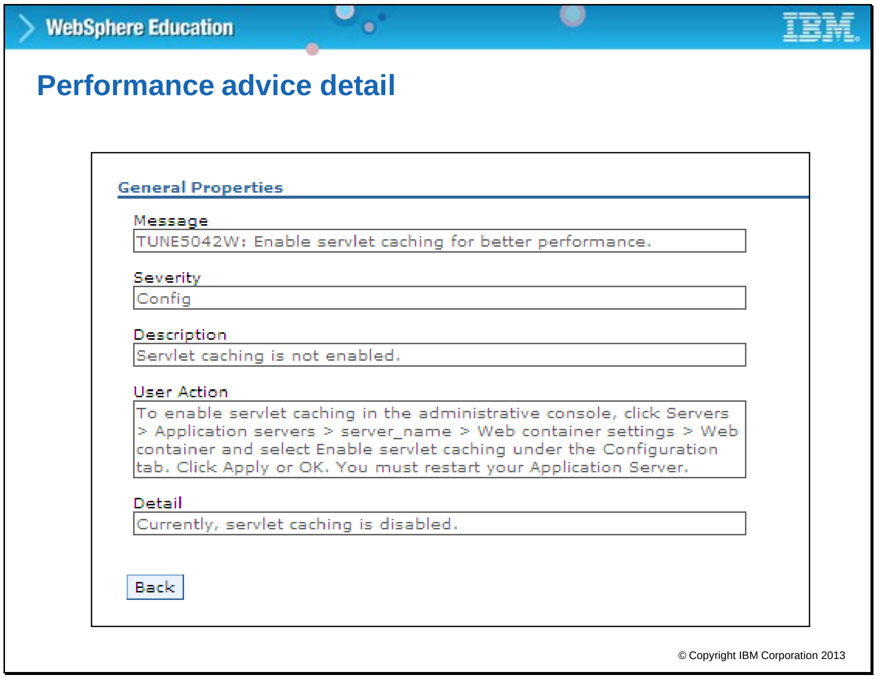
Page: 1 of 3 Total 13

Title: Viewing performance advice

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

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

Slide 42



Performance advice detail

General Properties

Message
TUNE5042W: Enable servlet caching for better performance.

Severity
Config

Description
Servlet caching is not enabled.

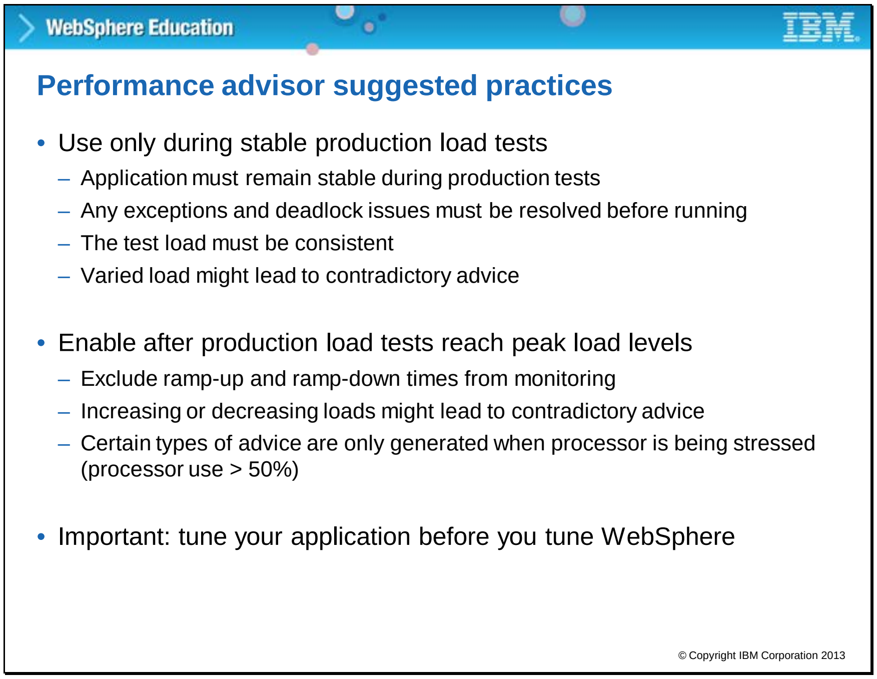
User Action
To enable servlet caching in the administrative console, click Servers > Application servers > server_name > Web container settings > Web container and select Enable servlet caching under the Configuration tab. Click Apply or OK. You must restart your Application Server.

Detail
Currently, servlet caching is disabled.

Title: Performance advice detail

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.

Slide 43



WebSphere Education

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

Title: Performance advisor suggested practices

When using the performance advisors, processor usage 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.

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 using this setting. The advisors do not take into account the amount of actual physical memory available on the system.

Slide 44

WebSphere Education

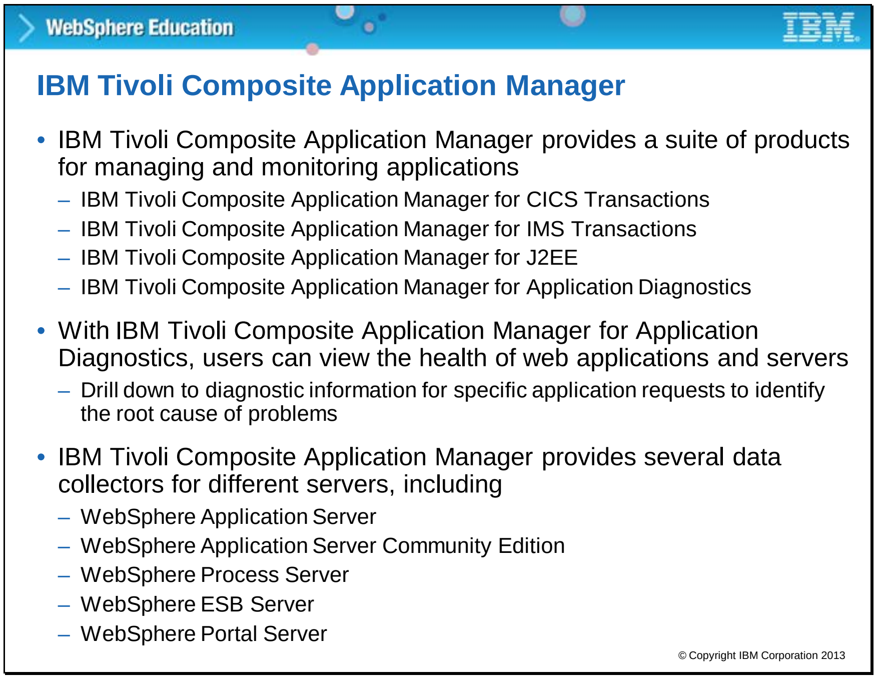
IBM Tivoli Composite Application Manager for WebSphere Application Server

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

Topic: ITCAM for WebSphere Application Server

This topic describes ITCAM for WebSphere and how to configure and use it.

Slide 45



WebSphere Education

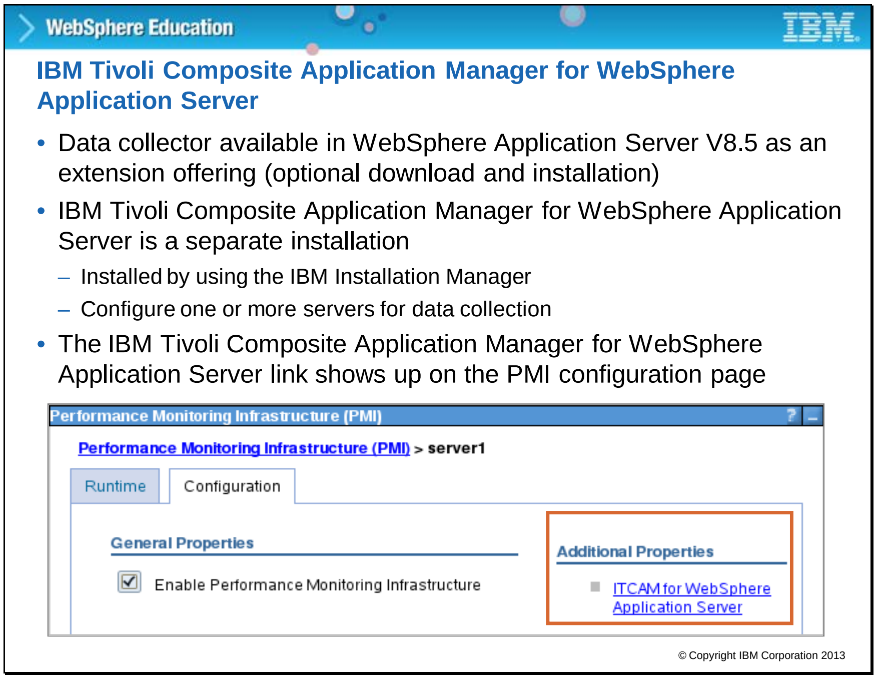
IBM Tivoli Composite Application Manager

- 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

Title: IBM Tivoli Composite Application Manager (ITCAM)

IBM Tivoli Composite Application Manager is a suite of products that are used to monitor and manage applications. ITCAM enables users to view the health of applications and servers. ITCAM has several data collectors for different servers.

Slide 46



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

Performance Monitoring Infrastructure (PMI)

Performance Monitoring Infrastructure (PMI) > server1

Runtime | Configuration

General Properties

☑ Enable Performance Monitoring Infrastructure

Additional Properties

- ITCAM for WebSphere Application Server

© Copyright IBM Corporation 2013

Title: ITCAM for WebSphere Application Server

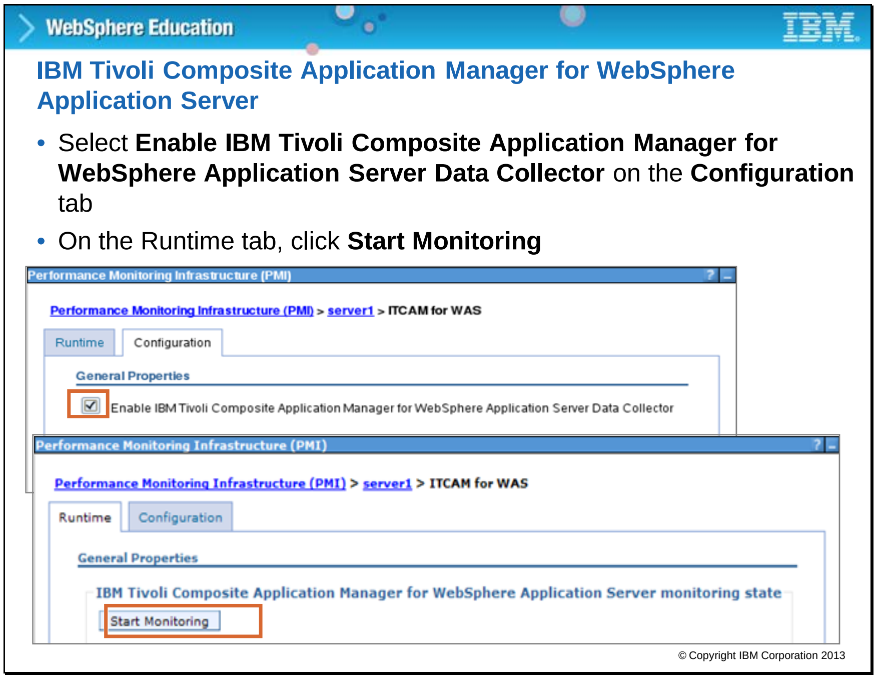
IBM Tivoli Composite Application Manager (ITCAM) for WebSphere Application Server is enhanced in version 8.0 and can be installed together with the application server. This integrated monitoring tool allows you to view the health of web applications and servers, and drill down to diagnostic information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

ITCAM 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 ITCAM interface before you can see the ITCAM for WebSphere Application Server link under Additional Properties on the PMI configuration tab of the server.

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

Slide 47



WebSphere Education

IBM

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**

Performance Monitoring Infrastructure (PMI)

Performance Monitoring Infrastructure (PMI) > server1 > ITCAM for WAS

Runtime | Configuration

General Properties

☑ Enable IBM Tivoli Composite Application Manager for WebSphere Application Server Data Collector

Performance Monitoring Infrastructure (PMI)

Performance Monitoring Infrastructure (PMI) > server1 > ITCAM for WAS

Runtime | Configuration

General Properties

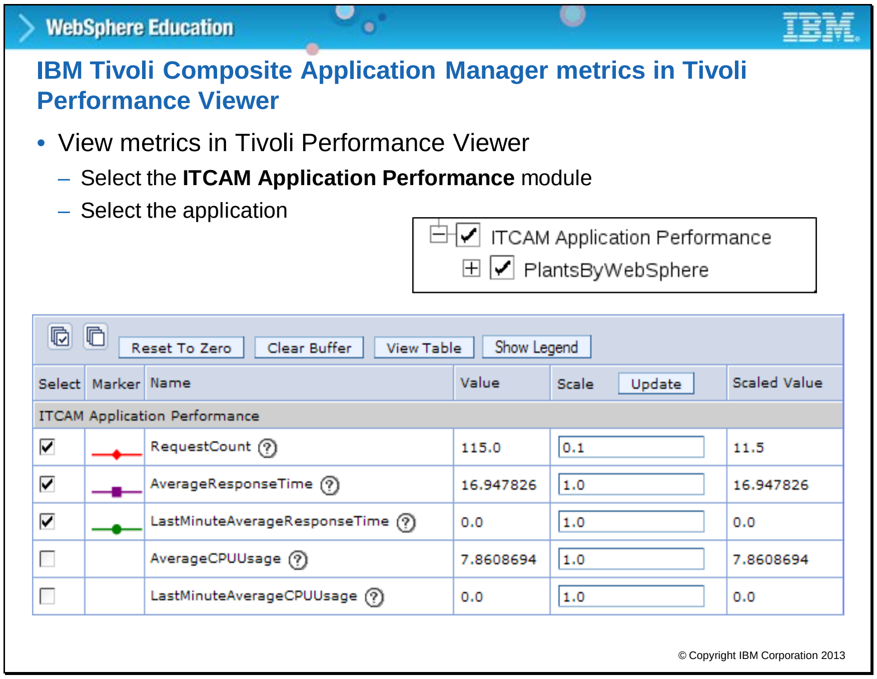
IBM Tivoli Composite Application Manager for WebSphere Application Server monitoring state

Start Monitoring

Title: ITCAM for WebSphere Application Server

To see the ITCAM 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 the **Start Monitoring** button on the runtime tab.

Slide 48



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

⊟ ☑ ITCAM Application Performance
⊞ ☑ PlantsByWebSphere

Select	Marker	Name	Value	Scale Update	Scaled Value
ITCAM Application Performance					
☑	◆	RequestCount ?	115.0	0.1	11.5
☑	■	AverageResponseTime ?	16.947826	1.0	16.947826
☑	●	LastMinuteAverageResponseTime ?	0.0	1.0	0.0
□		AverageCPUUsage ?	7.8608694	1.0	7.8608694
□		LastMinuteAverageCPUUsage ?	0.0	1.0	0.0

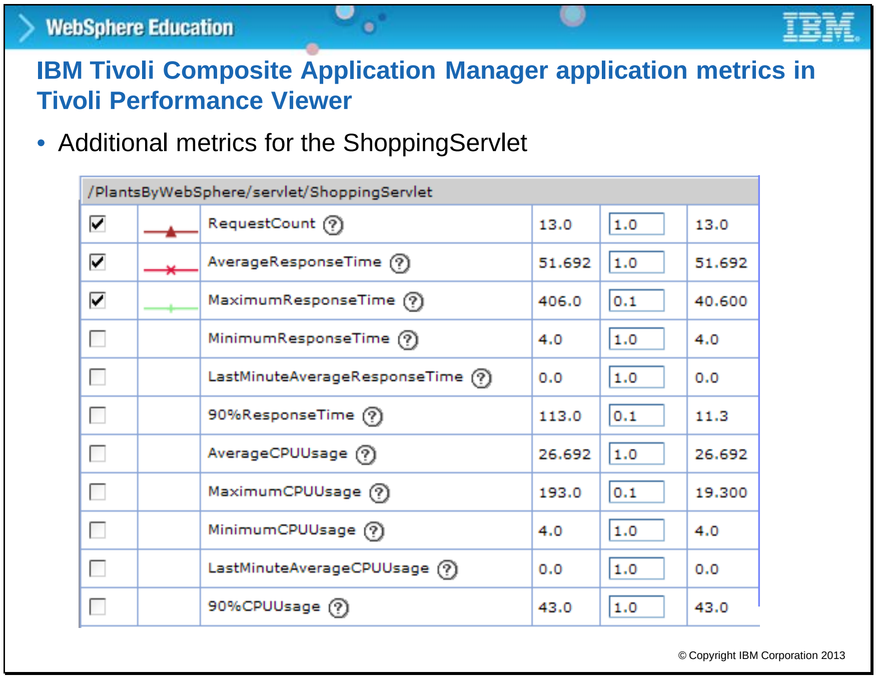
Title: ITCAM metrics in Tivoli Performance Viewer

To view ITCAM monitoring information in Tivoli Performance Viewer, you must enable the counters.

1. In the navigation pane, click **Monitoring and Tuning** > **Performance Viewer** > **Current Activity**.
2. Expand **Performance Modules** and click **ITCAM Application Performance**.
3. To refresh the view, click **Tivoli Performance Viewer** and select the *application_server* instance for which you want to view performance data.

ITCAM for WebSphere Application Server provides more request-based response time and processor metrics.

Slide 49



IBM Tivoli Composite Application Manager application metrics in Tivoli Performance Viewer

- Additional metrics for the ShoppingServlet

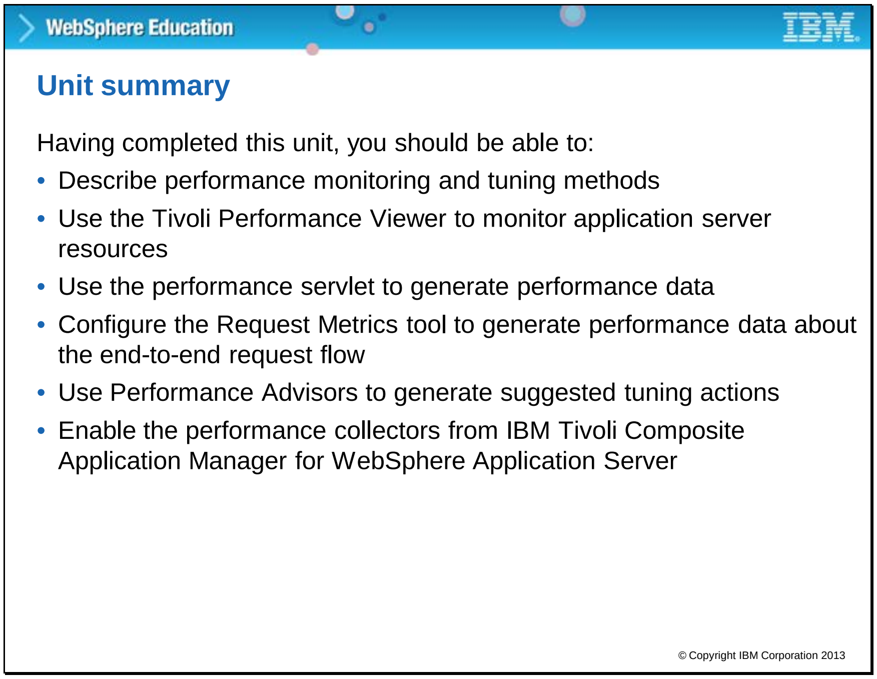
/PlantsByWebSphere/servlet/ShoppingServlet

☑	▲	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

Title: ITCAM application metrics in Tivoli Performance Viewer

Several performance metrics are collected and displayed for each component of an application. This screen 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.

Slide 50



WebSphere Education

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

Title: 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 tuning actions
- Enable the performance collectors from IBM Tivoli Composite Application Manager for WebSphere Application Server