The Oracle BPEL Process Manager

BPEL Performance Tuning

Feature Preview Webinar Series

Kevin Clugage, kevin.clugage@oracle.com
Product Director

David Shaffer, david.shaffer@oracle.com
Director of Product Management, Oracle Integration

Francis Ip, francis.ip@oracle.com
Principal Solution Architect

---

Agenda

- Agenda
- Benchmarks
- Concepts
- Architecture
- Design
- Tuning
- Tools
- Case Study
Agenda

- Agenda
- Benchmarks
- Concepts
- Architecture
- Design
- Tuning
- Tools
- Case Study

### Benchmarks

<table>
<thead>
<tr>
<th>185 req/sec</th>
<th>BankTransferFlow</th>
<th>Sync. transient, Small request size, Dehydration OFF</th>
<th>2 CPU Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Retek Integration</td>
<td>•Simple batching process, DB adapter in/outbound</td>
<td>4 CPU Solaries</td>
<td></td>
</tr>
<tr>
<td>6 million DB rows/hr</td>
<td>Based on European Space Agency use case</td>
<td>Async. Durable, 5-10 steps, SOAP call-out, Dehydration ON</td>
<td>4 CPU Linux</td>
</tr>
<tr>
<td>3.2 Million req/day (2KB payload)</td>
<td>Based on a U.S. Gov. Agency use case</td>
<td>Sync. Transient, Small request size (1K), JMS-&gt;BPEL-&gt;EJB-&gt;DB, Dehydration OFF</td>
<td>2x 2CPU Linux</td>
</tr>
<tr>
<td>1.92 Million req/day (25KB payload)</td>
<td>Based on a U.S. Gov. Agency use case</td>
<td>Sync. Transient, Small request size (1K), JMS-&gt;BPEL-&gt;EJB-&gt;DB, Dehydration OFF</td>
<td>2x 2CPU Linux</td>
</tr>
</tbody>
</table>
Agenda

- Agenda
- Benchmarks
- Concepts
- Architecture
- Design
- Tuning
- Tools
- Case Study

Concepts

- Average response time per request
- Throughput: # requests/second
- Concurrency: average # of concurrent requests
Concepts

• Performance tuning objectives:
  • Minimize response time of single request
  • Identify the optimal point of concurrency
Common Bottlenecks

- Response time
  - Too much dehydration store activities
  - Large payload (> 100K)
  - Too many layers
- Concurrency:
  - Too few BPEL engine threads
  - Too few Adapter threads (adapter not moving messages to BPEL engine fast enough)
- Others:
  - Not enough connections to dehydration store database
  - JVM under utilize OS memory
  - Dehydration store database is slow

Agenda

- Agenda
- Benchmarks
- Concepts
- Architecture
- Design
- Tuning
- Tools
- Case Study
BPEL Performance Architecture

Process Types

- By interface
  - Asynchronous: one-way invocation
  - Synchronous: two-way invocation

- By durability
  - Durable process: long running
    - Has mid process breakpoint activities (Receive, onMessage, onAlarm, Wait)
    - Or, Has non-idempotent invoke
    - Or, Has non-blocking invoke
  - Transient process: short lived
    - No mid process breakpoint activities (Receive, onMessage, onAlarm, Wait)
    - And, No non-idempotent invoke
    - And, No non-blocking invoke
# Process Types

<table>
<thead>
<tr>
<th>Durable</th>
<th>Transient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Async.</td>
<td>• Common</td>
</tr>
<tr>
<td></td>
<td>• Example: LoanFlow</td>
</tr>
<tr>
<td></td>
<td>• One way invocation</td>
</tr>
<tr>
<td>Sync.</td>
<td>• Two way invocation</td>
</tr>
<tr>
<td></td>
<td>• Less Common</td>
</tr>
<tr>
<td></td>
<td>• Example: Customer case</td>
</tr>
<tr>
<td></td>
<td>• Common</td>
</tr>
<tr>
<td></td>
<td>• Example: Customer case</td>
</tr>
</tbody>
</table>

---

# Process Types

<table>
<thead>
<tr>
<th>Durable</th>
<th>Transient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Async.</td>
<td>• Common</td>
</tr>
<tr>
<td></td>
<td>• Example: LoanFlow</td>
</tr>
<tr>
<td></td>
<td>• Has mid process breakpoint activities (Receive, onMessage, onAlarm, Wait)</td>
</tr>
<tr>
<td></td>
<td>• Or, Has non-idempotent invoke</td>
</tr>
<tr>
<td></td>
<td>• Or, Has non-blocking invoke</td>
</tr>
<tr>
<td>Sync.</td>
<td>• Less Common</td>
</tr>
<tr>
<td></td>
<td>• Example: Customer case</td>
</tr>
<tr>
<td></td>
<td>• Common</td>
</tr>
<tr>
<td></td>
<td>• Example: CreditRatingService</td>
</tr>
</tbody>
</table>
**Process Types**

- Breakpoint activities
  - Receive
  - onMessage
  - onAlarm
  - Wait
- Idempotent invoke
  - idempotent = retry-able
  - BPEL engine dehydrates after non-idempotent activities
  - E.g. Assign is idempotent
  - Invoke is idempotent by default, can be set non-idempotent
- Non-blocking invoke
  - Means Engine perform invoke within FlowN in parallel
  - Default to be “blocking”, can be set non-blocking

**Async. Durable**

*(Example: LoanFlow)*

- Caller thread to receive message
- Separate threads to create/run instance
- Persist all messages
- Dehydrates “in flight” at breakpoints (multi txn)
Sync. Transient
(Example: CreditRatingService)

- Caller thread to receive message and create/run instance
- Not persist message
- Dehydrates once at the end (one txn)

Threading and Dehydration

<table>
<thead>
<tr>
<th>Durable</th>
<th>Transient</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Caller thread to receive message and create/run instance</td>
<td>• Caller thread to receive and persist incoming messages</td>
</tr>
<tr>
<td>• Separate &quot;worker-bean&quot; threads to create/run instance</td>
<td>• Separate &quot;worker bean&quot; threads to create/run BPEL instance</td>
</tr>
<tr>
<td>• Persist all messages</td>
<td>• Persist initial message</td>
</tr>
<tr>
<td>• Dehydrates “in flight” at breakpoints (multi txn)</td>
<td>• Dehydrates once at the end (one txn)</td>
</tr>
<tr>
<td>• Example: LoanFlow</td>
<td>• Example: HelloWorld</td>
</tr>
</tbody>
</table>

Async

- Call thread to receive message and create/run instance
- Separate “work-bean” threads to continue after breakpoints
- Not persist message for initial Receive, but persist for mid process Receive.
- Dehydrates “in flight” at breakpoints (multi txn)

Sync

- Call thread to receive message and create/run instance
- Never persist messages
- Dehydrates once at the end (one txn)
- Example: CreditRatingService
Threads and Dehydration

- Durable processes
  - Mandatory to save states
  - Save state “in flight” at breakpoints
- Transient processes
  - Optional to save state
  - Save state at the end

Agenda

- Agenda
- Performance Benchmarks
- Performance Concepts
- BPEL Performance Architecture
- Process Design Tips
- Tuning
- Tools
- Case Study
Process Design Tips

• Make async. transient -> sync. transient
  • Sync process does not persist incoming messages
  • Throw exception to client on error
• Combine many small processes into bigger ones
• Optimize transformation
  • Use Oracle XDK
    <copy>
    <from expression="ora:processXSLT('transform_1.xsl',
      bpws:getVariableData('input','payload'))"/>
    <to variable="input2" part="payload"/>
    </copy>
  • Use J2EE data source in DB lookup

Change to: xdk:processXSLT()
Tuning

• Rule #1: NO Oracle Lite for production or performance tests
  • Could be 10 times slower!
• Rule #2: If CPU and memory under utilized, increase concurrency
• Rule #3: If CPU and memory are already heavily utilized, reduce dehydration store activities
• Rule #4: Reduce audit trail and logging
Tuning Knobs

- By process types
  - For durable processes
  - For transient processes
  - For both
- By levels
  - Process level (bpe1.xml)
  - Domain level (BPEL Console)
  - J2EE container level (Ex: j2ee/OC4J_BPEL/config)
  - Database level (init.ora/spfile)
- By categories
  - Dehydration store activities
  - Concurrency
  - Misc.

Dehydration Store Knobs

- For all process types
  - auditLevel (domain level)
    - Control amount of audit events logged to database
    - Supported values: development, production, minimal, off
- For Transient types:
  - completionPersistPolicy (process/domain level)
    - Whether and when to save instance
    - Supported values: on, deferred, faulted, off
    - Must also set inMemoryOptimization=true in pre 10.1.3
  - completionPersistLevel (process/domain level)
    - Control amount of instance state saved to database
    - Supported values: all, instHeader
Process Level Dehydration Knobs

CreditRatingService/bpel.xml

```xml
<BPELSuitcase>
  <BPELProcess id="CreditRatingService" src="CreditRatingService.bpel">
    <partnerLinkBindings>
      <partnerLinkBinding name="client">
        <property name="wsdlLocation">CreditRating.wsdl</property>
      </partnerLinkBinding>
    </partnerLinkBindings>
    <configurations>
      <property name="inMemoryOptimization">true</property>
      <property name="completionPersistPolicy">faulted</property>
      <property name="completionPersistLevel">all</property>
    </configurations>
  </BPELProcess>
</BPELSuitcase>
```

Concurrency Knobs

<table>
<thead>
<tr>
<th></th>
<th>Durable</th>
<th>Transient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Domain</td>
<td>dspMaxThread</td>
</tr>
<tr>
<td></td>
<td>Container</td>
<td>Worker Bean threads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Invoker Bean threads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Datasource max con</td>
</tr>
</tbody>
</table>

Async

<table>
<thead>
<tr>
<th></th>
<th>Process</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Domain</td>
<td>dspMaxThread</td>
</tr>
<tr>
<td></td>
<td>Container</td>
<td>Worker Bean threads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Invoker Bean threads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Datasource max con</td>
</tr>
</tbody>
</table>

Sync

<table>
<thead>
<tr>
<th></th>
<th>Process</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Domain</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Container</td>
<td>Datasource max con</td>
</tr>
</tbody>
</table>
Concurrency Knobs

- dspMaxThreads: (domain level)
  - Throttle performance of processing asynchronous messages
  - Applicable to all process types except sync. transient
  - Total(all domains) <= work bean threads
Concurrency Knobs

- Worker Bean Threads (durable, container level)
  - Mid-tier installation:
    IAS_HOME/j2ee/OC4J_BPEL/application-deployments/orabpel.ejb_ob_engine.jar/orion-ejb.jar.xml
  - Developer edition:
    HOME/integration/orabpel/system/appsver/orabpel/j2ee/home/application-deployments/orabpel/ejb_ob_engine.jar/orion-ejb.jar.xml

```xml
<message-driven-deployment name="WorkerBean" destination-location="jms/collaxa/BPELWorkerQueue" connection-factory-location="jms/collaxa/BPELWorkerQueueFactory"
    listener-threads="70"
    min-instances="100">
    <ejb-ref-mapping name="ejb/local/DispatcherLocalBean" />
```

Concurrency Knobs

- Max Datasource Connections (both, container level)
  - Mid-tier installation:
    IAS_HOME/j2ee/OC4J_BPEL/config/data-sources.xml
  - Developer edition:
    HOME/integration/orabpel/system/appsver/orabpel/j2ee/home/config/data-sources.xml

```xml
<data-source class="com.evermind.sql.DriverManagerDataSource"
    name="BPELServerDataSource"
    location="jdbc/BPELServerDataSource"
    xa-location="BPELServerDataSource"
    ejb-location="jdbc/BPELServerDataSource"
    connection-driver="oracle.jdbc.OracleDriver"
    max-connections="125"
    min-connections="50"
    connection-retry-interval="30"
    max-connect-attempts="10"
    url="jdbc:oracle:thin:orabpel/orabpel@bpelhost:bpel:1521"/>
```
Other Knobs

- idempotent (durable, process level)
  - idempotent = retry-able
  - Engine dehydrates after non-idempotent activities

<table>
<thead>
<tr>
<th>true (default)</th>
<th>Engine will not dehydrate after invoke</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>Engine will dehydrate after invoke</td>
</tr>
</tbody>
</table>

File: bpel.xml, per partner link setting

```xml
<BPELSuitcase>
  <BPELProcess src="Invoke.bpel" id="Invoke">
    <partnerLinkBindings>
      <partnerLinkBinding name="PartnerService">
        <property name="wsdlLocation">MyPartner.wsdl</property>
        <property name="idempotent">false</property>
      </partnerLinkBinding>
    </partnerLinkBindings>
  </BPELProcess>
</BPELSuitcase>
```

Other Knobs

- nonBlockingInvoke (durable, process level)
  - Perform parallel invoke within FlowN

<table>
<thead>
<tr>
<th>true</th>
<th>Engine will spawn a new thread to execute each invoke (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>false (default)</td>
<td>Engine will execute the invoke activity in the process thread (sequential)</td>
</tr>
</tbody>
</table>

File: bpel.xml, per partner link setting

```xml
<BPELSuitcase>
  <BPELProcess src="Invoke.bpel" id="Invoke">
    <partnerLinkBindings>
      <partnerLinkBinding name="PartnerService">
        <property name="wsdlLocation">MyPartner.wsdl</property>
        <property name="nonBlockingInvoke">true</property>
      </partnerLinkBinding>
    </partnerLinkBindings>
  </BPELProcess>
</BPELSuitcase>
```
Other Knobs

- Invoker Bean Threads: (container level, only for nonBlockingInvoke)
- Mid-tier installation:
  IAS_HOME/j2ee/OC4J_BPEL/application-deployments/orabpel/ejb_ob_engine.jar/orion-ejb.jar.xml
- Developer edition:
  HOME/integration/orabpel/system/appserver/oc4j/j2ee/deployments/orabpel/ejb_ob_engine.jar/orion-ejb.jar.xml

<message-driven-deployment name="InvokerBean" destination-location="jms/collaxa/BPELInvokerQueue" connection-factory-location="jms/collaxa/BPELInvokerQueueFactory"
lis
<ejb-ref-mapping name="ejb/local/ProcessManagerLocalBean" />
...</message-driven-deployment>

Tuning Knobs: Summary

<table>
<thead>
<tr>
<th>Async</th>
<th>Durable</th>
<th>Transient</th>
<th>Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>idempotent</td>
<td>completionPersistPolicy</td>
<td>Process</td>
</tr>
<tr>
<td></td>
<td>nonBlockingInvoke</td>
<td>completionPersistLevel</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>dspMaxThread</td>
<td>dspMaxThread</td>
<td>Domain</td>
</tr>
<tr>
<td></td>
<td>auditLevel</td>
<td>auditLevel</td>
<td></td>
</tr>
<tr>
<td>Container</td>
<td>Worker Bean threads</td>
<td>Worker Bean Threads</td>
<td>Container</td>
</tr>
<tr>
<td></td>
<td>Invoker Bean threads</td>
<td>Datasource max con</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Datasource max con</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To be deprecated in 10.1.3

- deliveryPersistPolicy
- optCacheOn
- instCachePolicy
- instCacheHighWatermark
- instCacheLowWatermark

Example: JMS adapter instantiate a BPEL process once it detects a new message at JMS queue
Tuning Adapters

- Adapters are clients of BPEL engine
- Tuning objectives
  - move messages as fast as possible to engine
- Typical approach:
  - Concurrency
  - Message batching

Tuning JMS/AQ Adapters

- Create multiple activation agents (cut-and-pasting in bpel.xml)
- AQ: Use OCI drivers (oc4j-ra.xml, data-sources.xml)

File: bpel.xml, per partner link settings
<activationAgents>
  <activationAgent className="..." partnerLink="MsgQueuePL">
    <property name="JndiLocation">eis/Jms/ZMSG1</property>
    <property name="JmsQueue">java:comp/.../Queues/ZMSG1QUEUE</property>
  </activationAgent>
  <activationAgent className="..." partnerLink="MsgQueuePL">
    <property name="JndiLocation">eis/Jms/ZMSG1</property>
    <property name="JmsQueue">java:comp/.../Queues/ZMSG1QUEUE</property>
  </activationAgent>
</activationAgents>
Tuning Database Adapters

<table>
<thead>
<tr>
<th>Knobs</th>
<th>Usage</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(inbound) NumberOfThreads</td>
<td># of concurrent threads</td>
<td>As many as possible</td>
</tr>
<tr>
<td>(10.1.3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxTransactionSize</td>
<td>max # rows fetch from DB to adapter in one</td>
<td>• Single BPEL node: ‘unlimited’</td>
</tr>
<tr>
<td>transaction</td>
<td></td>
<td>• Cluster: Set according to data</td>
</tr>
<tr>
<td>MaxRaiseSize</td>
<td>max # rows sent from adapter to BPEL as one</td>
<td>• Set to &gt;1 if outbound is also</td>
</tr>
<tr>
<td>message</td>
<td></td>
<td>• Set to 1 if outbound is NOT</td>
</tr>
<tr>
<td>PollingStrategyName</td>
<td>How to poll table for new rows</td>
<td>Logical delete</td>
</tr>
<tr>
<td>(inbound)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DmlType</td>
<td>How to write to database</td>
<td>Prefer insert, then write, then</td>
</tr>
<tr>
<td>(outbound)</td>
<td></td>
<td>merge</td>
</tr>
</tbody>
</table>

<binding name="pollingService_binding" type="tns:pollingService_ptt">
  <pc:inbound_binding/>
  <operation name="receive">
    <jca:operation
      ActivationSpec="oracle.tip.adapter.db.DBActivationSpec"
      DescriptorName="pollingService.Movies"
      QueryName="pollingService"
      PollingStrategyName="LogicalDeletePollingStrategy"
      MaxRaiseSize="1"
      MaxTransactionSize="unlimited"
      PollingInterval="5"
      NumberOfThreads="1"
      DelayCommit="false"
      PollForChildUpdates="false"
      UseBatchDestroy="false"
      MappingsMetaDataURL="pollingService_toplink_mappings.xml" />
  </operation>
</binding>

• *** Some parameters can be set with the DBAdapter wizard ***
Tuning Database Adapters

```xml
<binding name="insertService_binding" type="tns:insertService_ptt">
  <jca:binding />
  <operation name="write">
    <jca:operation
      InteractionSpec="oracle.tip.adapter.db.DBUpsertInteractionSpec"
      DescriptorName="insertService.SenderDept"
      DmlType="insert"
      MappingsMetaDataURL="insertService_toplink_mappings.xml" />
    <input>
      <jca:header message="hdr:OutboundHeader_msg" part="outboundHeader"/>
    </input>
  </operation>
</binding>
```

• *** Some parameters can be set with the DBAdapter wizard ***

---

Tuning File/FTP Adapters

• FileAdapter/FtpAdapter
  • De-batch inbound file into multiple messages (Wizard)
    • Increase concurrency
    • Decrease the use of While loop
Tuning JVM

- Max. heap size: Set to max. addressable OS memory size
  - 32 bit Linux: 4G
  - 32 bit Windows: 2G
  - E.g. –Xmx 2048m
- Min. heap size: Set to the same as max heap size
  - E.g. –Xms 2048m
- Eden space: Set to 60% of max heap
  - Dictates JVM garbage collection behaviors
  - E.g. -Xmn1228m
- Aggressive heap: for multiple CPU non-Windows machines
  - E.g. -XX:+AggressiveHeap

```
java ... -Xms2048m -Xmx2048m -Xmn1228m -XX:+AggressiveHeap ...
```

Tuning JVM

- Developer edition:
  HOME/integration/orabpel/bin/startorabpel.bat
- Mid-tier installation:
  IAS_HOME/opmn/conf/opmn.xml
Tuning Database

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_buffer</td>
<td>1048576</td>
</tr>
<tr>
<td>shared_pool_size</td>
<td>400M</td>
</tr>
<tr>
<td>job_queue_processes</td>
<td>1</td>
</tr>
<tr>
<td>db_cache_size</td>
<td>1000M</td>
</tr>
<tr>
<td>db_file_multiblock_read_count</td>
<td>8</td>
</tr>
<tr>
<td>undo_retention</td>
<td>0</td>
</tr>
<tr>
<td>processes</td>
<td>300</td>
</tr>
<tr>
<td>session_cached.Cursors</td>
<td>100</td>
</tr>
</tbody>
</table>

BPEL Clustering

- BPEL Optimized SOAP Stack
- Write-through Cache
- Dehydration Store (Oracle Database)
- Binary DOM
  - Lazy Loading
  - Smart Partitioning
  - W3C DOM Interface
  - Support for Large Documents
- Stateless Architecture
  - Clustering
  - Fail Over
- Support for large BPEL Processes (20,000+ activities)
Clustering with Adapters

- Adapters that support all cluster scenarios:
  - JMSAdapter
  - AQAdapter
  - DBAdapter
  - Oracle e-business adapters
  - 3rd party application adapters

- Adapters that support active-passive cluster only
  - FileAdapter
  - FtpAdapter

Agenda

- Agenda
- Benchmarks
- Concepts
- Architecture
- Design
- Tuning
- Tools
- Case Study
Performance Tools

- BPEL Console

<table>
<thead>
<tr>
<th>Tools</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Page</td>
<td>Set domain level knobs</td>
</tr>
<tr>
<td>Statistics Page</td>
<td>Identify bottlenecks</td>
</tr>
<tr>
<td>Threads Page</td>
<td>Examine concurrency</td>
</tr>
<tr>
<td>Logging Page</td>
<td>Optimize logging level</td>
</tr>
</tbody>
</table>

Performance Tools

- Stress Testing Tools
  - From BPEL Console: for quick test
  - Jmeter and commercial tools: for serious testing
Demo

Agenda

- Agenda
- Benchmarks
- Concepts
- Architecture
- Design
- Tuning
- Tools
- Case Study
The Scenario

- Process type: async, transient (no breakpoints)
- Message size: 1K
- JMS adapter

Tuning

<table>
<thead>
<tr>
<th>Modification</th>
<th>Throughput (txn/min)</th>
<th>Improve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>~1,800</td>
<td>--------</td>
</tr>
<tr>
<td>• Change async. to sync. (WSDL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Set completionPersistPolicy=faulted (bpel.xml)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapter</td>
<td>2,500</td>
<td>35%</td>
</tr>
<tr>
<td>Use OCI driver</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j2ee/OC4J_BPEL/config/application.xml)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapter</td>
<td>10,000</td>
<td>400%</td>
</tr>
<tr>
<td>Create multiple (8) adapter agents (bpel.xml)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster</td>
<td>19,000 (virtually unlimited scalability)</td>
<td>95%</td>
</tr>
<tr>
<td>• Stripe the one inbound queue into two</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Create a 2-node queue cluster</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Create a 2-node BPEL cluster</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Crisscrossing adapter agents to queues</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more details, see extra slides section
Conclusions

• Oracle BPEL is designed to perform
• Oracle BPEL has delivered good performance
• Little design and tuning tricks make big difference

For more information ...

• BPEL Performance Guide
  • to be sent after presentation
• BPEL Performance Best Practices Series
  • to be sent after presentation
• Adapter Concepts Guide
  • http://download-west.oracle.com/docs/cd/B14099_19/integrate.1012/b25307/toc.htm
• Oracle Database Tuning Guide
  • http://www.oracle.com/pls/db102/portal.portal_db?selected=4
Q&A and Feedback

Kevin Clugage, kevin.clugage@oracle.com
Product Director

David Shaffer, david.shaffer@oracle.com
Director of Product Management, Oracle Integration

Francis Ip, francis.ip@oracle.com
Principal Solution Architect

Extra Slides
Case Study: Other Tunings

- Cache queue connection and session at en-queue application
- Batch en-queue commit per 10 messages
- Increase EJB max instances
- Set BPEL logging level to “fatal”
- Test with pre-load queues
- Test with pure Java en-queuers and de-queuers

Case Study: Tuning Decisions

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Diagnosis</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Backlog in queue • BPEL CPUs and memory under utilized</td>
<td>Bottleneck at JMS adapter layer.</td>
<td>Increase the number of JMS agents in bpel.xml</td>
</tr>
<tr>
<td>• No backlog in queue • BPEL CPUs and memory are under utilized</td>
<td>Bottleneck at the inbound queue layer.</td>
<td>• First try to adjust the number of en-queue and de-queue threads and their ratio. • Then try to add a node to the inbound message queue cluster</td>
</tr>
<tr>
<td>• Backlog in queue • BPEL CPUs and memory are saturated</td>
<td>Bottleneck is at the BPEL engine layer</td>
<td>Add a BPEL node (and AQ node)</td>
</tr>
</tbody>
</table>
Case Study: Hardware View

Case Study: Software View
Extra: Async. Transient Threads

- Caller thread to receive message
- Separate threads to create/run BPEL instance
- Save messages for Receives

Extra: Sync. Durable Threads

- Initial messages are handled similarly to sync. transient
- Partner callback messages are handled similar to async. durable
Extra: Async. Durable Dehydration

- Save messages for Receives
- Save process states “in flight” at breakpoints (multiple txn)

Extra: Async. Transient Dehydration

- Save incoming messages for initial Receive
- Save process state at the end (one transaction)
Extra: Sync. Durable Dehydration

- Not save messages for initial Receive
- Save messages for mid process Receive
- Save process states “in flight” at breakpoints (multiple transactions)

Extra: Sync. Transient Dehydration

- Not save messages for initial Receive
- Save process state at the end (one transaction)
### Extra: Dehydration Store

#### Knobs

<table>
<thead>
<tr>
<th></th>
<th>Durable</th>
<th>Transient</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Async</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>•auditLevel</td>
<td>•completionPersistPolicy</td>
</tr>
<tr>
<td>Container</td>
<td>N/A</td>
<td>•completionPersistLevel</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>•auditLevel</td>
<td>•auditLevel</td>
</tr>
<tr>
<td>Container</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*ORACLE*