A Stock Prediction System using open-source software

Fred Melo
fmelo@pivotal.io
@fredmelo_br

William Markito
wmarkito@pivotal.io
@william_markito
It's all about **DATA**

**Data Sources**

Look for patterns

**Prediction**
Binary Options Trading Signals

- Start Time: 2015.01.19 08:10:00
- Expire Time: 2015.01.19 08:15:00
- Last Close: 1.16121
- Predicted Close Direction: Down
- Accuracy: 70.7%
Machine Learning is the answer

Clustering

Neural Networks

Genetic Algorithms
Applying Machine Learning

Train with historical dataset

Apply model to the new input
Why so hard?

Hard to scale

Hard to make it real-time

Hard to add new data sources

Why?
Traditional models are reactive and static

- No real-time information
- ETL based
- Data-source specific

- Hard to change
- Labor intensive
- Inefficient
Stream-based, real-time closed-loop analytics are needed

Data Stream Pipeline

In-Memory Real-Time Data

Expert System / Machine Learning

Data Lake HDFS

Continuous Learning
Continuous Improvement
Continuous Adapting

Multiple Data Sources
Real-Time Processing
Store Everything
How can it be addressed?

Info

Analysis

Neural Network

Look at past trends
(for similar input)

Evaluate current input

Score / Predict
How can it be addressed?

Info

Analysis

Filter

[json]

Neural Network
How can it be addressed?
How can it be addressed?
How can it be addressed?
How can it be addressed?
How can it be addressed?
How can it be addressed?
Streaming real-time analytics architecture

Ingest → Transform → Sink

Distributed Computing
Spark

Other Sources and Destinations
JMS
kafka
mongoDB

Fast Data
GEODE

Store / Analyze
Hadoop

Predict / Machine Learning
MAClib
Python
R
Demo Architecture

Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable

Machine Learning

Fast Data

Dashboard

HTTP
- SpringXD
- shell - R
- Transformer
- geode-json client
- http-client
- http-server
- obj-to-json splitter
- tap
- Simulator
- Yahoo!
- JavaFX
- Chrome
- d3.js
- GEODE
- tap
- Input
- Hidden
- Output
Data Stream Pipelining

**INGEST / SINK**
- Little or no coding required
- Dozens of built-in connectors
- Seamless integration with Kafka, Sqoop
- Create new connectors easily using Spring

**PROCESS**
- Call Spark, Reactor or RxJava
- Built-in configurable filtering, splitting and transformation
- Out-of-box configurable jobs for batch processing

**ANALYZE**
- Import and invoke PMML jobs easily
- Call Python, R, Madlib and other tools
- Built-in configurable counters and gauges
Demo Architecture

Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable

Split → Filter → Transform → Sink

Predict → Sink

Machine Learning

Fast Data

Dashboard

HTTP

HTTP

HTTP
Geode client-server architecture

- **GemFire Client**
  - Local Cache
  - Connection pool
    - Send address and load information to locator
    - Request server information from locator. Locator responds with least loaded server address.

- **GemFire Locator**
  - Send, receive cache data. Receive server events

- **GemFire Server**
  - Partitioned Region
Partitioned Regions

Partitioned Regions allow for the separation of data across multiple servers. In the diagram, GemFire Server1 and GemFire Server2 each have two regions: Region A and Region B. Each region is further divided into primary and redundant buckets. The arrows indicate primary to redundant replication, ensuring data availability and fault tolerance.
The pool propagates the event to the cache server, where the region is updated.

The server distributes the event to its peers and also places it into the subscription queue for Client 2.
Demo Architecture

Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable

Split → Filter → Transform → Sink

SpringXD

Predict → Sink

Machine Learning

Fast Data

Dashboard

HTTP

Push
Neural Networks

- Dendrites
- Nucleus
- Axon
- Boutons
Neural Networks
Neural Network

# R sample source code for NN using RSNNS project
# http://cran.r-project.org/web/packages/RSNNS/
# (...)

inputs <- techIndicators[,inputColumns(techIndicators)]
targets <- techIndicators[,outputColumns(techIndicators)]
patterns <- splitForTrainingAndTest(inputs, targets, ratio = 0.15)

model <- jordan(patterns$inputsTrain, patterns$targetsTrain,
+       size = c(8), learnFuncParams = c(0.1), maxit = 5000,
+       inputsTest = patterns$inputsTest, targetsTest = patterns$targetsTest,
+       linOut = FALSE)
Demo Architecture

Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable

Transform
Sink

Split ➔ Filter ➔ Transform ➔ Sink

Machine Learning

Fast Data

GEODE

Dashboard

Push

HTTP

HTTP

HTTP
SO YOU MEAN TO TELL ME...

...IT'S THAT SIMPLE?
Demo Time

LET'S MAKE SOMETHING HAPPEN
SpringXD  
http://projectgeode.org  
http://projects.spring.io/spring-xd  
http://www.r-project.org
ANY questions?
The demo code is on GitHub!

@fremelo_br
@william_markito

Follow-up: In-Memory Unconference

"A place for all things in-memory: projects, people, ideas, roadmaps, discussions."
Location: Hill Country A/B"

Weds 4:15pm - 6pm. (after this talk)