Performance monitoring

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

Topics

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

Performance tuning and monitoring

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

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

Performance terminology

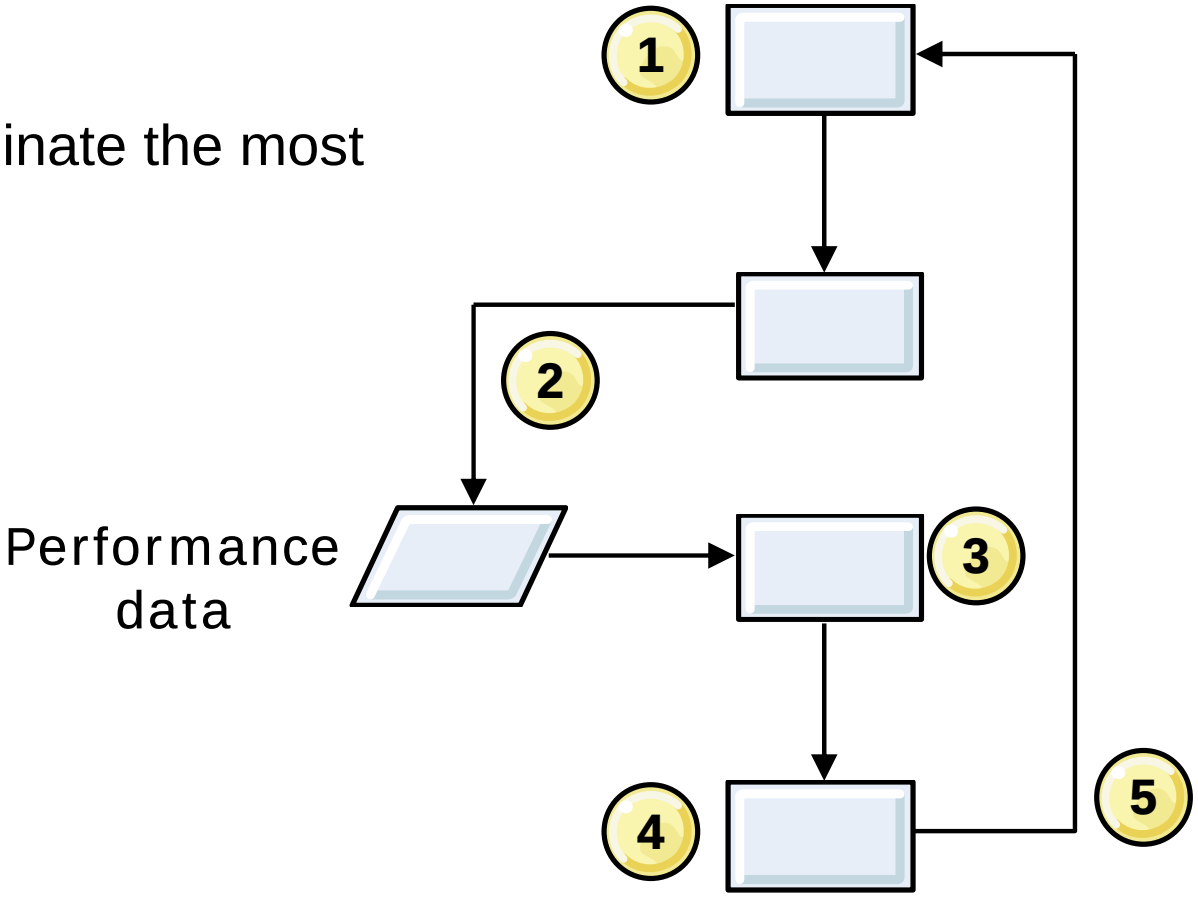
- **Response time** measures an **individual** user's average wait for a request
- Response time includes:
  - Processing time
  - Transit time
  - Wait time in queues
- **Throughput** measures activities that are completed in a unit of time
  - Example: Website pages that are served per second
- **Bottleneck** 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

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

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



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

WebSphere performance tools (1 of 3)

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

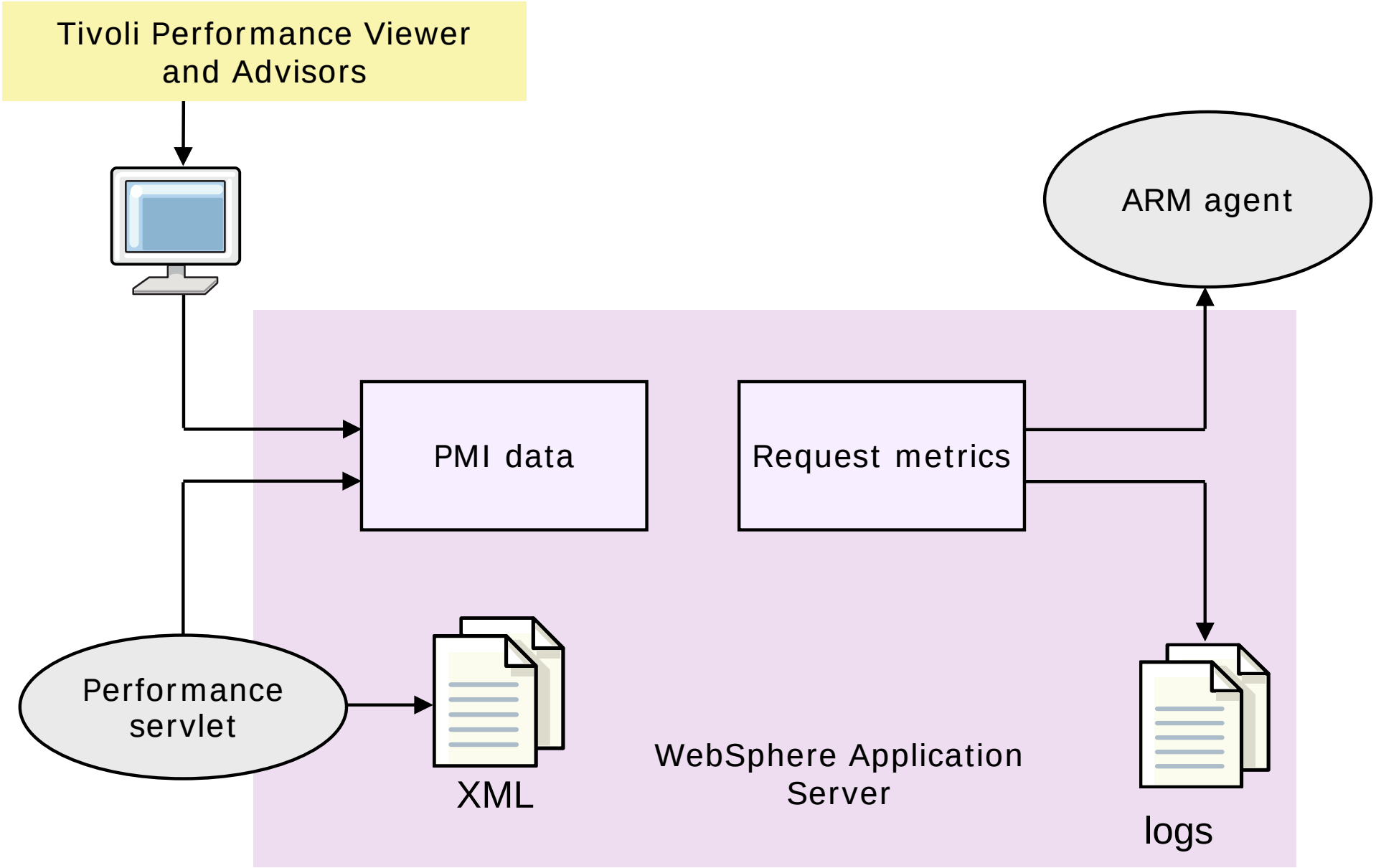
- Tivoli Performance Viewer
  - With Tivoli Performance Viewer, administrators can monitor the overall health of WebSphere Application Server
  - Accessed from within the administrative console

- 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

WebSphere performance tools (2 of 3)

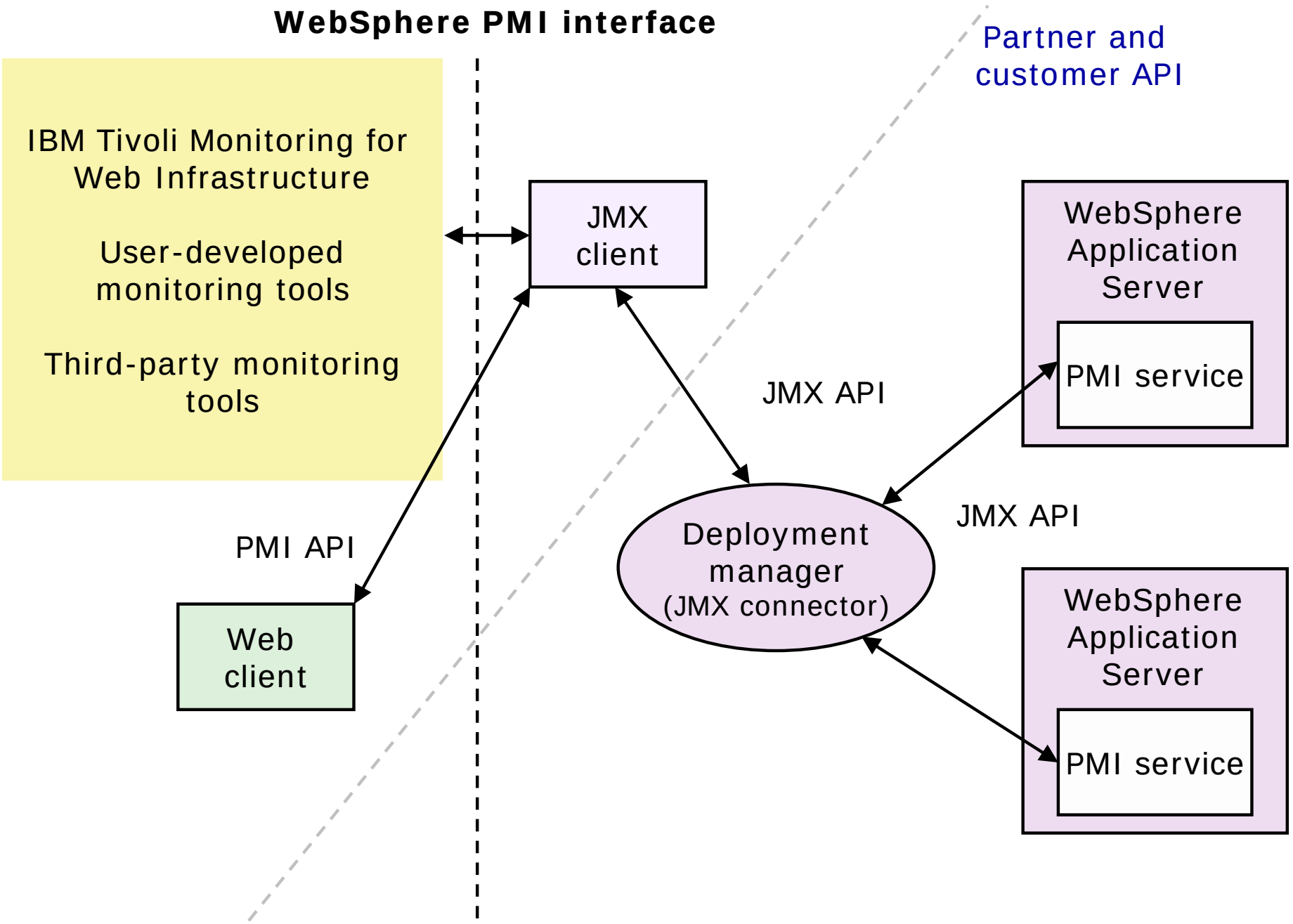
- Request Metrics (tool)
  - With request metrics, you can track individual transactions, recording the processing time in each of the major WebSphere Application Server components
  - Output is viewed in standard logs or by using an Application Response Measurement (ARM)-based tool
- Performance servlet
  - 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

WebSphere performance tools (3 of 3)



ARM= Application Response Measurement

PMI architecture



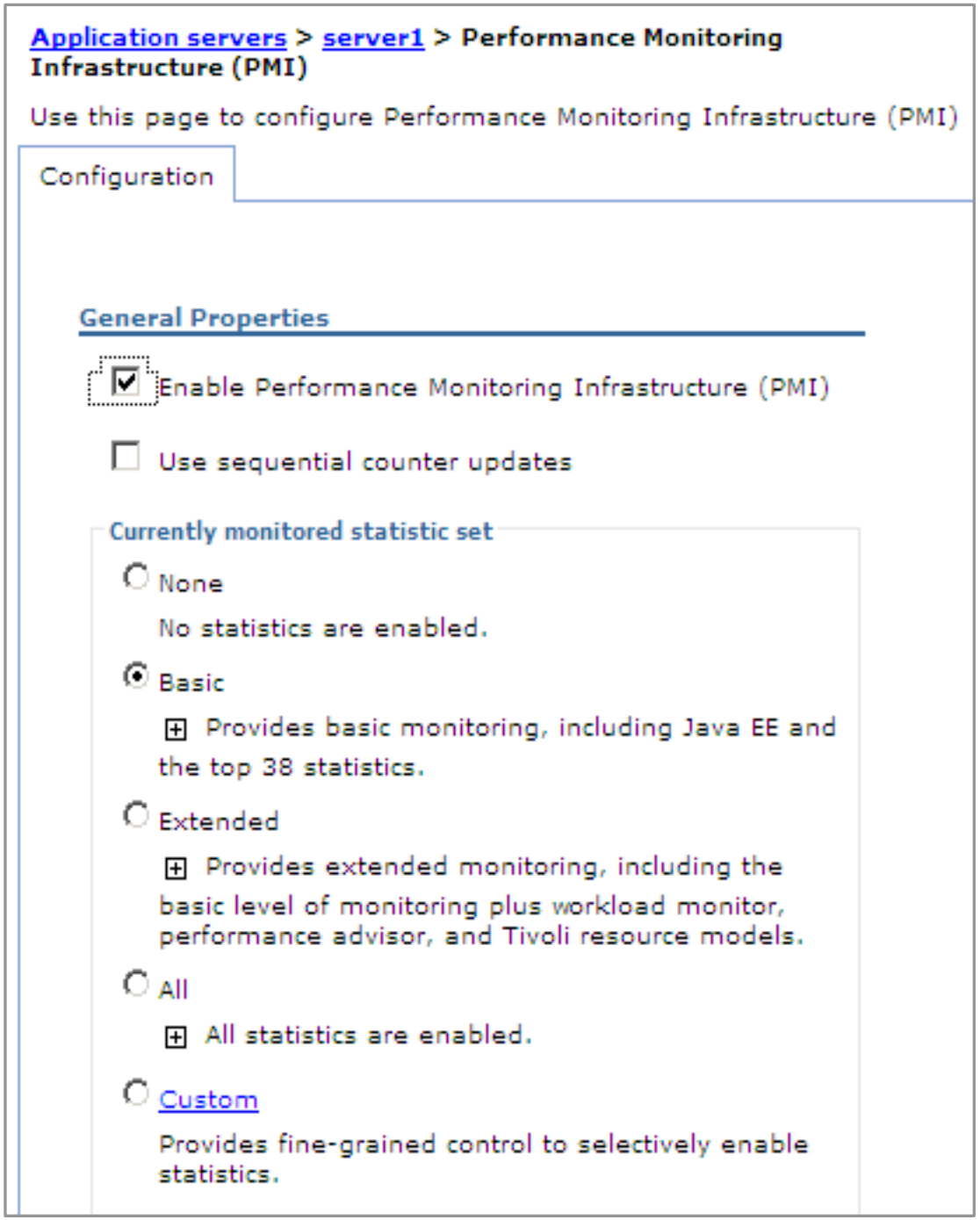
Types of performance data

- 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

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

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

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

Tivoli Performance Viewer

Preferences

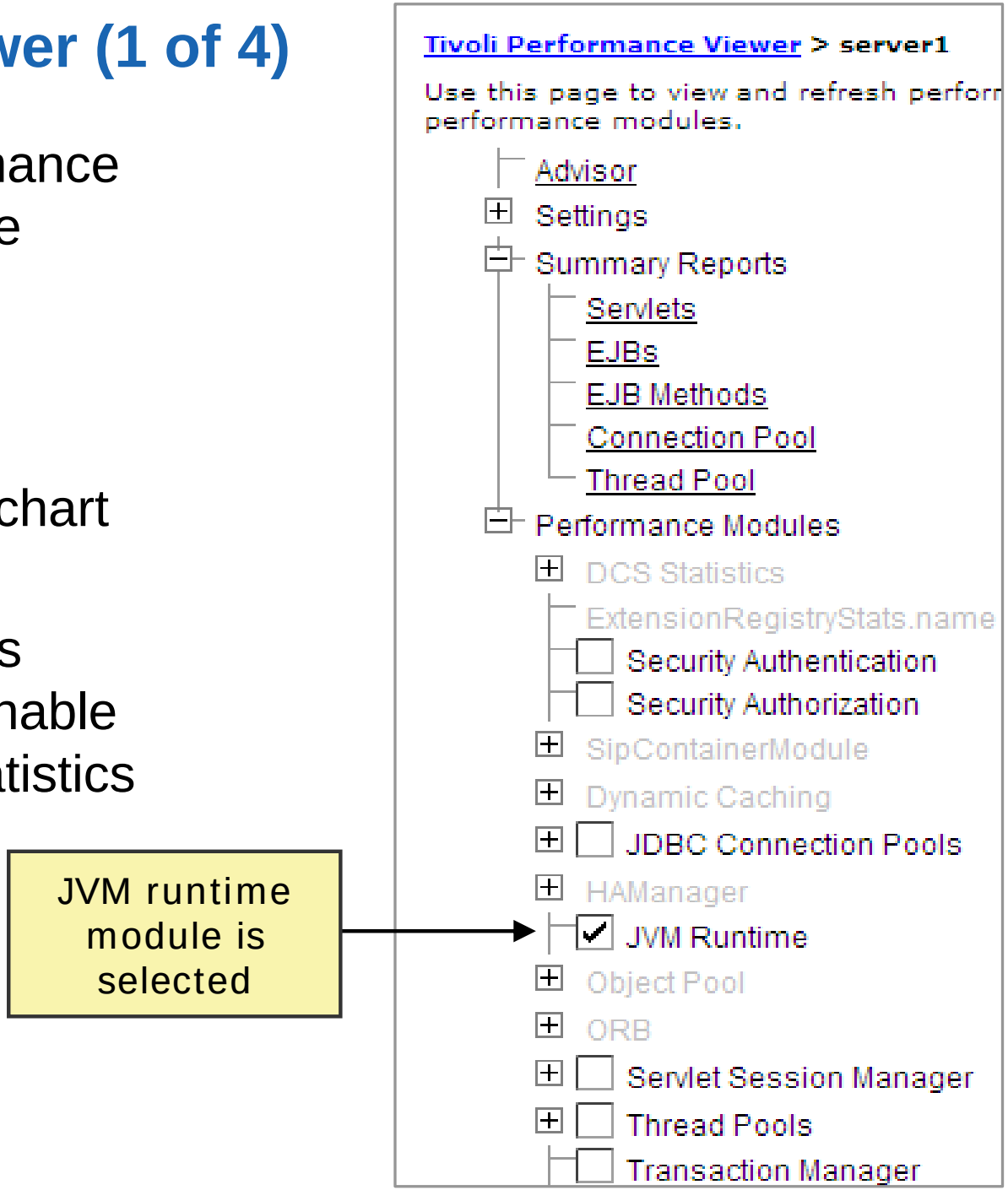
Start Monitoring | Stop Monitoring

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

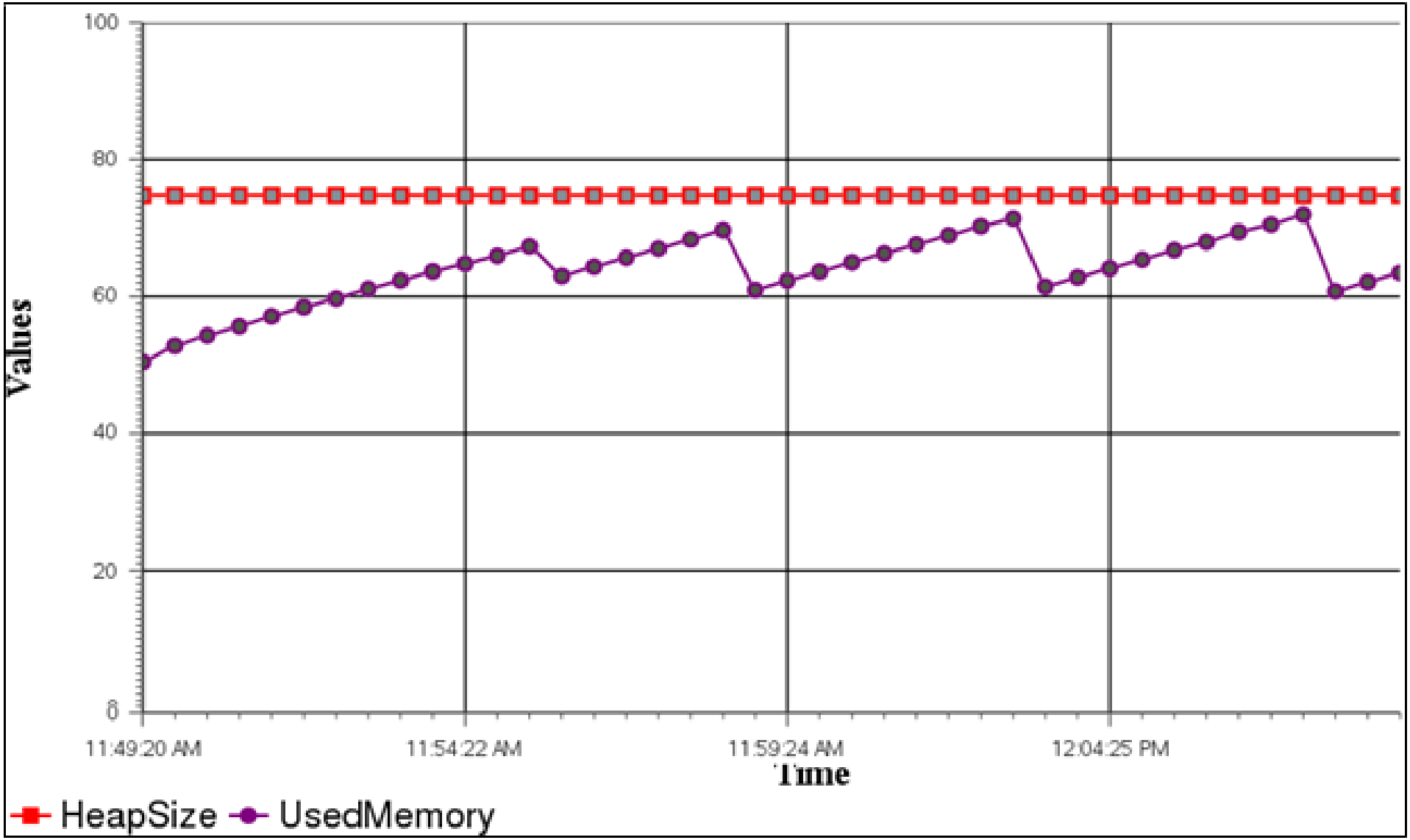
Tivoli Performance Viewer (1 of 4)

- 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



Tivoli Performance Viewer (2 of 4)

- Chart view of JVM metrics



Tivoli Performance Viewer (3 of 4)

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

Reset To Zero | Clear Buffer | View Table | Hide Legend

Select	Marker	Name	Value	Scale (Update)	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

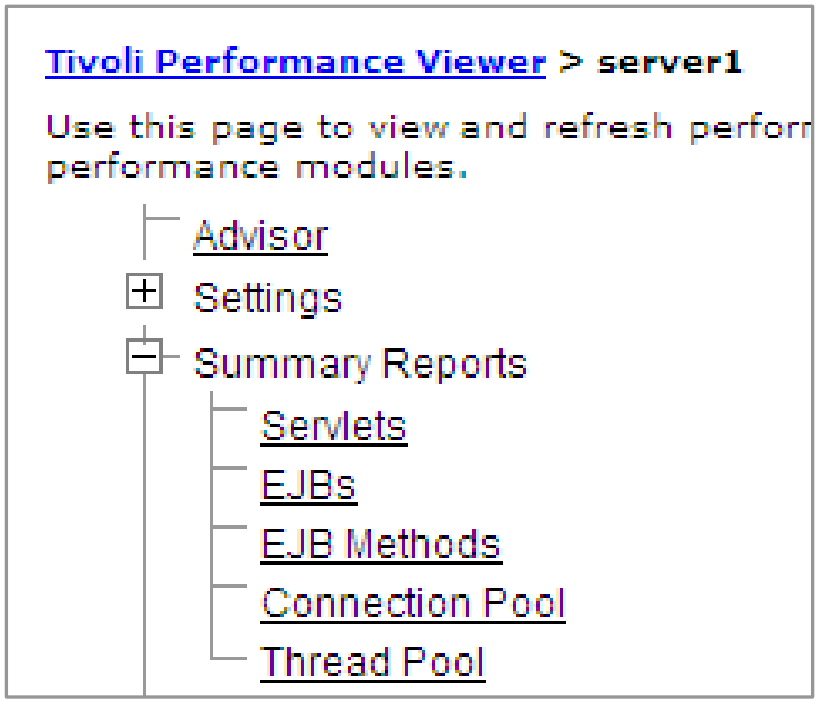
Tivoli Performance Viewer (4 of 4)

- 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

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



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

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`

Performance servlet output

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:9080/wasPerfTool/servlet/perfservlet

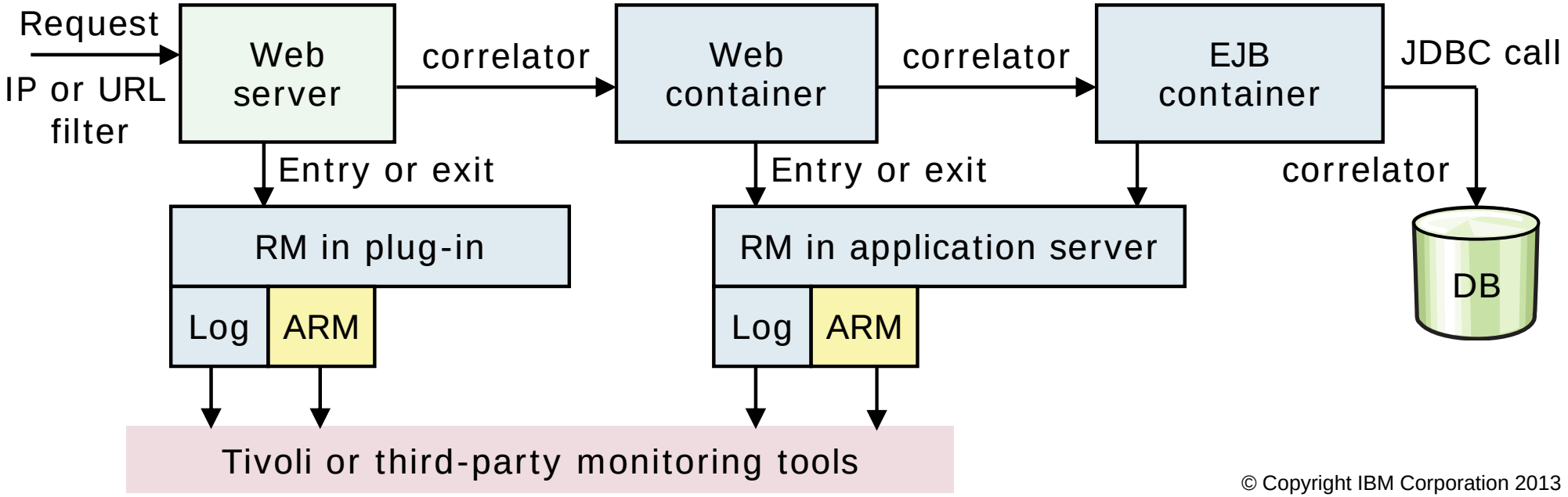
http://localhost:90...servlet/perfservlet


```
- <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"/>
```

Request metrics

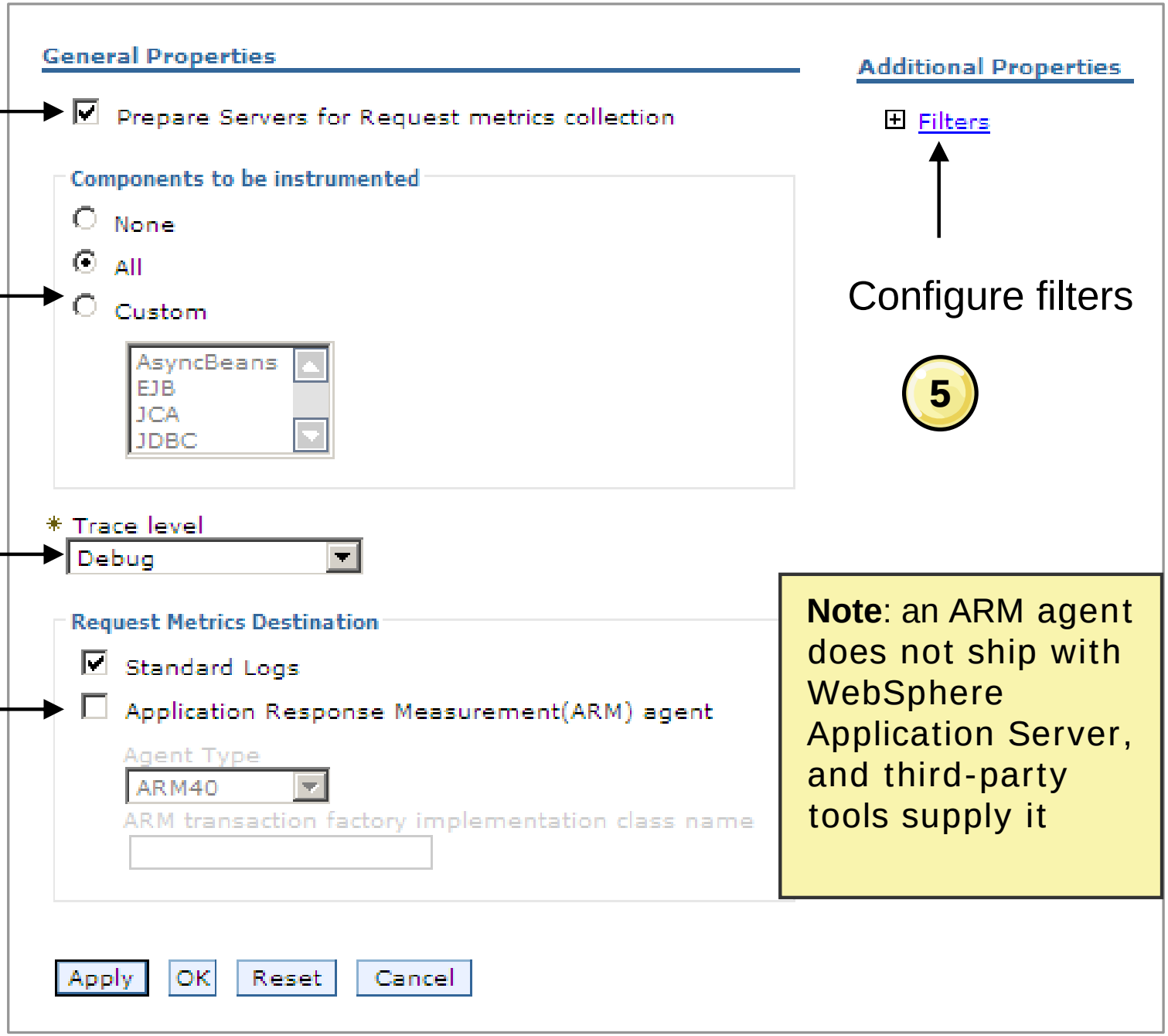
Request metrics (RM) overview

- 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



Enabling request metrics collection

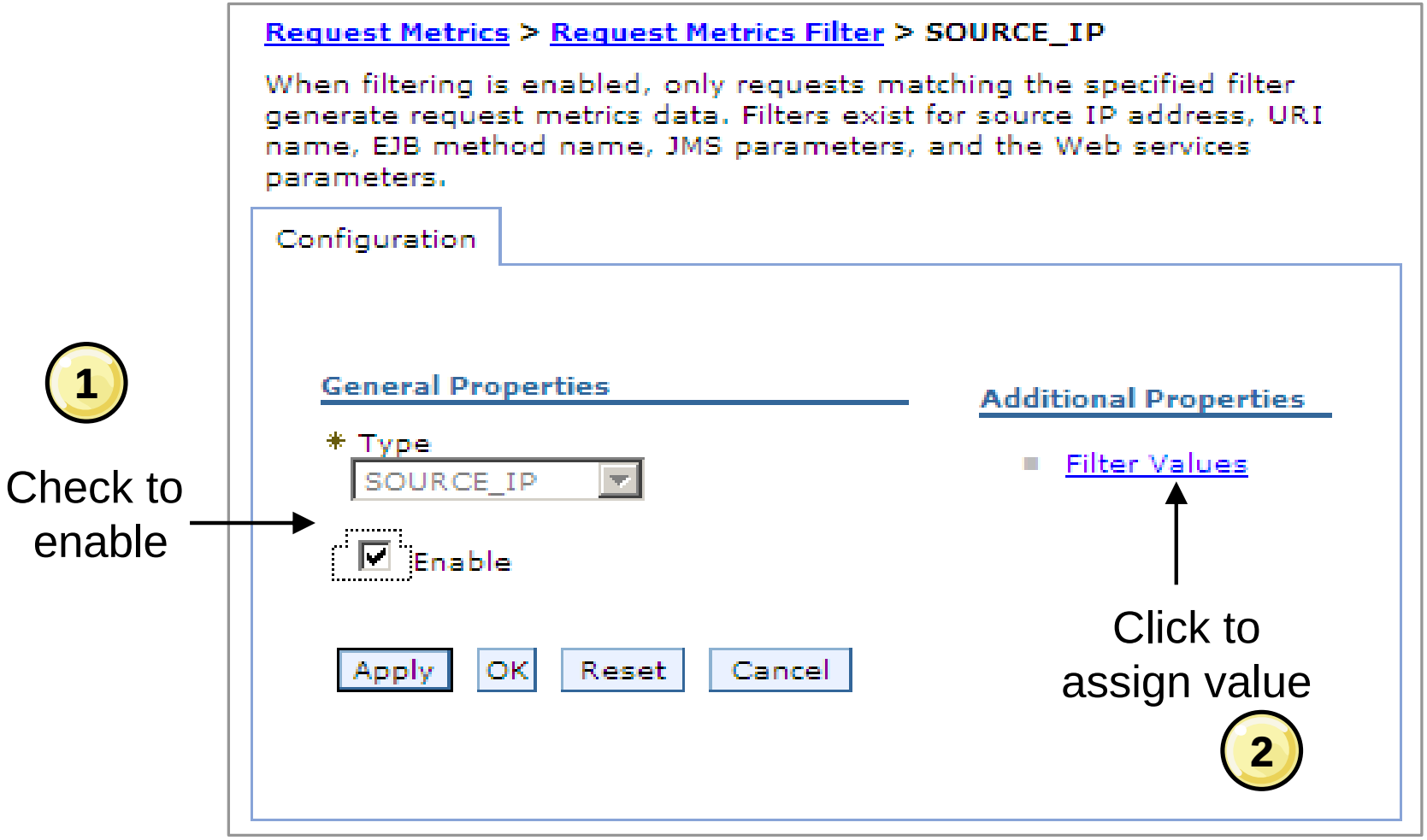
1. Check to enable
2. Select components
3. Select trace level
4. Choose output method
5. Configure filters



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

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



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
```

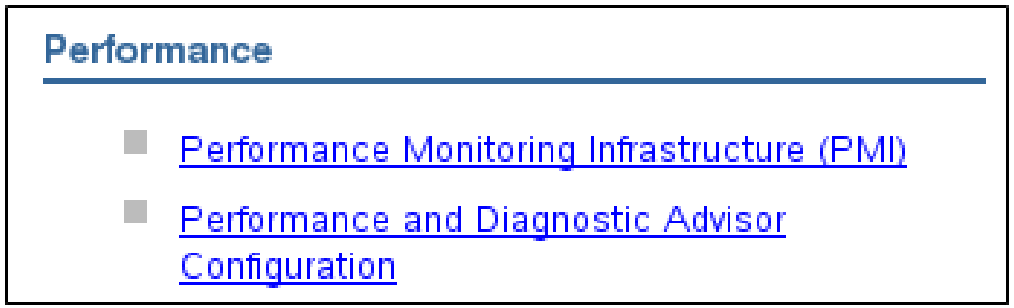
Performance advisors

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

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

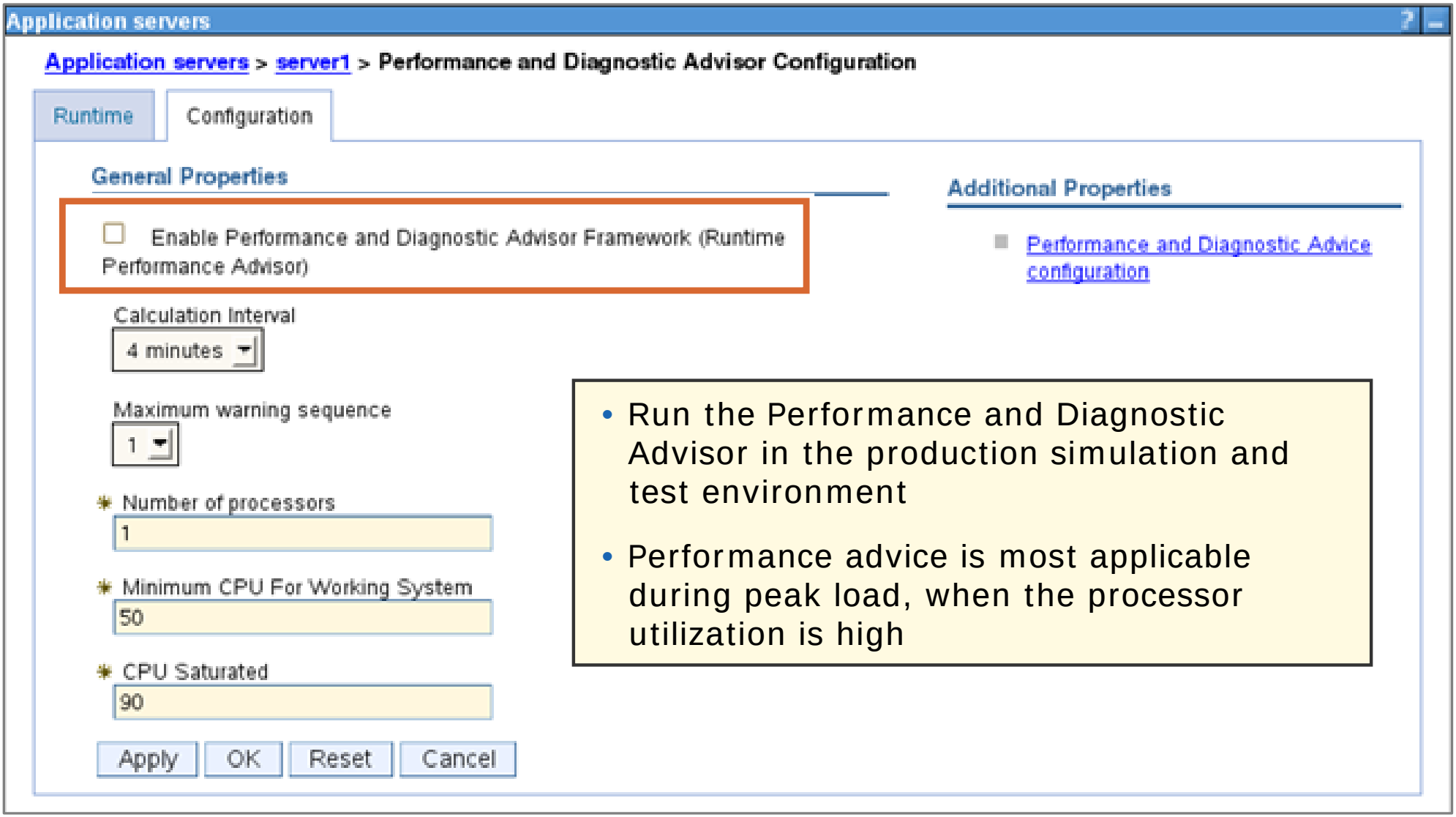


- Diagnostic advice:
  - Connection factory diagnostic messages
  - Data source diagnostic messages

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

Performance and Diagnostic Advisor (2 of 5)

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



- 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

Performance and Diagnostic Advisor (3 of 5)

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

Start 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	

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

Performance and Diagnostic Advisor (5 of 5)

Runtime Events

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.

Tivoli Performance Viewer advisor

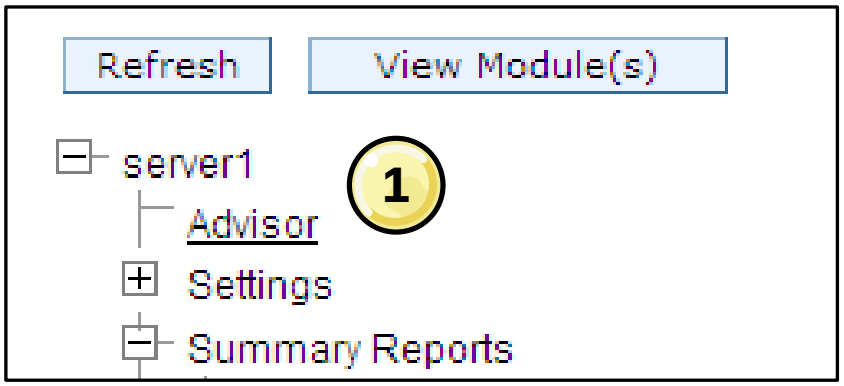
- Performance advice:
  - ORB service thread pools
  - Web container thread pools
  - Connection pool size
  - Persisted session size and time
  - Prepared statement cache size
  - Session cache size
  - Dynamic cache size
  - Java virtual machine (JVM) heap size
  - DB2 Performance Configuration wizard

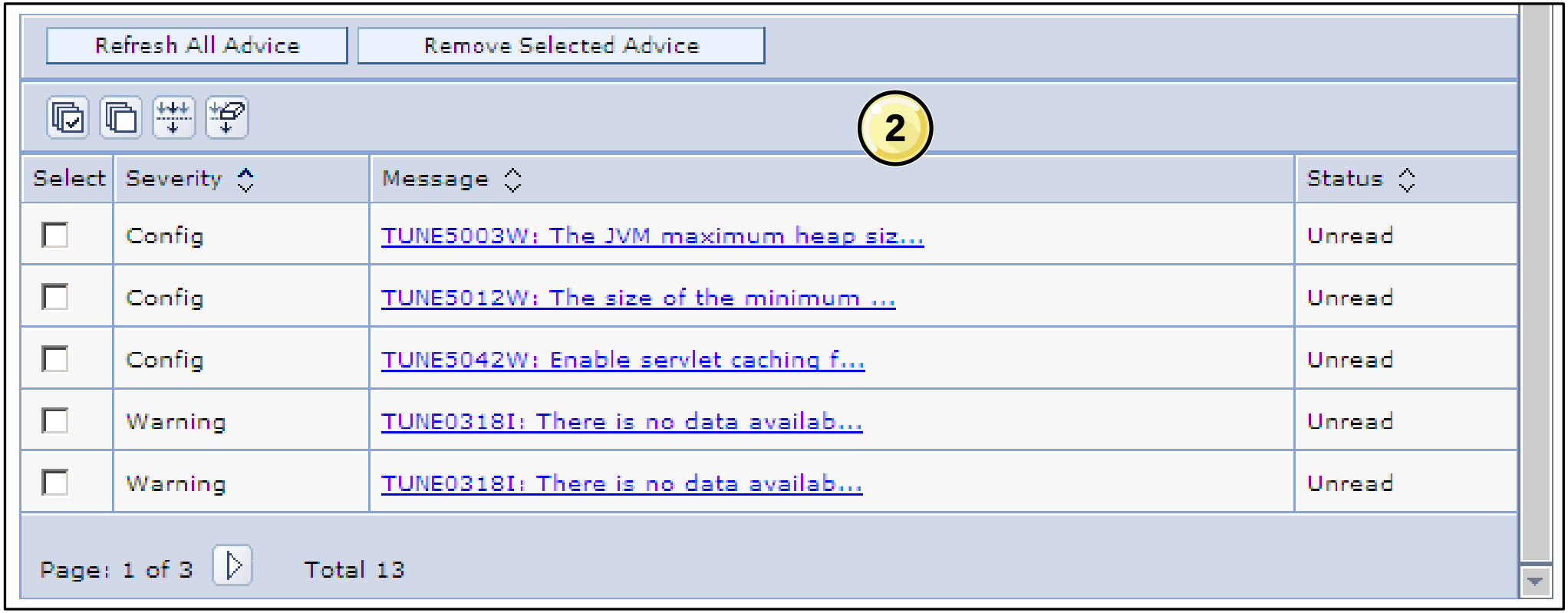
Examples of performance advice

- 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

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





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.

Back

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

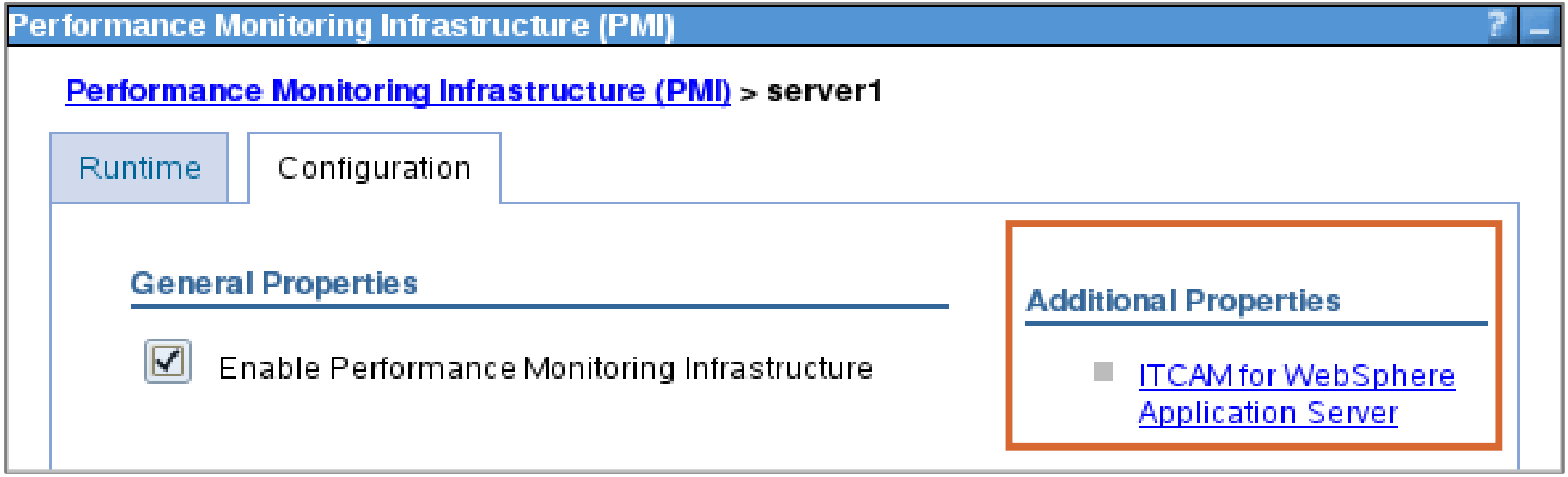
IBM Tivoli Composite Application Manager for WebSphere Application Server

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

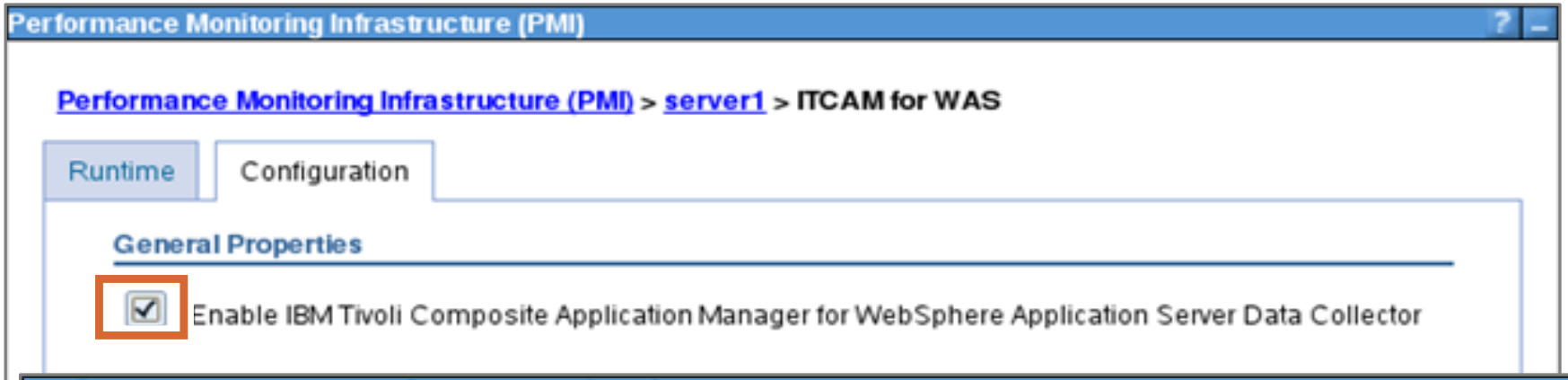
IBM Tivoli Composite Application Manager for WebSphere Application Server

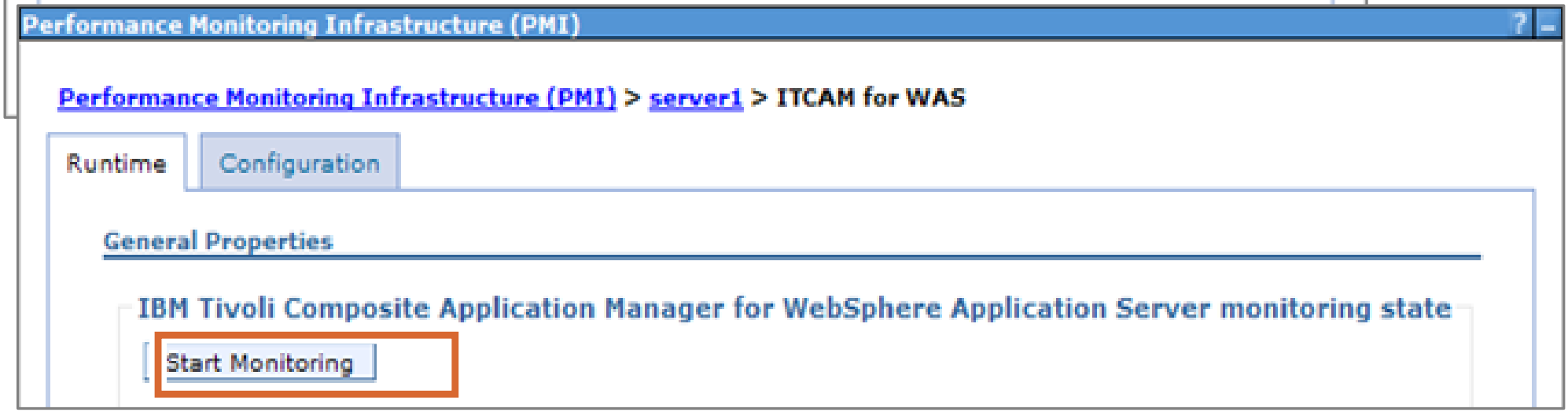
- Data collector available in WebSphere Application Server V8.5 as an extension offering (optional download and installation)
- IBM Tivoli Composite Application Manager for WebSphere Application Server is a separate installation
  - Installed by using the IBM Installation Manager
  - Configure one or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



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**





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

Reset To Zero | Clear Buffer | View Table | Show Legend

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

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

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

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.

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.

Exercise 16

Using the performance monitoring tools

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