Lab 16 – Using the performance monitoring tools

At the end of 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

Section 1: Verifying the environment

If you changed the maximum heap size for the deployment manager to a lower value to conserve system resources, you must restore the maximum heap size to the default setting. More memory is required because the Tivoli Performance Viewer runs inside the deployment manager.

Section 2: Enabling performance monitoring and setting user preferences

Before the Tivoli Performance Viewer can begin monitoring data, the performance monitoring service must be started. The monitoring service is turned on by default on the application server.

Section 3: Viewing servlet and web applications module data

In this section, you use the Tivoli Performance Viewer to generate and view performance metrics.

Section 4: Using the Tivoli Performance Viewer performance advisor

WebSphere Application Server includes a performance advisor, the Tivoli Performance Viewer advisor, which is accessed from inside Tivoli Performance Viewer. The Tivoli Performance Viewer advisor provides helpful tuning advice for various resources, cache size, JVM heap size, and more. The Tivoli Performance Viewer advisor also provides recommendations to address inefficient settings.

Section 5: Using request metrics

Request metrics log the time that is spent at major components of the application server, such as the web server plug-in, web container, EJB container, and more.

The request metric architecture differs from the Performance Monitoring Infrastructure (PMI). PMI provides information about average system resource usage, with no correlation between the data across different WebSphere components.

The request metrics tool tracks each individual transaction within WebSphere Application Server, recording the response time of the major components. Some of these response times include time in the web server or in the Enterprise JavaBeans (EJB) container. The collected information can be saved to log files or forwarded to an Application Response Measurement (ARM) agent.

Setting the trace level to **Debug** provides detailed instrumentation data, including response times for all intra-process servlet and Enterprise JavaBeans (EJB) calls. This trace level provides a fine level of detail on each method call.

Setting the trace level to **Hops** generates instrumentation information about process boundaries only (for example, a servlet request that comes from a browser or a web server, and a JDBC request that goes to a database).

You can also filter requests so that only specific incoming requests result in request metrics that are logged. Such filtering can keep the logs from being overloaded with request metrics for every request.