Decisions at a fast pace

Scaling to multi-million transactions/second at Amadeus

Tarek Zaigouche, Manager Software Development at Amadeus
Matteo Casalino, Software Development Engineer
May 2017
Agenda

_ Context
  • Amadeus in a few words
  • Business Rules
    • Functional usage
    • Technical challenges
    • New Requirements

_ Red Hat BRMS integration in Amadeus
  • Study/prototype/benchmarks
  • Industrialization

_ Red Hat & Amadeus collaboration
Amadeus and the Business Rules
Amadeus in a few words

- We are the leading technology company dedicated to the global travel industry
- We are present in 190+ countries and employ 14,000+ people worldwide
- Our solutions enrich travel for billions of people every year
- We work together with our customers, partners and other players in the industry to improve business performance and shape the future of travel

595 million total bookings processed in 2016 using the Amadeus distribution platform

1.4 billion passengers boarded in 2016 with Amadeus and Navitaire solutions

1 of the world’s top 10 software companies
Forbes 2016 global rankings

5th consecutive year included in the DJSI
The only travel technology company in the Dow Jones Sustainability Index in 2016
Amadeus architecture core pillars

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications. Applications & Isolation

Health active monitoring

Flexibility
Scalability
Reliability
Security

Amadeus architecture core pillars

Service Oriented Architecture

Enterprise Service Bus

Red Hat Enterprise Application Platform

Amadeus OTF Application Server

Travel Agents, Airlines, Hotels, e-Commerce, Airports, Load Control, Self-service, Impacted Systems

Red Hat
Enterprise
Application
Platform

Amadeus OTF Application Server

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications

Health active monitoring

Amadeus

Flexibility
Scalability
Reliability
Security

Amadeus architecture core pillars

Service Oriented Architecture

Enterprise Service Bus

Red Hat Enterprise Application Platform

Amadeus OTF Application Server

Travel Agents, Airlines, Hotels, e-Commerce, Airports, Load Control, Self-service, Impacted Systems

Red Hat
Enterprise
Application
Platform

Amadeus OTF Application Server

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications

Health active monitoring

Amadeus

Flexibility
Scalability
Reliability
Security

Amadeus architecture core pillars

Service Oriented Architecture

Enterprise Service Bus

Red Hat Enterprise Application Platform

Amadeus OTF Application Server

Travel Agents, Airlines, Hotels, e-Commerce, Airports, Load Control, Self-service, Impacted Systems

Red Hat
Enterprise
Application
Platform

Amadeus OTF Application Server

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications

Health active monitoring

Amadeus

Flexibility
Scalability
Reliability
Security

Amadeus architecture core pillars

Service Oriented Architecture

Enterprise Service Bus

Red Hat Enterprise Application Platform

Amadeus OTF Application Server

Travel Agents, Airlines, Hotels, e-Commerce, Airports, Load Control, Self-service, Impacted Systems

Red Hat
Enterprise
Application
Platform

Amadeus OTF Application Server

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications

Health active monitoring

Amadeus

Flexibility
Scalability
Reliability
Security

Amadeus architecture core pillars

Service Oriented Architecture

Enterprise Service Bus

Red Hat Enterprise Application Platform

Amadeus OTF Application Server

Travel Agents, Airlines, Hotels, e-Commerce, Airports, Load Control, Self-service, Impacted Systems

Red Hat
Enterprise
Application
Platform

Amadeus OTF Application Server

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications

Health active monitoring

Amadeus

Flexibility
Scalability
Reliability
Security

Amadeus architecture core pillars

Service Oriented Architecture

Enterprise Service Bus

Red Hat Enterprise Application Platform

Amadeus OTF Application Server

Travel Agents, Airlines, Hotels, e-Commerce, Airports, Load Control, Self-service, Impacted Systems

Red Hat
Enterprise
Application
Platform

Amadeus OTF Application Server

The Open Transaction Framework (OTF) is the Amadeus’ Enterprise Application Server, a container to deploy, configure and manage applications

Health active monitoring

Amadeus
Enriching every step of the traveler's journey

Business Rules everywhere!
## Amadeus Business Rules

### Example

<table>
<thead>
<tr>
<th>Criterion Name</th>
<th>Market Origin</th>
<th>Market Destination</th>
<th>Flight Group</th>
<th>Start Date</th>
<th>End Date</th>
<th>Meal</th>
<th>Rule Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>World</td>
<td>UK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Rule 2</td>
<td>UK</td>
<td>World</td>
<td>BA1-15</td>
<td>01APR17</td>
<td>30APR17</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Rule 3</td>
<td>London</td>
<td>Paris</td>
<td></td>
<td>01APR17</td>
<td>30APR17</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

**Rule Type Definition**

- **Most Precise Rule**: The rule with the highest weight is considered the most precise rule.

**Criterion Values**

- **Decision Value**: The specific criteria values used in the rules.
Customization of the Behavior of Amadeus Applications

Amadeus Business Rules

- In-house rule engine **embedded in all Amadeus applications**
- Tailored for the Travel Industry

Markets
Customized aggregation of geographical locations

Points of Sale
Customized points where the purchase is performed

Flight Groups
Customized list of flights

Standard Types:
String, date, integer ...

- Rules administration User Interfaces
- Access control
- Excel and csv Import Export for massive manipulation
- Rules History
Customization of the Behavior of Amadeus Applications

Amadeus Business Rules

- ~2 Million rules
- ~150,000 flight groups
- ~85,000 markets
- ~40,000 points of sale

Biggest clients manage more than 40,000 business rules

1500 business rule owners in Production

✓ Matching logic: Most precise rule
✓ High performances
✓ ~3 Million aggregated evaluations/sec
ABR High Level Architecture
New Requirements

- More complex usecases
  - Expressivity
    - No constraint on the number of conditions
    - Ex: ABR limitation to 99 criteria
  - Calculation
    - Inference

- Agility for faster time to market
  - Dynamicity
    - ABR is static because of rule template (criteria and action)
    - More operations for more complex conditions than ABR: =, <, >, regexp, contains, memberOf, join, inference...

- Reduce operational footprint
  - Decrease CPU cost for cloud deployment
Red Hat BRMS Integration in Amadeus
Challenges

Business
- Custom Travel Semantics to be integrated
- Rule management serviceability (easy integration in business UIs)

Technology
- Integration with C++ applications
- Business rules sharing between C++ and Java applications

Operability
- Distributed application servers
- Rolling deployment
Red Hat BRMS in C++
Evaluation Prototype

SW/HW Configuration
- SLES11/JDK8
- 24 cores @ 3.33GHz / 48 GB Ram

Client-Server setup
- Client (C++) side: Zero-MQ client
- Server (Java) side: Zero-MQ multithreaded server / Red Hat Enterprise BRMS 6.3.0

Test protocol
- Conversion of ABR rules into Drools rules
  - 1 Simple Use case
  - 1 Complex Use case (inference)
- Measure rule evaluation round trip latency
  - 1 (random) input fact per fireAllRules call
  - From 10 up to 100k rules
Performances

Scalability

- Minimum connecting time rules
  - 1 input fact
  - 1 (most precise) rule match

- Red Hat BRMS outperforms ABR when ruleset grows past 1K rules
Performances
Market and Point Of Sale Semantics

Availability suppression rules

Market and Point of Sale matching
• E.g., most precise geographical location, or travel agent office

Two strategies
• Use inference: market rule triggers other rules
• Hardcoded facts preprocessing logic

Availability Suppression - Latency (us)

Drools (~300 rules w/ inference)
Drools (~150 rules no inference)
ABR (~150 rules)

Rules Evaluation
IPC
Facts Encoding / Decoding
IPC solutions

2 candidate solutions for Latency/CPU compromise

**ZEROMQ IPC**
- Latency: 108 us
  - 26% Drools evaluation
  - 74% IPC
- Latency: 110 us
  - 27% Drools evaluation
  - 73% IPC
- Latency: 120 us
  - 33% Drools evaluation
  - 67% IPC

**INFORMATICA IPC**
- Payload size 100 Bytes
  - Latency: 84 us
  - Drools evaluation 5%
  - IPC 95%
- Payload size 1K Bytes
  - Latency: 84 us
  - Drools evaluation 5%
  - IPC 95%
- Payload size 10K Bytes
  - Latency: 90 us
  - Drools evaluation 11%
  - IPC 89%
Definitely candidate for integration in Amadeus Backend Applications

- BRMS Drools scalable and faster than ABR
  - BRMS: Average latency of 80us from 10 to 200 000 rules
  - ABR: Average latency of 100us to 20ms from 10 to 200 000 rules

- BRMS Drools better CPU consumption on certain usecases*
  - BRMS: 1 Million evaluations/sec with 70 cores
  - ABR: 1 Million evaluations/sec with 700 cores
  - BRMS: 24 cores machine: 350 000 evaluations/sec
  - ABR: 24 cores machine: 35 000 evaluations/sec

- Validation of our long term vision to have a common rule engine for C++ and Java applications

* N.B.: This extrapolation is an order of magnitude measured in local development machines for MCT & FAC rules. CPU usage may defer depending on how complex the rules are.
Architecture for C++ Application Servers
Rolling Deployment

Goal: avoid outage while switching to new rules version
- Load new KieContainer in separate thread
- Switch reference when ready
- Weak References to avoid classloader leaks

Server startup time still impacted
- Critical operability criterion
- KieBase serialization helps
BRMS Drools for C++

Technical Stack

ZeroMQ-managed IPC channel
- Rule evaluation jobs **queueing**
- Load balancing over multiple worker threads
- Seamless crash **resiliency**
Drools Fiddle

https://github.com/droolsfiddle
Red Hat & Amadeus Collaboration
RedHat & Amadeus collaboration

- **Openshift**
  - Intensive collaboration for Amadeus Cloud Services

- **BRMS**
  - Excellent contact with Gabriel!
  - Kick off with 1 months consultancy
    - Functional and technical understanding
    - Architecture
    - Benchmarks
  - 3 days technical workshop in Boston with Edson
  - BRMS Enterprise Subscription
  - 3 months consultancy per year
Upstream and next steps

- Alias feature to emulate the rolling deployment
  - ✔️ BRMS 7

- Drools Fiddle: potential integration in BRMS Workbench
  - ✔️ New repository & blog

- Protobuf encoding for KieServer commands

- KieBase protobuf serialization in the KJar
Memory profiling of the JVM

Current worse case scenario: 1 rule type MCT, 1 owner, 200k rules
Memory profiling of the JVM

Average application: 1 rule type, 25 owners, 1k rules, 1 JVM, 10 workers
Drools scalability

**Average Evaluation response time/rule set size**

- **Drools**

**Evaluation response time for 200k rules**

- ~~80 us~~

**Avg Latency**

- **avg_fire**
- **avg_all**
Different IPC solutions comparison

Latency
Different IPC solutions comparison

CPU Active

![CPU Active Graph][1]

[1]: http://example.com/graph.png
Different IPC solutions comparison

CPU Idle

![CPU Idle graph](chart)

- CPU Idle

© 2015 Amadeus IT Group SA
Time to load rules package in memory

- DRL
- KJAR
- Serialized kBase
Throughput and CPU measures

- Concurrent c++ clients sending rules evaluation to a single java server
- Measure of the CPU consumed by the JVM
- On local development machines, we can serve 100k TPS by using around 7 cores (for FAC)

- Extrapolation
  - 1M TPS => ~70 cores
  - 1M TPS for ABR => 700 cores!

CPU/TPS graph shows the CPU cost of 1 transaction