IBM WEBSPHERE 5.0 SKILLS TRANSFER - LAB EXERCISE

Distributed System Management / Workload Management

What This Exercise is About

Workload Management owes a lot of it's changes to the changes in the system management infrastructure. Multiple application servers can now be federated into cells for the purposes of administration. However, cells are not the final configuration for Workload Management. In this lab, we will explore clusters - application servers grouped for the purpose of workload management.

User Requirement

This lab assumes a Network Deployment installation - you should have the Deployment Manager installed and a node federated.

You should have completed the Installation lab - specifically, Part Two, where the configuration is backed up.

> **Note:** If you did not complete the Installation lab, navigate to **C:\WebSphere\AppServer\bin** and run the command **backupConfig baseWASnode.zip**

What You Should Be Able to Do

When you are finished with this lab, you will have created a cluster of servers and installed an application to that cluster. You will see how WebSphere handles failover, and you will be able to visualize the system configuration. You will also use the Tivoli Performance Viewer to watch the servers handling servlet requests.

Introduction

When an application server process fails or is stopped, it is often important to maintain session information such that end users do not see any change. Persisting session information in memory is one way to accomplish this. The following instructions will guide you through setting up a cluster of application servers with in-memory session replication.

Exercise Instructions

> Throughout these instructions, the node name and machine name are assumed to be the same. Since each installation is different, **<NodeName>** will represent the name of your node and the host name of your machine.

Part One - Setting up the environment

__1. Check that the Deployment Manager is running by entering **serverStatus dmgr** from **C:\WebSphere\DeploymentManager\bin**. If it is not running, start it by running **startManager** from the same directory.

__2. Check that the Node Agent is running by entering **serverStatus nodeagent** from **C:\WebSphere\AppServer\bin**. If it is not running, start it with **startNode**.

Part Two: Setting up a cluster

__1. Open the Admin Console.

__ a. In a Browser, go to http://<System A>:9090/admin .

__2. Log in as **wsdemo**. (Note: The name you enter is only used for tracking changes, so any name will work.)

__3. Open the **Servers** item in the Task Navigation frame.

__4. Click on **Clusters**.

__5. In the Work Area frame, Click on **New**.

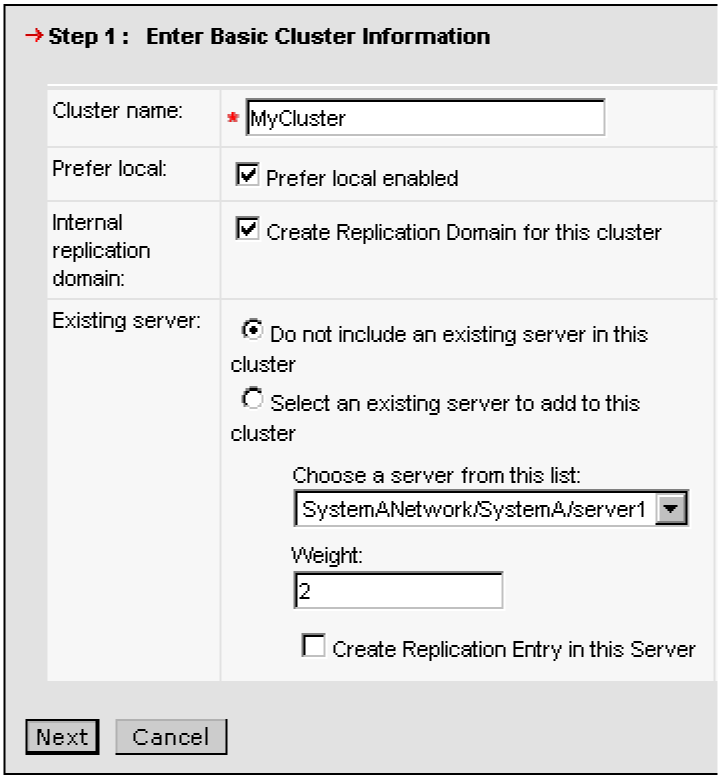
__6. In the **Step 1** panel:

__ a. Type **MyCluster** for a cluster name.

__ b. Verify that **Prefer Local Enabled** is checked.

__ c. Select **Create Replication Domain for this cluster**.

__ d. Verify that **Do not include an existing server in this cluster** is selected.



→ Step 1 : Enter Basic Cluster Information

Cluster name:	* MyCluster
Prefer local:	☑ Prefer local enabled
Internal replication domain:	☑ Create Replication Domain for this cluster
Existing server:	● Do not include an existing server in this cluster ○ Select an existing server to add to this cluster Choose a server from this list: SystemANetwork/SystemA/server1 Weight: 2 ☐ Create Replication Entry in this Server

__ e. Click **Next**.

__7. In **Step 2: Create New Clustered Servers** panel:

__ a. Type **WLM_Server1** for a server name.

__ b. Select **<NodeName>** for **Select Node**.

__ c. Type **2** for a Weight.

__ d. Verify that **Generate Unique Http ports** is selected.

__ e. Select **Create Replication Entry in this server**.

__ f. Select **Existing application server**.

__ g. Select **<NodeName>Network/<NodeName>/server1** for a template.

__ h. Click **Apply**.

__8. Still in **Step 2: Create New Clustered Servers** panel:

__ a. Type **WLM_Server2** for a server name.

__ b. Select **<Nodename>** for **Select Node**.

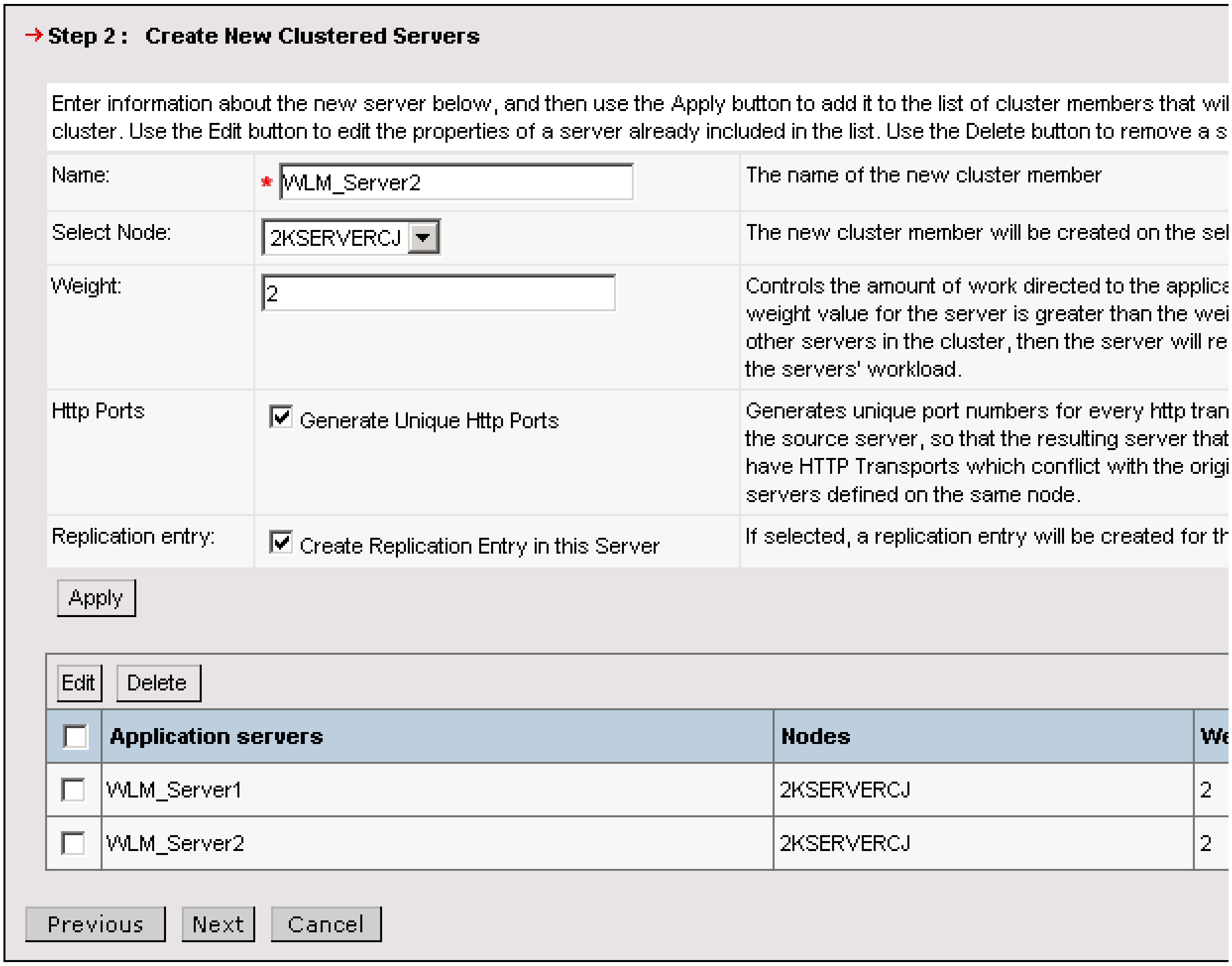
__ c. Type **2** for a Weight.

__ d. Verify that **Generate Unique Http ports** is selected.

__ e. Select **Create Replication Entry in this server**.

__ f. Click **Apply**.

__9. Your configuration should now look something like this:



→ Step 2 : Create New Clustered Servers

Enter information about the new server below, and then use the Apply button to add it to the list of cluster members that wil
cluster. Use the Edit button to edit the properties of a server already included in the list. Use the Delete button to remove a s

Name:	* WLM_Server2	The name of the new cluster member
Select Node:	2KSERVERCJ ▼	The new cluster member will be created on the sel
Weight:	2	Controls the amount of work directed to the applica weight value for the server is greater than the wei other servers in the cluster, then the server will re the servers' workload.
Http Ports	☑ Generate Unique Http Ports	Generates unique port numbers for every http tran the source server, so that the resulting server that have HTTP Transports which conflict with the origi servers defined on the same node.
Replication entry:	☑ Create Replication Entry in this Server	If selected, a replication entry will be created for th

□	Application servers	Nodes	We
□	WLM_Server1	2KSERVERCJ	2
□	WLM_Server2	2KSERVERCJ	2

__10. Click **Next**.

__11. Review the configuration and click **Finish**.

__12. Add the port numbers 9082 and 9083 to the default host aliases list. (**Note:** If you did not create Server 2 in the previous lab, use ports 9081 and 9082)

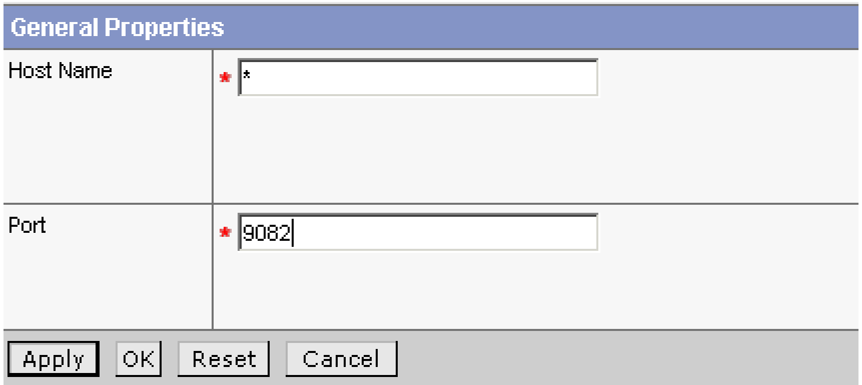
__ a. Under **Environment** click **Virtual hosts**.

__ b. Click **default host** in the Work-Area panel.

__ c. Scroll down and click **Host Aliases**.

__ d. Click **New**.

__ e. In the New window enter the following:
HostName: *****
Port: **9082**



General Properties

Host Name	* *
Port	* 9082

__ f. Click **OK**.

__ g. Scroll down and click **Host Aliases** again. Click **New** to add port number 9083 to the host aliases list

__ h. In the New window, enter the following:
HostName: *****
Port: **9083**

__ i. Click **OK**.

__13. Save your configuration.

__ a. On the Task bar at the top of the admin console, click **Save**.

__ b. In the Work Area frame, expand and review the list of files to be changed.

__ c. Click **Save**.

Part Three: Installing an Application

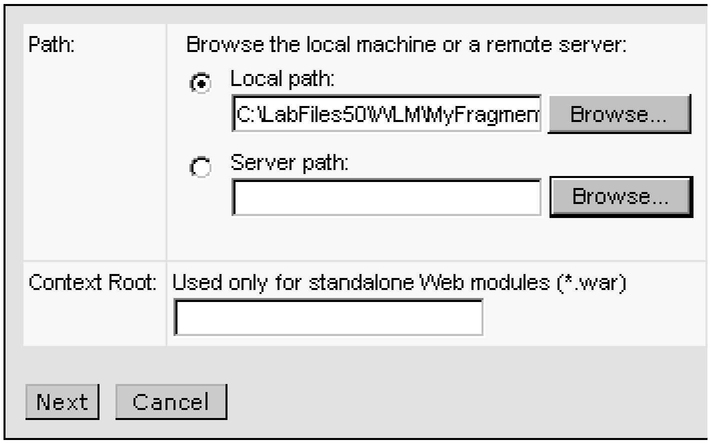
__1. In the Task Navigation frame, expand **Applications**.

__2. Click on **Install New Application**.

__3. Select the application **C:\LabFiles50\WLM\MyFragment.ear**.

__ a. Select **Local Path**.

__ b. Browse to **C:\LabFiles50\WLM** and select **MyFragment.ear**.



Path: Browse the local machine or a remote server:

● Local path: C:\LabFiles50\WLM\MyFragmen

○ Server path:

Context Root: Used only for standalone Web modules (*.war)

__ c. Click **Next**.

__4. Change nothing on the next page (the page starts with **Generate default bindings**) and click **Next**.

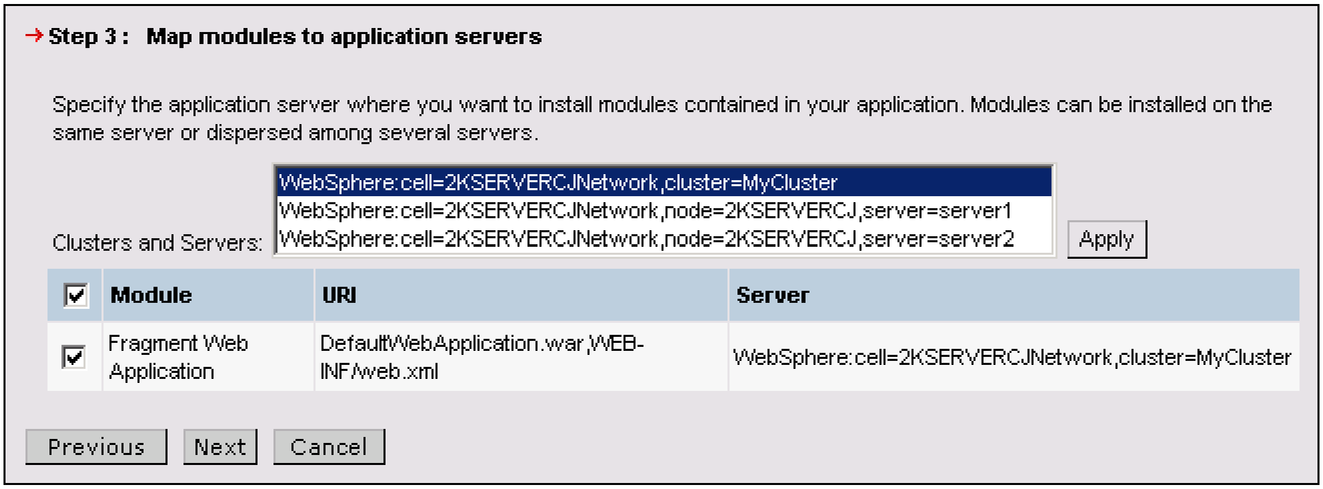
__5. Click **Step 3 Map modules to application servers**.

__ a. Notice that the target server for installation is server1.

__ b. Select **MyCluster** as shown in the screen capture below.

__ c. Select the checkbox next to **Fragment Web Application**.

__ d. Click **Apply**. Note that the server changes from **WebSphere:cell=<NodeName>Network,node=<NodeName>,server=server1** to **WebSphere:cell=<NodeName>Network,cluster=MyCluster**.



→ **Step 3 : Map modules to application servers**

Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Clusters and Servers:

WebSphere:cell=2KSERVERCJNetwork,cluster=MyCluster
WebSphere:cell=2KSERVERCJNetwork,node=2KSERVERCJ,server=server1
WebSphere:cell=2KSERVERCJNetwork,node=2KSERVERCJ,server=server2

☑	Module	URI	Server
☑	Fragment Web Application	DefaultWebApplication.war,WEB-INF/web.xml	WebSphere:cell=2KSERVERCJNetwork,cluster=MyCluster

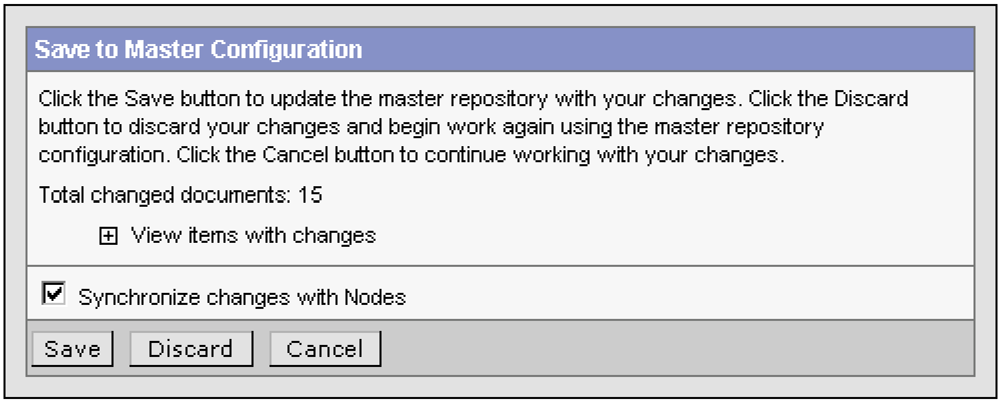
__6. Click **Step 5 Summary** and view the install options.

__7. Click **Finish**.

__8. You will get several progress messages, ending with:

Application MyFragment installed successfully

__9. **Save** the configuration, with the **Synchronize changes with Nodes** box checked.



Save to Master Configuration

Click the Save button to update the master repository with your changes. Click the Discard button to discard your changes and begin work again using the master repository configuration. Click the Cancel button to continue working with your changes.

Total changed documents: 15

⊞ View items with changes

☑ Synchronize changes with Nodes

__10. Navigate to **C:\WebSphere\AppServer\installedApps\<NodeName>Network** and verify that the application binaries (MyFragment.ear directory) were copied there. This may take 30 to 45 seconds for the files to copy and for Windows to update the folder views. Automatic distribution of application binaries is new in WebSphere 5.0.

Part Four: Configuring the cluster members

Note: These steps will have to be done for each cluster member. Repeat these steps twice - for WLM_server1 and for WLM_server2.

__1. In the admin console, on the Task Navigation panel, click on **Servers**, then on **Application Servers**.

__2. Click on **WLM_server** N, then **Web Container**, then **Session Management**.

__3. Click **Distributed Environment Settings**.

__4. Select **Memory to Memory Replication** (click on the radio button AND on the link).

__5. From the first dropdown, verify that **MyCluster** is selected.

__6. From the second dropdown, verify that the replicator name matches the server name - WLM_Server1 or WLM_Server2. (The replicator is created with the same name as the cluster member.)

__7. Click **OK**.

__8. Click on **Custom Tuning Parameters**

__9. Click on **Medium**

__10. Click **OK**

__11. After repeating steps 1 through 10 for both of the cluster members, remember to **Save**.

Part Five: Starting the Cluster

__1. In the Administrative Console's Task Navigation frame, click on **Servers**.

__2. Click **Clusters**.

__3. Select the box by **MyCluster**.

__4. Click **Start**. This will take a few minutes; while the servers are starting, continue with the instructions.

__5. Re-generate the plug-in.

__ a. Click on **Environment**, then **Update Web Server Plugin**, then **OK.** The plugin will be created in the DeploymentManager\config directory. The last line in the Web Server's configuration file (C:\IBMHTTPServer\conf\httpd.conf) points to where the Plugin expects to find the configuration file -

WebSpherePluginConfig "C:\WebSphere\AppServer/config/cells/plugin-cfg.xml"

__6. The plugin-cfg.xml file needs to be in a location where the HTTP server plug-in will read it. Select only **one** of the following three options:

__ a. You can right-click on the link that says View or download the current web server plugin configuration file and select "**Save Link As...**" (Netscape) or "**Save Target As...**" (Internet Explorer) and save the file to **C:\WebSphere\AppServer\config\cells\plugin-cfg.xml**.

Or you could manually copy **C:\WebSphere\DeploymentManager\config\cells\plugin-cfg.xml** to **C:\WebSphere\AppServer\config\cells\plugin-cfg.xml**.

Or you could edit the last line in the C:\IBMHTTPServer\conf\httpd.conf file to point to the directory where the plugin is generated - C:\WebSphere\DeploymentManager/config/cells/pugin-cfg.xml.

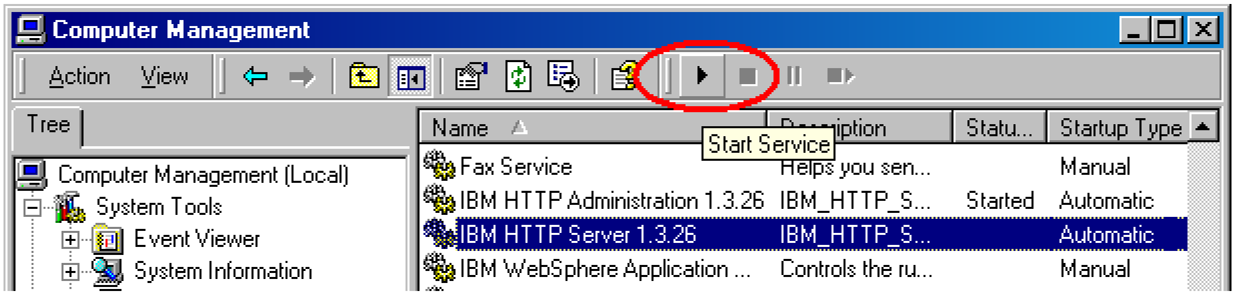
Note: The last option has the advantage of pointing the plug-in at the location where plugin-cfg.xml is created, and therefore, future changes to the plug-in will be read by the HTTP server without copying files anywhere. In a production environment, where the HTTP server is on a separate physical box, this method is not always practical, and manual copying of files is the next best solution.

__7. Start up the HTTP server.

__ a. Right-click on **My Computer**, then click **Manage**.

__ b. Double-click **Services and Applications**, then **Services**.

__ c. Click on **IBM HTTP Server 1.3.26**, then click the **Start** button.

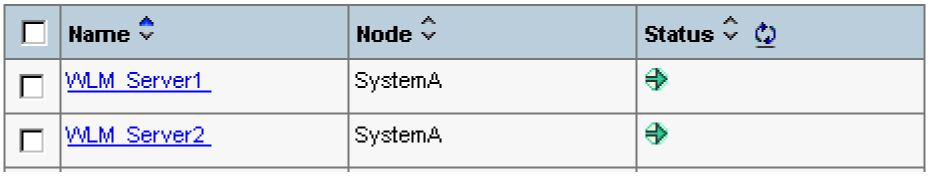


Name	Description	Statu...	Startup Type
Fax Service	Helps you sen...		Manual
IBM HTTP Administration 1.3.26	IBM_HTTP_S...	Started	Automatic
IBM HTTP Server 1.3.26	IBM_HTTP_S...		Automatic
IBM WebSphere Application ...	Controls the ru...		Manual

__8. Check that the cluster has started.

__ a. In the Admin Console, click on **Servers**, then **Application Servers**

__ b. If the cluster has finished starting, all the cluster members will have green arrows.



□	Name	Node	Status
□	WLM_Server1	SystemA	➜
□	WLM_Server2	SystemA	➜

Part Six: Workload Management: Failover in Action

In order to demonstrate failover, you will need to identify a particular server by PID number in the Windows Task Manager, and kill that process. This is **not** normal operating procedure!

__1. Open a new Browser and point it to http://<NodeName>/Fragment/HitMe.jsp.

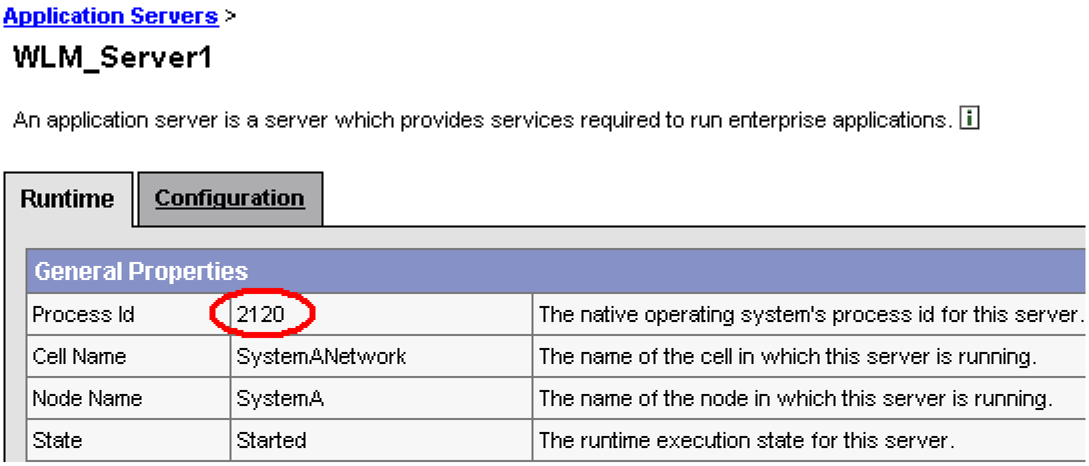
__2. Click **Increment** three or four times to see what server is serving your requests.

__3. Locate the PID of the server that serviced your request.

__ a. In the Admin Console, open **Servers** and click on **Application Servers**.

__ b. Click on the server that processed the request - (These instructions assume WLM_server1, but it could be any of the configured cluster members.)

__ c. Click on the **Runtime** tab and note the Process ID.



Application Servers >

WLM_Server1

An application server is a server which provides services required to run enterprise applications.

General Properties		
Process Id	2120	The native operating system's process id for this server.
Cell Name	SystemANetwork	The name of the cell in which this server is running.
Node Name	SystemA	The name of the node in which this server is running.
State	Started	The runtime execution state for this server.

__4. Using the Task Manager, end that java process. You can right-click an empty area of the taskbar, and then click Task Manager. On the Processes tab, click on the java.exe line that has the PID number that matched the Process ID of the application server. Then click End Process

__5. In the web browser, click **Increment** again.

__6. You should see that the counter moves correctly to the next value (after a brief pause), but the server name changes to reflect that a new application server has handled your request. The fact that the integer value incremented correctly indicates that the session information was communicated between application servers.

Part Seven: Viewing Performance Information (Optional)

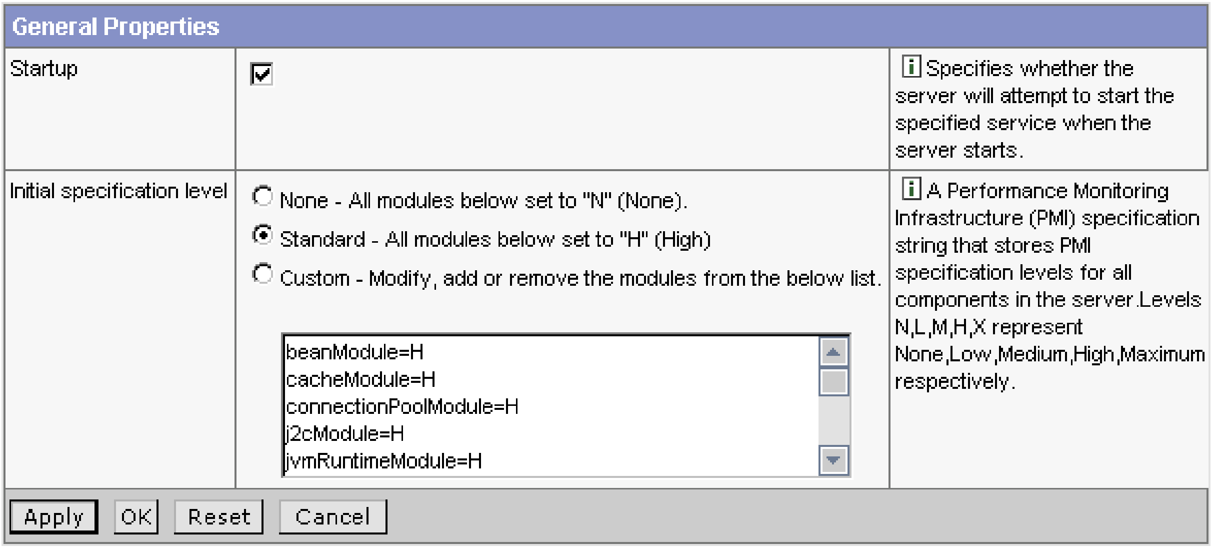
> Note: If you elect to bypass Part Seven, please proceed to Part Eight to restore your computer configuration to pre-WLM lab status.

__1. Enable monitoring on an application server.

__ a. In the Admin Console, open **Servers**, then click **Application Servers**, then **WLM_Server1**.

__ b. Click on **Performance Monitoring Service**.

__ c. Select **Startup** and set the initial specification level to **Standard**.



General Properties

Startup	☑	Specifies whether the server will attempt to start the specified service when the server starts.

Initial specification level	○ None - All modules below set to "N" (None).
	● Standard - All modules below set to "H" (High)
	○ Custom - Modify, add or remove the modules from the below list.
	beanModule=H
	cacheModule=H
	connectionPoolModule=H
	j2cModule=H
	jvmRuntimeModule=H
		A Performance Monitoring Infrastructure (PMI) specification string that stores PMI specification levels for all components in the server.Levels N,L,M,H,X represent None,Low,Medium,High,Maximum respectively.

__ d. Click **OK.**

__ e. **Save** the changes.

__2. Stop (if needed) and restart WLM_Server1 to start performance monitoring.

__3. From a Command prompt, navigate to **C:\WebSphere\AppServer\bin**.

__4. Type **tperfviewer** and press **Enter**. Wait several seconds for the JVM to start. There will be a warning that PMI Service is not started for the other servers. Click **OK**.

__5. When the Tivoli Performance Viewer launches, it will connect to the Deployment Manager.

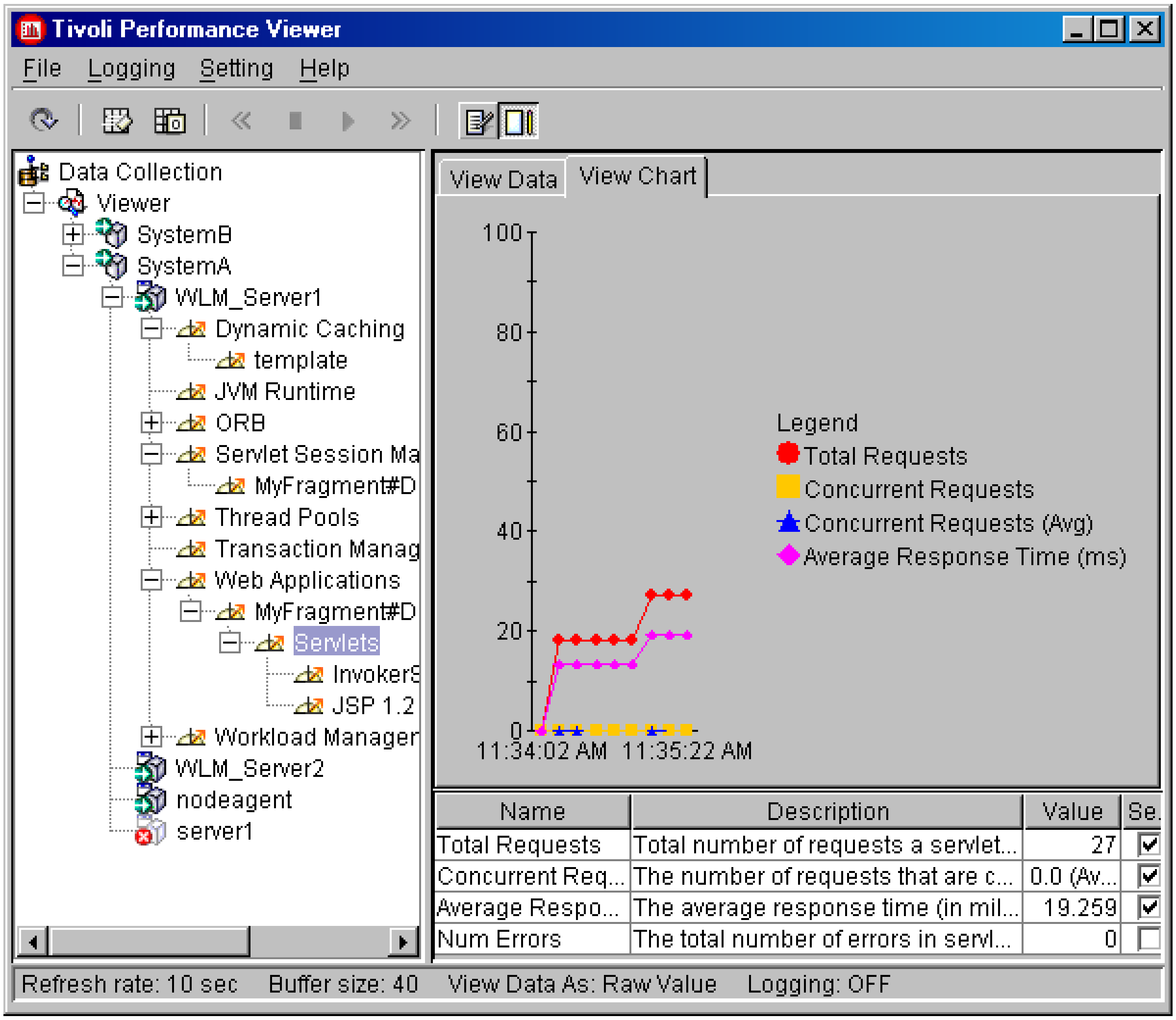
__6. Drill down to WLM_Server1's servlets as illustrated (See graphic for Step 9.).

__7. In the data panel, select the **View Chart** tab.

__8. From a browser, go to http://<NodeName>/Fragment/HitMe.jsp. Click on **Increment** several times.

__9. If the server is **not** WLM_Server1, you will need to open new browser instances and repeat the previous step until the round-robin routing algorithm sends your request to WLM_Server1.

__10. Notice the servlet response time and the number of invocations in the graph.



Name	Description	Value	Se.
Total Requests	Total number of requests a servlet...	27	☑
Concurrent Req...	The number of requests that are c...	0.0 (Av...	☑
Average Respo...	The average response time (in mil...	19.259	☑
Num Errors	The total number of errors in servl...	0	□

Refresh rate: 10 sec   Buffer size: 40   View Data As: Raw Value   Logging: OFF

__11. Browse around and explore the Performance Viewer options.

__12. Close the Tivoli Performance Viewer. (Do not save the monitor settings.)

Part Eight: Restoring the original configuration

__1. Using the Admin console, stop all application servers and cluster members.

__2. Click Logout to exit the Administrative Console.

__3. Remove the node:

- __ a. Open a command window and navigate to **C:\WebSphere\AppServer\bin**.
- __ b. Type **Removenode** and hit Enter.
- __ c. Wait for the confirmation message.

__4. Stop the HTTP server from the Windows services panel.

__5. Stop the Deployment Manager process

- __ a. From a Command prompt, navigate to C:\WebSphere\DeploymentManager\bin
- __ b. Type **stopManager**

What you did in this exercise

You federated a second machine into the Deployment Manager, created a cluster, and installed an application to servers on a remote node. You used a fragment of the HitCount servlet from the default application to watch which servers were handling your sessions, and watched servlet request failover in action. Then you connected up the performance monitor to view servlet requests being handled by the servers.