Tuning Your SOA Infrastructure for Performance and Scalability

Vikas Anand
Senior Principal Product Manager
Vikas.anand@oracle.com
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Oracle Fusion Middleware
Complete, Open, Integrated, Best-in-Class
So what does performance mean.

- a dramatic or musical entertainment; "they listened to ten different performances"; "the play ran for 100 performances"; "the frequent performances of the symphony testify to its popularity"

- the act of presenting a play or a piece of music or other entertainment; "we congratulated him on his performance at the rehearsal"; "an inspired performance of Mozart's C minor concerto"

- the act of performing; of doing something successfully; using knowledge as distinguished from merely possessing it; "they criticised his performance as mayor"; "experience generally improves performance"

- any recognized accomplishment; "they admired his performance under stress"; "when Roger Maris powered four home runs in one game his performance merits awe"

- operation: process or manner of functioning or operating; "the power of its engine determines its operation"; "the plane's operation in high winds"; "they compared the cooking performance of each oven"; "the jet's performance conformed to high standards"
Performance – SOA Context

- **Reliable** -
  - End to end transactionality i.e. no order or message loss
  - Recovery of messages – as and when required
  - Automated transaction recovery and rollback capabilities
  - A “view” of the transaction

- **Scalable** -:
  - Vertical vs horizontal scale out – H/W, OS, JDK
  - Sizing – DB, AQ, MQ, Messaging infrastructure, Middleware

- **Throughput** -:
  - Transactions-per-second (TPS)
  - End to end transaction time
  - Messages completed in a given time frame
  - Throughput vs Response Time
Technology...!!!
Specialized Needs For Different Architectural Layers

COMPOSITE APPLICATIONS
High Concurrency, Long lived business processes, Very high reliability, Secure

SHARED SERVICES PLATFORM
Low latency, high throughput, high scalability, very high availability, Secure

Event Management
Real time latency, very high throughput
Scalable foundation

TPS msg/s

Message pattern
- Multi-dimensional scaling
- Service virtualization: Universal connectivity
- Protocol translation
- Location transparency
- Automatic pooling

Data integration pattern
- E-L-T
- Batch operation
- Bulk transfers
- Change Data Capture (CDC)

Eventing pattern
- Unstructured data
- Real-time patterns identification

1,000’s services

ERP MAINFRAME SERVICES DB PARTNERS EVENTS
11gR1 architecture

- Optimized SOA runtime
- Running on Application Grid
- Coherence
- WLS
- JRockit

Oracle Service Bus

Unified Runtime

Common JCA-based connectivity infrastructure

Policy Manager

BPEL
BPMN
Human Workflow (+AMX, AG, Orgn)
Business Rules
Mediator

Process Core

Optimized binding

Repository

EM console +BPMN Screens

©2010 Oracle Corporation
## Typical SOA Deployments

<table>
<thead>
<tr>
<th>Customer Type</th>
<th>Details</th>
</tr>
</thead>
</table>
| Large Telco Customer                 | • 1200 TPS  
• Synchronous /Asynchronous Processes  
• Large Architecture, and Development Teams |
| Retail Customer                      | • Scaling to 1000 TPS from 100 /200 TPS currently  
• 400 CPU’s                   |
| Air Traffic Management customer      | • 300 TPS  
• Millions of lines of Code                                    |
| Front Office Trading                 | • 1 Million Events/Sec @ 69 [$\mu$s] avg (CEP)  
• Stock price increases or decreases by more than 2% percent between 2 ticks |
| Credit Card Company                  | • 40 million msg/day  
• msg size = 5-10KB  
(Credit card transactions)            |
Performance – Integration use case
Design it Right !!!
Performance Anti Patterns – Some Examples...

- Design of the application(s)
  - Development anti-patterns
- Sizing and Capacity
  - Over and under sizing of components
- Deployment Architecture
- Data Retention Policy and Maintenance
Usage of synchronous processes to call asynchronous processes and waits for a response

- May lighten the load on the server but adds a transaction abstraction that needs to be handled
- No “view of the transaction” – need a mechanism to provide location of the ongoing transaction (tracking)
- No server session affinity during callbacks in a cluster environment
- Timeout issues, hanging threads and thread deadlocks
- CPU, Garbage Collection and server threads usage will be high
- Exception and error handling may require customizations
- Scalability may be an issue which can only be resolved with a design change
Anti-Patterns – Synchronous-Asynchronous

Front end and other incoming channels

Oracle Siebel system

Timeout scenarios

Oracle E-Business Suite

Oracle BRM

Oracle OSM

Cluster of Oracle FMW

AQ/JMS Adapters

AQ Cluster

Oracle 11g RAC

PC

Web Channel

Mobile Channel

Sales Partner

BPEL

BPEL

BPEL

BPEL
Anti-Patterns – Synchronous Fault Handling
Anti-Patterns – Design of Applications

- **Usage of extensive loops**
  - Using very big loops to conduct iterative processing – does not scale with large numbers
  - Can lead to big JVM object heaps and potential GC bottlenecks

- **Abuse of FlowN (Batch/event paradigm)**
  - Very large number of FlowN – slows down server, large number of threads
  - How to deal with long running BPEL processes
Anti-Patterns – Design of Applications

- Over usage of asynchronous processes (Mediator/BPEL)
  - Adds dehydration overhead
  - Developers are not aware of transactional boundaries which can quickly complicate the message lifecycle

- Over usage of callbacks (chatty system) in a single BPEL process
  - Every callback is a dehydration point which will cause DB activity
  - Having too many callback activities can quickly impact performance
Anti-Pattern – Asynchronous
Anti-Patterns – Sizing and Capacity

- Over sizing some components while not adding capacity for other components
  - Most notably adding more FMW servers while not adding any more DB capacity
  - Not sizing the dehydration DB correctly, or not able to add disk at the right time (real performance issue)
  - Transaction systems not able to sustain load of incoming requests – leads to “lost messages”, system down scenarios, overall system slow down
Anti-Patterns – Deployment Architecture

- Trying to overextend a cluster’s capability
  - One set of processes consume more resources which impact other set of processes for e.g. OrderConsumption processes vs OrderProcessing processes
  - OrderProcessing takes much longer however OrderConsumption takes away CPU processing power impacting overall performance
Anti-Patterns – Deployment Architecture
Anti-Patterns – Deployment Architecture
Hmm.. how much Data do I really need...!!!
Purging and DB retention policies

- DB purging not being optimized
  - Has one of the biggest impacts on a SOA business application
  - Impacts performance of runtime SQL queries
  - Impacts asynchronous routing rules and asynchronous BPEL processes
  - Slows down console access when querying for auditing data
  - Very large overhead when purging data – longer cycles required, defragmentation of DB, maintenance window
  - If retention periods are long, larger disk space is required – BLOB column tables grow taking up bigger chunk of the DB
Anti-Patterns – Purging and DB Retention Policies

Front end and other incoming channels

- PC
- Web Channel
- Mobile Channel
- Sales Partner

Oracle Siebel system

Mediator

AQ/JMS Adapters

Cluster of Oracle 11g FMW servers

- Slows down console access

Impacts DB performance

Longer purging cycles
Performance Tunings – SOA Infrastructure

- Why Purging?
  - Large volume of stale data

- Choices
  - Loop Purging
  - Parallel Purging
  - Partitioning

- Purging Strategies
  - Small Installations
    - <100k rows
    - Use Loop Purging
  - Medium Installations
    - generates <10G data per day and retains < 500G data
    - Use Parallel Purging
  - Large Installations
    - generates > 10G data per day or retains < 500G data
    - Consider the combination of Parallel Purging and Partitioning
Perfect ... Now let's tune it for performance...
### Performance Tunings – Overview

#### SOA Component Level
- BPEL/BPMN: Thread tuning, Audit-Trail, dehydration, message persistence, Stats
- Mediator: Audit Level, Worker Threads
- Adapters: Threads, batch size

#### SOA Infrastructure
- Audit Level
- Payload Validation
- Purging

#### Generic
- OS
- JVM
- Database
- Weblogic Application Server
Performance Tunings – BPEL/BPMN

BPEL/BPMN Engine Level Properties
Performance Tunings
BPEL/BPMN Engine Level Properties

Concurrency
- # of Threads to process invocation messages (dspInvokeThreads)
- # of Threads to process engine messages (dspEngineThreads)
- # of Threads to process system messages dspSystemThreads

Memory / DB
- Audit Trail logging levels (AuditLevel)
- Message Persistence (oneWayDeliveryPolicy)

CPU
- XML Validations (validateXML)
- Statistics for most recently processed requests (statsLastN)
Performance Tunings - BPEL
BPEL Process Level Properties

Response Time
- Parallel processing of invoke in multiple branches (nonBlockingInvoke)
- Synchronous / Transient process design

Database
- Dehydration needed or not (inMemoryOptimization)
- Instance data storage in DB (completionPersistPolicy)

CPU
- Payload Validation (validateXML)
Performance Tunings – Mediator

Mediator Engine Level Properties

Properties

Edit property values and click Apply to save the changes.

- **Audit Level**
  - Production

- **Metrics Level**
  - Disabled

- **Parallel Worker Threads**
  - 4

- **Parallel Maximum Rows Retrieved**
  - 200

- **Error Locker Thread Sleep (sec)**
  - 2

- **Parameters**
  - Container ID Refresh Time (sec) 60
  - Container ID Lease Timeout (sec) 300
  - Resequencer Locker Thread Sleep (sec) 10
  - Resequencer Maximum Groups Locked 4
  - Resequencer Worker Threads 4

More Mediator Configuration Properties...
Performance Tunings – Mediator

Mediator Level Properties

Concurrency
- Parallel Routing
  - # of parallel threads for message processing
- Resequencer
  - # of worker threads

Database
- Audit Level
- Parallel Routing
  - # of rows that are fetched by each thread per iteration
- Resequencer
  - # of Groups that are locked by each thread at a time

CPU
- Metrics Level
- Sleep intervals before next polling
### Performance Tuning – Adapters

**DB Adapter Properties**

You can edit or delete the following binding properties. Click Add to add additional properties.

<table>
<thead>
<tr>
<th>Name (Operation)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumberOfThreads (receive)</td>
<td>1</td>
</tr>
<tr>
<td>Query/Name (receive)</td>
<td>SyncCustomerPartyReqDBAdapterSelect</td>
</tr>
<tr>
<td>MaxTransactionSize (receive)</td>
<td>10</td>
</tr>
<tr>
<td>MaxReadValue (receive)</td>
<td>1</td>
</tr>
<tr>
<td>PollingInterval (receive)</td>
<td>5</td>
</tr>
<tr>
<td>MaxUnreadValue (receive)</td>
<td>FOD</td>
</tr>
<tr>
<td>ReturnSingleResultSet (receive)</td>
<td>False</td>
</tr>
<tr>
<td>MaxReadSize (receive)</td>
<td>1</td>
</tr>
</tbody>
</table>
Performance Tunings - Adapters
Adapter Level Properties

Concurrency
- # of worker / Poller / Dequeue threads

Database
- # of rows to be processed per transaction (MaxTransactionSize)
- # of file records to be processed (MaxRaiseSize)
- Collect stats on relevant tables
- Large files

CPU
- Process design
- IO
- # Threads at the application server level
Performance Tunings – SOA Infrastructure

SOA Infra Properties

SOA Infrastructure Common Properties

- Audit Level: Development
- Capture Composite Instance State: off
- Payload Validation: off

UDDI Registry Properties

- Inquiry URL
- User
- Password

Server URLs

- Callback Server URL: http://sdd-s20n717.siebel.com:80
- Server URL: http://sdd-s20n717.siebel.com:80
Performance Tunings – SOA Infrastructure

- Audit Level
- Purging

- Payload Validation
Performance Tunings – Generic

**Weblogic**
- Production Mode
- Connection Pooling
- Logging
- Self tuning

**JVM**
- Heap Size
- Nursery Size
- GC Algorithm
- Use Large pages
- 64 bit vs 32 bit

**Database**
- SOA & Application Schema tuning
- Tuning Parameters
- Redo Logs
Performance Tools – Monitoring, Profiling and Debugging

Choice of tool depends on:
- Which components being monitored, which metrics
- Monitoring, Profiling, or Debugging

Profiling
- JRockit Mission Control (JRA cpu profiling, Latency Analyzer, Memory Leak Detector)

Monitoring
- Enterprise Manager, WebLogic Server Admin Console, DMS Spy, WebLogic Diagnostic Framework

Debugging
- Enterprise Manager (Audit Trail enabled), Server Logs

Others
- AWR, DB Enterprise Manager
Summary..

- Understand the requirements and the Use Cases
- Choose the right Technology
- Size it !!!
- Define the right design Patterns
- Deploy the right way !!!
- Leverage the Knobs to Tune
- And Thank You For attending our session…