Enhancing the Transactional Performance and Reliability on IBM WebSphere Application Servers

Louie Ching
Tivoli Solution Architect
IBM Software Group
What is a Composite Application?

- **A Composite Application**
  - Consists of multiple application logic components
  - Is distributed over multiple host and/or server environments
  - Links together over multiple interface types
  - Services customer transactions

- **Composite Applications have one or more of the following properties**
  - Multi-tiered – Spans HTTP / J2EE / BI / Legacy / RDBMS / Web Services
  - Web Accessible and leverages J2EE as the key enabling technology
  - Interacts with Legacy or Back-Office systems (such as CICS, IMS, SAP) that act as authoritative data repositories
  - Requires Business Integration Middleware such as WMQ/ WBI Connectivity, Brokering, Transformation, Workflow
  - May span corporate organizational boundaries (particularly true with Service-Oriented Architectures)
Composite Applications Have Unique Needs

A Composite Web Application, Involving J2EE, Integration Middleware and Legacy Systems

- **Business processes increasingly depend on multi-tier composite applications**
  Composite applications use business logic and data from sources spanning multiple systems and transactional environments

- **Composite applications are difficult to design, build, test and manage for high performance and availability**
  Many applications are not meeting performance expectations in production
  Customers are dissatisfied with application downtime

- **Traditional management processes and tools exacerbate the problem**
  Application lifecycle processes and tools do not follow transactions end-to-end
  Unable to proactively recognize and prevent performance problems at the end user
The Impact of Composite Application Problems

<table>
<thead>
<tr>
<th>IT Challenge</th>
<th>Business Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority of IT problems are still being identified by end user complaints</td>
<td>Lost business when problems go undetected; impacts customer satisfaction</td>
</tr>
<tr>
<td>Composite application performance problems can take many hours to isolate</td>
<td>Support budget spent isolating problems instead of fixing them</td>
</tr>
<tr>
<td>Once problem is identified a large team of specialists is required to resolve</td>
<td>Large amount of money from both development and support spent solving problems</td>
</tr>
<tr>
<td>the problem</td>
<td></td>
</tr>
<tr>
<td>Most application problems go back to development group, which cannot</td>
<td>Development budget spent fixing problems instead of building new applications</td>
</tr>
<tr>
<td>recreate the problem</td>
<td></td>
</tr>
</tbody>
</table>
Three Dimensions of Effective Composite Application Management

Transactions

Service Level Response Times
Problem Isolation

Applications

Deep-Dive Diagnostics
Correlate Across Subsystems

Resource Monitoring

Transactional Server Monitoring
Automation and Take Action

- End user service level monitoring
- Follow transaction flows
- Isolate problems by component

- Diagnostics at the application & middleware level
- Application performance analysis
- Drill down to code level detail

- Performance monitoring of J2EE, CICS, MQ, IMS
- Application resource consumption analysis
- Visualize workload trends and tune environment
Our Unique Answer To The Challenge

Repeatable, integrated approach to identify, isolate, diagnose and take action to resolve composite application issues before they become a problem...

What is End-To-End Composite Application Management (CAM)?

- Reduce application downtime
- Resolve application problems quickly

**Transactions**

Service Level Response Times
Problem Isolation

**Applications**

Managed Transactions
Detailed Diagnostics

**Infrastructure**

Monitoring
Automation

- End user transaction response times
- Follow transaction flows
- Isolate problems by component

- Diagnostics at the application & middleware level
- Workload resource analysis
- Deep dive to code level detail

- Resource analysis of system infrastructure components
- Event notification
- Integration with TEC & TBSM
- Automated responses

- Context passed between the layers
- Seamless cross organizational hand-offs
What is Our Approach?

An integrated offering that provides:

End-To-End Application Management

End user perspective, in context application and resource analysis

Integrated Lifecycle Support

Consistent context from development to production

Integration for IT Process Optimization

Provides data and context for service level management, business views and policy orchestration

IBM will deliver the most strategic, comprehensive solution
Tivoli’s End-To-End Application Management Offerings

IBM Tivoli CAM for Transaction Tracking
- Identify/Isolate problems quickly
- End user service level monitoring
- Users: 1st Level Support; Operators; Application Help Desk

IBM Tivoli CAM for J2EE
- Diagnose hard to find problems
- Users: 2nd & 3rd Level Support; Application Support; WebSphere support; WebSphere administrators

IBM Tivoli Monitoring Solution
- Fix problems automatically
- System infrastructure component analysis and health monitoring

- **Context passed** between the layers
- **Seamless** cross organizational hand-offs
Who Are the Users?

Problem Resolution Process

1. Identify and Prioritize
2. Isolate and Assign
3. Diagnose and Config
4. Fix

Annette – L1 Operator/Help Desk
- Wants increased ability to fix problems or route accurately without increased skill level
- More efficient access to alert handling procedures

Bernie – L2 Application Support Engineer
- Responsible for Application Availability and Response Time
- Must isolate problem and assign to right SME quickly

Jim – L3 Subject Matter Expert
- Wants to receive clear & succinct domain-specific alerts
- Would like access to history and change data to diagnose and fix

Rich – Developer
- Responsible for Debugging Production Applications
- Wants access to data from the production system to quickly find the bug and fix it
Which Tools Do They Use?

Problem Resolution Process

Identify and Prioritize

Isolate and Assign

Diagnose and Config

Fix

Annette – L1 Operator/Help Desk
- IT CAM for Transaction Tracking

Bernie – L2 Application Support Engineer
- IT CAM for Transaction Tracking and IT CAM for J2EE

Jim – L3 Subject Matter Expert
- IT CAM for J2EE
- IBM Tivoli Monitoring

Rich – Developer
- IBM Rational Tools
IT CAM for TT  Features

- Passive Monitoring (ISOLATE)
  ARM instrumentation of
  - Web Servers
  - Web Application Servers
  - In-house or third party application
  High-level decomposition using a reverse proxy

- Active Monitoring (IDENTIFY)
  URL record and playback using Synthetic Transaction Investigator
  Windows client record and playback using Rational Robot GUI scripts
  Protocol record and playback using Rational Robot VU scripts

- Operations-friendly GUI
  Transaction Topology
  “Fast path to failure”
IT CAM for TT Identifies the Problems

Dashboard shows transactions with a availability or response time problem

Click to see response time trend

Identify and Prioritize
Isolate and Assign
Diagnose and Config
Fix
IT CAM for TT Isolates and Assigns the Problem

- Identify and Prioritize
- Isolate and Assign
- Diagnose and Config
- Fix

Topology shows problem system so that it is assigned to correct people
Transactions Decomposition View
Is the network the possible cause of transaction latency?
Are memory heap sizes indicating a leak?

Is performance trending towards a failure?

Can I easily identify if the problem is with WAS or elsewhere?
**Application View – All Currently Running Requests**

In the Search Request box, type the name of the request for which you are searching. If you leave this box empty, all active requests will display.

<table>
<thead>
<tr>
<th>Search Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
</tr>
<tr>
<td>Server</td>
</tr>
<tr>
<td>Search Request/Transaction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search Results</th>
<th></th>
<th></th>
<th>Start Date/Time</th>
<th>Thread/Task ID</th>
<th>Total Resident Time (ms)</th>
<th>User ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Name</td>
<td>Client Request/Transaction</td>
<td>Timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADCDPL.M2L2.M2L2 : maven (L3)</td>
<td>jms/MSP/homeBookServiceWeb扬册YourBrother</td>
<td>Oct 8, 2004 11:29:40 AM</td>
<td>758027208</td>
<td>22214</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>network1:ADCDPL.M2L2.M2L2 : 1729.36 (L3)</td>
<td>IVTNQ</td>
<td>Oct 8, 2004 11:29:41 AM</td>
<td>19C9B06B-759B-BBF047753B23400</td>
<td>21981</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

- **Search for a specific request**
- **Select Request with Longest Resident Time**
- **Click “Server Activity Display”**

“In-Flight Request Search” can be narrowed down with user-defined criteria. This display is sorted by resident time, longest first. Drill down further by clicking the AAD button (“Application Activity Display”)

*Image and text from Tivoli software.*
Thread View – Detail on a Selected Thread

- Details for the chosen thread are displayed
- Execute a Stack Trace or Method Trace on the Left Nav to determine next course of action

Click "Stack Trace"
### Stack Trace

The Stack Trace shows all the methods that have not completed execution, including the class name and stack depth of each method.

<table>
<thead>
<tr>
<th>Depth</th>
<th>Class</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>java.lang.Thread</td>
<td>stack</td>
</tr>
<tr>
<td>1</td>
<td>com.ibm.ws.webcontainer.servlet.Servlet</td>
<td>process</td>
</tr>
<tr>
<td>2</td>
<td>com.ibm.ws.webcontainer.servlet</td>
<td>doPostAction</td>
</tr>
<tr>
<td>3</td>
<td>com.ibm.ws.webcontainer.servlet</td>
<td>doGet</td>
</tr>
<tr>
<td>4</td>
<td>java.servlet.http.HttpServlet</td>
<td>service</td>
</tr>
<tr>
<td>5</td>
<td>java.servlet.http.HttpServlet</td>
<td>service</td>
</tr>
<tr>
<td>6</td>
<td>com.ibm.ws.webcontainer.servlet.ServletInstance</td>
<td>doPostService</td>
</tr>
<tr>
<td>7</td>
<td>com.ibm.ws.webcontainer.servlet.Servlet</td>
<td>doPostService</td>
</tr>
<tr>
<td>8</td>
<td>com.ibm.ws.webcontainer.servlet.Servlet</td>
<td>doPostService</td>
</tr>
<tr>
<td>9</td>
<td>com.ibm.ws.webcontainer.servlet.Servlet</td>
<td>doPostService</td>
</tr>
<tr>
<td>10</td>
<td>com.ibm.ws.webcontainer.servlet.Servlet</td>
<td>doPostService</td>
</tr>
<tr>
<td>11</td>
<td>com.ibm.ws.webcontainer.servlet.Servlet</td>
<td>doPostService</td>
</tr>
<tr>
<td>12</td>
<td>com.ibm.ws.webcontainer.servlet.servletServlet</td>
<td>doPost</td>
</tr>
<tr>
<td>13</td>
<td>com.ibm.ws.webcontainer.webapp.ServletRequestDispatcher</td>
<td>forward</td>
</tr>
<tr>
<td>14</td>
<td>com.ibm.ws.webcontainer.webapp.ServletRequestDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>15</td>
<td>com.ibm.ws.webcontainer.webapp.ServletRequestDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>16</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>17</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>18</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>19</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>20</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>21</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>22</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>23</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>24</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>25</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>26</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>27</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>28</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>29</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
<tr>
<td>30</td>
<td>com.ibm.ws.webcontainer.servlet.WebAppDispatcher</td>
<td>dispatch</td>
</tr>
</tbody>
</table>
“Method Trace” lists the method information for uncompleted requests.
Return to “Request Details” to take action against the request.
The request is cancelled

Thread Status and Priority may also be changed from this screen

**REQUEST DETAIL**
The Request Detail page provides data for one thread only. Use the tools to trace a stack, method, or request and session of a thread; cancel a request; or change the thread's priority or status.

**REQUEST PROPERTIES**
- Snapshot Date: Dec 22, 2004
- Snapshot Time: 6:54:24 PM
- Platform CPU % Utilization: 16.00%
- Application Server Name: [Details]
- Application Server IP Address: [Details]

**REQUEST DETAIL**

<table>
<thead>
<tr>
<th>Thread ID</th>
<th>Last Known CPU</th>
<th>Last Known Method</th>
<th>Last Known Action</th>
<th>Last Known Class</th>
<th>Thread Status</th>
<th>Thread Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>709446712</td>
<td>3.803 ms</td>
<td>processRequest</td>
<td>UNKNOWN</td>
<td>comtestware/web</td>
<td>Waiting</td>
<td>Benefit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Cancel Request**

Click "Cancel Request"
“Decomposition Report” shows a further breakdown of “Trend Report”, according to the criteria selected ("Application Name", in this case).

Clicking on a data point provides more details.
The "Request Report Detail" opens.
Drilling-down is available through the "Trace Report".
Trace Report

The "Trace Report" provides details on any one record selected.
Modernizing Operations for On Demand

- **Integrate Application and Infrastructure management**
- **Extend Composite Application Management into Development**

![Diagram of Composite Application Management for On Demand]

- **Resource Monitoring Visualization**
- **Common Monitoring Foundation**
- **Application Context Visualization**

**IBM Tivoli Monitoring Family**

**Composite Application Management IT CAM Family**

**Development Tools & Process Rational Tools**
Aligning with IBM Application Life Cycle Management

**Application Governance**

- Rational Portfolio Manager
- Rational Professional Bundle
- Team Unifying Platform

**Composite Application Management**

- ITCAM for Transaction Tracking
- ITCAM for J2EE
- ITM

**Build, Run and Manage Software Solutions**

**Align with Strategies, Make Investment Decisions**
Tivoli’s On Demand Availability Management

Business Service Management

- Service level planning and reporting
- Application service and Infrastructure Mapping
- Business Impact Monitoring/Event Mgmt Prioritization

Composite Application Management

- Transaction simulation Performance Testing
- Response time segmentation and isolation
- Service Response Time Monitoring
- Application trace and diagnostics passed to developer tools
- Transaction decomposition Application Component Diagnostics
- Application resource usage and bottleneck analysis

Infrastructure Management

- Event Automation
- Policy Driven Orchestration
- Infrastructure Monitoring and Diagnostics
# Mapping CAM to the Tivoli Strategy

## Business Service Management
- IBM Tivoli Business Systems Manager

## Event Correlation and Automation
- IBM Tivoli System Automation
- IBM Tivoli NetView family
- IBM Tivoli Service Level Advisor
- IBM Tivoli Enterprise Console

## Composite Application Management
- IT CAM for TT
- IT CAM for J2EE

## Resource Monitoring
- IBM Tivoli Monitoring

## Distributed Systems
- IBM Tivoli Monitoring

## Host Systems
- IBM Tivoli Monitoring
Composite Application Management will deliver the most comprehensive solution for On Demand Service Delivery

An integrated set of offerings that provide:

- **End-To-End Composite Application Management**
  - End user perspective, in context application and resource analysis

- **Integrated Lifecycle Support**
  - Consistent context from development to production

- **Business processes metrics**
  - Provides data for service level management and business views