IBM Tivoli Workload Automation technical presentation

- Tivoli Workload Automation Portfolio
  - Workload Scheduler for Distributed Systems
  - Workload Scheduler for z/OS
  - Workload Scheduler for Applications
  - Workload Scheduler End-to-End Configuration
  - Dynamic Workload Broker

IBM Tivoli Workload Automation solution suite

- Single solution to integrate workloads from multiple applications, across multiple platforms
- Improve availability and integrity of production systems with built-in High Availability and Fault Tolerance
- Dynamic real-time workload and resource utilization optimization to maximize workload velocity into existing resources
- Integrate with systems mgmt solutions to drive business value.

Manage critical workloads by exception in broader systems management context

Start/stop/provision resources on demand
Tivoli Workload Automation: solutions architecture

- Tivoli Workload Automation components can be deployed into different solutions, to accommodate any customer need and organization requirement.
  - Maximum flexibility without feature limitations
  - Meets any customer organization requirements
- High Availability, Scalability and Fault Tolerance
- Any deployment scenario can leverage dynamic workload brokering
- All the different solutions can be monitored by a single Web-based user interface

Parallel Schedulers with Common Console

Mainframe-centric Configuration

Peer to Peer Schedulers with Common Console and Connected Agents

Distributed-centric Configuration

Tivoli Workload Automation: latest progress applicability

<table>
<thead>
<tr>
<th>Feature vs Setup</th>
<th>TWS z/OS</th>
<th>E2E</th>
<th>Distributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWS 8.3 for z/OS</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TWS 8.3 for z/OS 2007 SPE</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TWS 8.4 Tivoli Dynamic Workload Console</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TWS 8.4 Event-Driven Workload Automation</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TWS 8.4 TDWC Reporting</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tivoli Dynamic Workload Broker 1.2</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Integration with Tivoli Enterprise Portal / IBM Tivoli Monitoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TWS and WLM</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IBM Tivoli Workload Automation

- Tivoli Workload Automation Portfolio
  - Workload Scheduler for Distributed Systems
  - Workload Scheduler for z/OS
  - Workload Scheduler for Applications
  - Workload Scheduler End-to-End Configuration
  - Dynamic Workload Broker

Tivoli Workload Scheduler for Distributed Systems

- Centralized Modeling for distributed applications
  - Plan/organize every phase of applications execution

- Reliable Execution of distributed applications
  - Automate applications execution and dependencies resolution
  - Execution flow never run out of sequence
  - Minimize idle time and improve throughput

- Single point of Monitoring and Control
  - Monitor/Manage applications execution
  - Automate recovery process

- Interface to external applications
  - ERP (SAP R/3, PeopleSoft, Oracle E-Business Suite)
  - Customer built-in applications
Tivoli Workload Scheduler Distributed architecture (details)

Components

Each Component*
- Processes its own work
- Resolves its own dependencies
- Pushes work to its subordinates
- Resolves subordinate inter-dependencies
Fault Tolerant architecture

- Single Action Switch - GUI or CLI
- Manual or Automated
- Unlimited Backup Masters
- Unlimited Backup Domain Managers

Unattended synchronization using “store and forward”

What's new in Tivoli Workload Scheduler 8.3 Distributed

Leverage IBM infrastructure

- Scheduling repository on a real RDBMS
- Replace Tivoli Management Framework with WAS technology
- Advanced planning capabilities like TWS for z/OS
- J2EE WEB Services API
- New Web-based monitoring UI

Benefits

- Improve reliability, availability and scalability
- Provide more flexibility in building and running workload schedules
- Easily integrate with business process and third-party products through a J2EE programming API
- Reduce installation and maintenance costs