Using Machine Learning, Red Hat Process Automation Manager (BPM Suite), and Reactive Microservices

S1506

Andy Bonham, Distinguished Engineer, Capital One
Hrishi Menon, Lead Software Engineer, Capital One
Murph, Solutions Architect, Red Hat
May 10, 2018
Capital One at a glance

- A leading diversified bank with $365.7 billion in assets, $255.4 billion in loans and $243.7 billion in deposits
  - 8th largest bank based on U.S. deposits
  - 6th largest retail depository institution in metro New York
  - Largest consumer and commercial banking institution headquartered in the Washington, DC region
  - 3rd largest credit card issuer in the U.S.
  - The 3rd largest issuer of small business credit cards in the U.S.
  - Largest financial institution auto loan originator
  - Largest U.S. direct bank
- Major operations in 15 U.S. cities, Canada, U.K.
- More than 70 million customer accounts and 49,000 associates
- A FORTUNE 500 Company - #100
- Numerous recent awards including:
  - Named to 100 Best Companies to Work For by FORTUNE Magazine
  - Best Places to Work for LGBT Equality by Human Rights Campaign
  - Received J.D. Power & Associates Call Center Certification
  - Aon Hewitt’s Top Companies for Leaders
  - Named to Working Mother’s 100 Best Companies list & Best Companies for Hourly Workers
  - Ranked #14 on Military Times’ 2017 “Best for Vets”
  - Recipient of the Secretary of Defense Employer Support Freedom Award

1) Source: Company reported data as of Q4’17
2) Source: FDIC, Domestic deposits ranking as of Q4’17
3) Source: FDIC, June 2017, deposits capped at $1B per branch
4) Source: Company-reported domestic credit card outstandings, Q4’17
5) Source: The Nilson Report, Issue #1111, June 2017
6) Note: Financial institutions includes banks & specialty finance lenders.
7) Source: Regulatory filings, company reports as of June 2017
We are one of the largest banks in the U.S.

<table>
<thead>
<tr>
<th>Position</th>
<th>Bank</th>
<th>2017 Total Loans ($B)</th>
<th>Bank</th>
<th>2017 U.S. Deposits ($B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wells Fargo</td>
<td>976.9</td>
<td>Bank of America</td>
<td>1,227.5</td>
</tr>
<tr>
<td>2</td>
<td>Bank of America</td>
<td>948.2</td>
<td>Wells Fargo</td>
<td>1,207.3</td>
</tr>
<tr>
<td>3</td>
<td>JPM Chase</td>
<td>930.7</td>
<td>JPM Chase</td>
<td>1,187.3</td>
</tr>
<tr>
<td>4</td>
<td>Citigroup</td>
<td>688.1</td>
<td>Citigroup</td>
<td>445.6</td>
</tr>
<tr>
<td>5</td>
<td>U.S. Bancorp</td>
<td>283.4</td>
<td>U.S. Bancorp</td>
<td>321.9</td>
</tr>
<tr>
<td>6</td>
<td>Capital One</td>
<td>255.4</td>
<td>TD Bank</td>
<td>264.1</td>
</tr>
<tr>
<td>7</td>
<td>PNC</td>
<td>223.1</td>
<td>PNC</td>
<td>262.4</td>
</tr>
<tr>
<td>8</td>
<td>TD Bank</td>
<td>154.3</td>
<td>Capital One</td>
<td>243.2</td>
</tr>
<tr>
<td>9</td>
<td>SunTrust</td>
<td>145.5</td>
<td>SunTrust</td>
<td>160.8</td>
</tr>
<tr>
<td>10</td>
<td>BB&amp;T</td>
<td>144.8</td>
<td>BB&amp;T</td>
<td>157.4</td>
</tr>
<tr>
<td>11</td>
<td>American Express</td>
<td>127.4</td>
<td>Citizens</td>
<td>115.0</td>
</tr>
<tr>
<td>12</td>
<td>Ally</td>
<td>123.0</td>
<td>Key</td>
<td>105.3</td>
</tr>
<tr>
<td>13</td>
<td>Citizens</td>
<td>111.3</td>
<td>Fifth Third</td>
<td>103.1</td>
</tr>
<tr>
<td>14</td>
<td>Fifth Third</td>
<td>92.5</td>
<td>Regions</td>
<td>97.2</td>
</tr>
<tr>
<td>15</td>
<td>M&amp;T</td>
<td>88.0</td>
<td>Ally</td>
<td>93.2</td>
</tr>
</tbody>
</table>

Notes: Excludes banks with high non-loan asset concentrations: Goldman Sachs, Morgan Stanley, BONY, State Street, Charles Schwab. Gross loans and domestic deposit data as of 12/31/2017. Based upon total gross loans and total aggregated domestic deposits for bank holding company. 
Sources: SNL, FDIC
We have transformed the company into a top 10 bank

- 2017 – Acquires Notch
- 2016 – Acquires Critical Stack and Paribus
- 2015 – Acquires GE Capital’s Healthcare Financial Services and Monsoon
- 2014 – Acquires Adaptive Path, a digital design leader
- 2013 – Acquires Beech Street Capital, an originator, underwriter and servicer of multifamily commercial real estate loans
- 2012 – Acquires ING DIRECT, HSBC US Card portfolio
- 2010 – Enters into card partnerships with Kohl's and Sony in the US and Hudson's Bay Company and Delta in Canada
- 2009 – Acquires Chevy Chase Bank in the Washington, DC area
- 2006 – Acquires North Fork Bank, one of the largest banks in the New York metro area
- 2005 – Acquires Hibernia National Bank, #1 bank in Louisiana
- 2002 – Launches its Small Business credit card
- 2000 – Introduces slogan, “What’s in your wallet?”
- 1998 – Enters Auto Finance Market
- 1996 – Expands into Canada and the U.K.
- 1994 – Initial Public Offering (IPO)
Previously, we presented how you can use BPM Suite, Reactive Microservices, Docker, and ECS

https://github.com/andy9876/ReactiveBPMDemo

- BPM Suite for coordinating a reactive workflow
- Leverage Kafka as the messaging mechanism between the microservices and BPM Suite
- Docker for microservice deployment
- ECS for docker container management

In today’s session, we are going to extend this pattern to show how you can integrate with machine learning
A monolithic application can be challenging...

A monolithic application puts all its functionality into a single process...

... and scales replicating the monolith on multiple servers

Monoliths can have many dependencies

Monoliths are also hard to change

*This illustration is from http://martinfowler.com/articles/microservices.html*
Microservices to the rescue! But are they the silver bullet?

A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.

*This illustration is from http://martinfowler.com/articles/microservices.html*
A reactive architecture leverages asynchronous messaging between microservices.
In 2013, the Reactive Manifesto was born and highlights the key principles of a reactive architecture.

http://www.reactivemanifesto.org/
A reactive architecture has both benefits and tradeoffs

**BENEFITS**

- **Better resource utilization, saving cost**
  - Can get higher efficiency out of CPUs (multi-core processors), doing more with less

- **More Agile**
  - Decoupling enables services to be updated independently

- **Faster response times as requests can run in parallel**
  - Back pressure can be used for flow control
    - Fast producers don’t overwhelm slower consumers
    - Enables a consumer to control queue bounds

- **Extensible**
  - New components can be added that listen to the event stream without re-writing the system

**TRADEOFFS**

- **Async programming is a mind shift**

- **Complexity**
  - The flow of the system is shifted from a central place to distributed services
Red Hat Process Automation Manager

- Bridges Business and Technical resources together
- Leverages BPMN 2.0
- Natively integrated with Drools Rules
- Supports human workflow and system workflow
- Various APIs
- Java based and very Extensible
- BAM, Dashboards, Reports, Analytics
Kafka is a distributed streaming platform

- Used in conjunction with Zookeeper
- Runs as a cluster
- Records are stored in categories called topics
- Provides 4 core APIs: Producer, Consumer, Streams and Connector
- Supports both publish-subscribe and queuing through a consumer group concept
- Very Fast and has very high throughput – many use it for backpressure
- Can be used for message replay as the messages do not have destructive reads like traditional messaging technologies
- Guarantees order of messages within a partition, but not across partitions
- Very easy to get up and running

http://kafka.apache.org/documentation.html
PAM Kafka Integration – KIE Server
kie-server extension

KIE Server is built on a concept of extensions. All capabilities are implemented as extensions

- KIE Server extension
- Drools extension
- jBPM extension
- jBPM UI extension
PAM Suite Kafka Integration – Extending KIE Server

kie-server extension

So we’re going to add one more
- KIE Server extension
- Drools extension
- jBPM extension
- jBPM UI extension
- Kafka extension
PAM Suite Kafka Integration Architecture
Machine Learning is a type of Artificial Intelligence

Artificial Intelligence (AI) systems are able to perform tasks that normally require humans. E.g., If/then logic.

Machine Learning (ML) is a subcategory of AI that provides the ability to automatically train the system. E.g., Regression.

Deep Learning is an ML subcategory that provides the ability to automatically train the system with multiple layers that mimic layers of neurons in the brain. E.g., Deep Neural Networks.

- **1950s:** Scripted Chatbots
- **1960s:** Medical Diagnosis Expert Systems
- **1970s:** Product Recommendation Engines (e.g., Netflix, Amazon, etc)
- **1980s:** Spam Filtering
- **1990s:** Adaptive Pricing Systems
- **2000s:** Image Recognition
- **2010s:** Self-Driving Cars

Examples of AI applications:
- Medical Diagnosis Expert Systems
- Product Recommendation Engines (e.g., Netflix, Amazon, etc)
- Spam Filtering
- Adaptive Pricing Systems
- Image Recognition
- Self-Driving Cars
- Siri, Alexa, Google Home, Cortana
There are two major classes of Machine Learning

**Supervised**
A human labels the data used
Labor intensive, but more interpretable

**Unsupervised**
An algorithm discovers relationships
Easier, but less interpretable
H2O Overview

- Open source (Apache license) in-memory big data machine learning platform
- Created in 2011 and is written in Java, Python and R
- Can be used with Python, R, H2O Flow, Scala, Tableau, Spotfire
- Uses parallelized and distributed algorithms like GLM, Random Forest, GBM, PCA, deep learning neural networks
- Also supports supervised and unsupervised learning
- Deploy model as a Java POJO or MOJO (Model ObJect, Optimized)

Image from https://www.h2o.ai/h2o/
Below are two options for integrating machine learning

- Plug into the event stream as another reactive microservice
- Integrate directly into BPM Suite as a workitem handler
Combining all of these technologies & patterns together can create a powerful solution

- PAM for coordinating a reactive workflow
- Leverage Kafka as the messaging mechanism between the microservices and BPM Suite
- Docker for microservice deployment
- ECS/Kubernetes for docker container management
- H20 for machine learning
Demo use case and Machine Learning components
Fraudulent Transaction Use Case
A business process to simulate the transaction

This business process submits the below event to Kafka:
{"id": "33bb75db-6e13-48ee-8a54-b3976d3d065b","action": "Transaction Received"}
For the machine learning components, we are using a Credit Card Fraud Detection dataset from kaggle.com:

- Contains 284807 total rows, 492 which are fraud
- V1-V28 are unidentifiable numeric features along with Time
- The time column contains the seconds elapsed between each transaction and the first transaction in the dataset.
- Amount is the transaction amount
- Class column is 1 for Fraud, 0 for not fraud
To build the model we used H2O Flow

- H2O Flow is an open-source user interface for building H2O models
- Very easy to get up and running. Basically download the zip file and run the jar file
  
  ```
  unzip h2o-3.16.0.2.zip
  cd h2o-3.16.0.2
  java -jar h2o.jar
  ```
- Navigate to [http://localhost:54321](http://localhost:54321)

---

#redhat #rhsummit
H2O Flow enables you to do multiple trial runs with various algorithms

- Important that the training and validation frame are not the same dataset
- Response_column is important as this is the data element you are trying to predict
Distributed Random Forest performed better than Gradient Boosting Machine (GBM)

- Both Random Forest and GBM are machine learning techniques for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

- Random Forest uses deep decision trees, while GBM uses shallower decision trees.
H20 identified which of the features were most important and revised the model until it was accurate.
We deployed the H2O model as a POJO

- Install H20, java (versions 7 or 8)
- Download the kaggle dataset, creditcard.csv
- Build the model, selecting the response_column
- Download the java generated code for the model and the packed library from the H2O instance
  curl -o h2o-genmodel.jar http://192.168.1.6:54321/3/h2o-genmodel.jar
- Compile the java generated code
  javac -verbose -cp h2o-genmodel.jar -J-Xmx2g -J-XX:MaxPermSize=128m
gbm_68aaf4ae_9808_48dc_b08d_383c94323392.java
- Create a main.java to invoke the model with features
  javac --cp h2o-genmodel.jar -J-Xmx2g -J-XX:MaxPermSize=128m
gbm_68aaf4ae_9808_48dc_b08d_383c94323392.java main.java
- Execute the model
  java -cp .:h2o-genmodel.jar main
- For our demo, we integrated main.java into the RunModelMS Kafka producer/consumer
Live DEMO!
Lessons Learned of this approach

- Coordinator can be a single point of failure
  - Evaluate multi-deployment for active/active (application and database layers)
- Need a Unique ID (correlation ID) that goes across all microservices
- For the machine learning dataset, use big data and unbiased data
- Leverage Chaos testing to validate resiliency
- Apply this pattern where:
  - there are synchronous blocks of asynchronous processing
  - there is a need to see the overall end-to-end business process at design time and run-time
  - there is a need to decouple as much as possible to eliminate dependencies
References

- Github repo for demo
  https://github.com/andy9876/MachineLearningReactiveBPM

- Microservices: When to react vs. orchestrate
  https://medium.com/capital-one-developers/microservices-when-to-react-vs-orchestrate-c6b18308a14c

- A Reactive Framework Comparison
  https://medium.com/capital-one-developers/building-microservices-a-reactive-framework-comparison-fb49d8f3c8f4

- Comparing and Contrasting Open Source BPM Products
THANK YOU

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

- Create cluster
- Create task definitions
- Create service
Create Cluster
Create Task Definitions
Create Service

Update Service
A service lets you specify how many copies of your task definition to run. You could also use Elastic Load Balancing to distribute incoming traffic to your tasks. Amazon EDS keeps that number of tasks running and coordinates task scheduling with the load balancer.

- **Task Definition**: botroute-3
- **Cluster**: RechatSummitDemo
- **Service name**: botroute
- **Number of tasks**: 1
- **Minimum healthy percent**: 50
- **Maximum percent**: 200

Optional configurations
Service Auto scaling
Automatically adjust your service's desired count up and down in response to CloudWatch alarms.

Configure Service Auto Scaling
ECS deploys the container to the ECS Cluster
Task is deployed successfully
Task is deployed successfully
EC2 Container Instance is running