Federal Agency Pursues Business Logic at the Speed of Big Data

Using jBPM, JDG and FUSE to integrate Microservices

George Batchvarov
Solution Architect at NCI, Inc.
05/03/2017
About the Presenter
About Us

Information about:

- **George Batchvarov**
  
  - 25 years of experience in IT in the private sector and the federal government
  
  - Wide range of technologies from Big Data analytics to high frequency transactional
  
  - LinkedIn: https://www.linkedin.com/in/batchvarov/

- **NCI, Inc.**
  
  - Federal IT and professional services provider serving DoD, Intel, and civilian and health agencies
  
  - Specialize in top-notch IT modernization, program integrity and custom agile development solutions and services
  
  - For more information go to: www.nciinc.com
About Us

Key personnel:

- Dave Omondi - lead developer
  - LinkedIn: https://www.linkedin.com/in/dave-omondi-msc-oca-java-se-7-ccdh-80b71425/
- Ballu Chegu - technical lead
  - LinkedIn: https://www.linkedin.com/in/balluchegu/
- Nathan Cowan - business analyst
  - LinkedIn: https://www.linkedin.com/in/nathan-cowan-3ba80b53/
- Scott Lawrence - project manager
  - LinkedIn: https://www.linkedin.com/in/scottalawrence/
The Challenge
The Challenge

Mission

A federal agency is looking to modernize so they can:

- Provide better service to the public by improving their experience
- Increase compliance by separating involuntary mistakes from fraudulent behavior
- Protect privacy and prevent ID Theft
- Maintain a high level of security
The Challenge

Business Obstacles

The agency identified the following business challenges:

- Data volume – billions of forms processed each year
- High complexity – 74,000 pages of rules and regulations
- Ever-changing business logic – the rules grow with 145,000 words annually
- Fluctuating workload – 90% of the forms arrive within 3 months
- Human intervention – around 20% of the forms require some type of manual intervention
The Challenge

Current State

Numerous legacy interfaces
- mostly batch/file oriented
- usually scheduled to run at a certain time
- dependencies resolved on batch/file level

Service Oriented Architecture (SOA) in infancy phase
- small number of common services
- most just a facade to the batch-based legacy processes

Integration through a mainframe VSAM based file
The Challenge

Current State Cons

Latency introduced by batched-based processing can amount to weeks delay for a single form

High level of code redundancy
  • costly code updates
  • error prone

Massive data copying
  • network capacity is never enough :-)
  • long-running jobs
The Challenge

Desired State

The federal agency seeks help changing to:

- Customer-centric approach
- Addressing most of the problems interactively during the form submission
- Processing of submissions on arrival
- Agility to maintain the complex rules and regulations
- Ability to scale up and down with the workload
- Minimal human intervention
The Solution

Principles

- Utilize Massive Parallel Processing (MPP) platform – shared-nothing architecture
- Process in memory
- Bring the code to the data instead of the data to the code
- Organize the common functionality as micro services
- Use Business Process Management (BPM) software for integration
The Solution

Logical Architecture

<table>
<thead>
<tr>
<th>API Layer</th>
<th>Interactive Service Proxies</th>
<th>Submission Service Proxies</th>
<th>Status Service Proxies</th>
</tr>
</thead>
</table>

In-Memory MPP Platform

Application A
- BPM Software
- Workflow
- Business Model

Application B
- BPM Software
- Workflow
- Business Model

Application C
- BPM Software
- Workflow
- Business Model

Service Façade Layer

Persistence

Reporting and Monitoring

Human Task Management

Micro Service
The Solution

Software stack

- MPP platform - JBoss Data Grid (JDG)
- API Layer and microservices mocks - JBoss FUSE
- BPM - JBoss BPM (jBPM)
- Persistence database - Oracle
- Human task management - jBPM Business Central
- Reporting - jBPM Business Central
- Monitoring - JBoss Operations Network (JON)
- All components run in JBoss Enterprise Application Platform (EAP) container
The Solution

Service Design

API Layer

Interactive Service Proxies
Submission Service Proxies
Status Service Proxies

JDG

Interactive Services
Submission Services
Status Services

write data
drop token
write data
drop token
get status

Bin
Synchronous Cache Listener
Worker

Bin
Asynchronous Cache Listener
Worker

Business Model Pool
The Solution
Parallelism Explained

API Layer (with failover)
- Built with Camel and Jetty
- On Jboss FUSE

Load balancing by round robin

JDG Node 1
- Service
- Cache Listener
- Worker

JDG Node 2
- Service
- Cache Listener
- Worker

JDG Node N
- Service
- Cache Listener
- Worker

Bin (distributed by the same key as the business model)

Business Model Pool (distributed by id)
The Solution
Persistence and High Availability
The Solution
BPM, Micro Services, and Human Tasks

JBoss EAP

JDG Library Mode  Bin  Business Model Pool

jBPM (embedded)

Workflow

Apache CXF

Facade A  Facade B

Async WorkItem

Microservice A  Microservice B

Microservice B

Jboss FUSE

Human Task

jBPM Business Central
The Solution

BPM Persistence

JDG

Workflow A Instance 1

Workflow A Instance 2

Workflow B Instance 3

Write behind

Write behind

Write behind

Evict

Restore

Instance 1 Context

Instance 2 Context

Instance 3 Context

BPM Persistence
The Solution

Reporting

API Layer

JDG

Micro Service A  Micro Service B
Jboss FUSE

JON

jBPM Business Central
Custom Dashboard
BPM Dashboard

JDG Persistence
BPM Persistence
The Solution
Integration Patterns

ORCHESTRATION

CHOREOGRAPHY

JDG
Worker
Human Task
Micro Service

Worker A
Worker B
Worker C
Micro Service
Human Task
The Solution
DevOps and Continuous Integration

BitBucket as code repository
JIRA for agile and issue tracking
Confluence for documentation and Wiki pages
Bamboo to streamline our building process for continuous integration
Dockerized architectural components to be able to quickly deploy and scale up
The Results
The Results

AWS Pilot Architecture

m1.large – 2 CPU, 7.5GB RAM, 200GB SSD
m3.large – 2CPU, 7.5GB RAM, 200GB SSD
db.t2.large – 2CPU, 8GB RAM, 400GB SSD

Cluster

JBoss EAP (m3 large)
- JDG (Library Mode)
- jBPM (embedded)
- Apache CXF
- KIE Remote API

JBoss EAP (m3 large)
- JDG (Library Mode)
- jBPM (embedded)
- Apache CXF
- KIE Remote API

JBoss EAP (m3 large)
- JDG (Library Mode)
- jBPM (embedded)
- Apache CXF
- KIE Remote API

JBoss FUSE (m1 large)
- API Layer
- Micro Services

jBPM (m1 large)
- Human Tasks
- Reporting

JON (m1 large)
- Monitoring

Oracle RDS (db t2 large)
- JDG Persistence
- jBPM Persistence
The Results

Test Plan

- Pilot logic represents approximately 10% of the logic related to a medium complexity form
- In each test, we simulate continuous use with increasing load by increasing the number of service consumer threads until we reach a choke point
- First – find the optimal concurrency level of the cache listener running a single node
- Second – run the test for configuration of 2, 3 and 4 nodes and collect performance metrics
## The Results

### Performance Metrics

<table>
<thead>
<tr>
<th>Single node (number of consumer threads)</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>20</th>
<th>24</th>
<th>28</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms / sec</td>
<td>18.1</td>
<td>23.7</td>
<td><strong>32.1</strong></td>
<td>30.1</td>
<td>29.5</td>
<td>26.6</td>
<td>15.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiple nodes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms / sec</td>
<td>32.1</td>
<td>52.5</td>
<td>71.2</td>
<td>99.0</td>
</tr>
</tbody>
</table>

| Number of consumer threads              | 16 | 32 | 40 | 64 |

Replication factor of 1 used in all tests. 7% performance penalty when increasing the replication factor.
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews