Inside Cigna’s big data journey

A Case Study of Hadoop in Healthcare

Jeff Shmain (Principal Solutions Architect - Cloudera)
Mohammad Quraishi (Senior Principal Technologist - Cigna) - @AtifQ
About - Jeff

- BS Computer Science, NYU Poly
- MBA Finance, NYU Stern School of Business
- 15+ years Financial and Healthcare Industries
  - Software Development for most of my career
  - Spend a number of years on Wall Street at NYSE and Bank of America, designing and developing automated trading strategies
  - Now working in Cloudera as a Solutions Architect working with a variety of clients in Financial Service and Healthcare

In my spare time
- Research Industry trends and regulations
- Play with gadgets
- Blogging

Fun Fact: Youngest MBA Graduate at Stern School of Business
About - Mohammad

• BS in Computer Science and Engineering from University of Connecticut
• In the Healthcare Industry for over 20 years
  • Programmer most of my career - Architect, Designer
  • Worked in the SOA space for a number of years
  • Lead engineer in the mobile application space
  • Now Lead Engineer in the Big Data Analytics Space - Hadoop

In my spare time
• Love to travel with the family
• Video games, music, movies
• Community relations work
• Fan of College basketball
Breakdown of the Hadoop Journey

1. Making the case
   - Vision
   - Architecture

2. The blowback
   - What we accomplished

3. Roadmap to the future
   - Lessons Learned
   - Questions?
The Elephant in the room

Image Credit: Guian Bolisay/Flickr
What’s the problem?

We already have a mature data analysis infrastructure
And it looks something like this...

What we already do

- We have independent data marts
- We have the Hub-and-spoke architecture, the centralized warehouse
The vision

The ability to perform
• Volume, Variety and Velocity
• Descriptive, Predictive and Prescriptive Analytics

Remove the traditional IT barriers separating the business users from insights
Benefits of Big Data

• Hadoop has the lowest cost per TB ratio of any data technology available

• Getting started with Hadoop is fairly inexpensive
  • “Entry-level” clusters relatively inexpensive
  • Grow in small steps

• Ease of “Iteration” & “Data Blending” for Agile Analytics
Benefits of Big Data

You don’t have to throw away data anymore!
The Initial Evaluation

• Vendor Evaluation: Which relationship best fits our needs without lock-in?

• Selection of use cases for demonstration

• Visualization of those use cases
Hadoop
Spark Ecosystem & Hadoop

- Spark Streaming
- MLlib
- SparkSQL
- GraphX
- Dataframes
- SparkR

Spark

Impala

Search

MR

Others

RESOURCE MANAGEMENT
YARN

STORAGE
HDFS, HBase
The Future of Data Processing on Hadoop

**General Data Processing w/ Spark**
Fast Batch Processing, Machine Learning, and Stream Processing

**Full-Text Search w/ Solr**
Querying textual data

**Analytic Database w/ Impala**
Low-Latency Massively Concurrent Queries

**On-Disk Processing w/ MapReduce**
Jobs at extreme scale and extremely disk IO intensive

Shared:
- Data Storage
- Metadata
- Resource Management
- Administration
- Security
- Governance
The Four Pillars of Security

Authentication
Authorization
Encryption
Auditing
Vision - Reference Architecture

   - Jobs
   - MapReduce Distributed Programming Framework

2. Edge Node For Hadoop Client
   - SQOOP/Flume/Chronsos
   - Python
   - Scalding (Scala)
   - Hive/Impala (SQL)
   - Cascading (Java)

3. Realtime Data Store or event processing
   - Logs
   - Web
   - IVR
   - Portal
   - Mobile
   - Log files to HDFS
   - NoSQL Storing weblogs
   - Data from tables to HDFS
   - Live Data
   - Streams
   - Web Analytics
   - Event detection (Storm)

4. External Hadoop Output Or in HDFS
   - Teradata
   - RDBMS

5. RDBMS
   - CED/Claims
   - Clinical Data
   - Teradata

6. Data Science Tools
   - Tableau
   - Spotfire
   - Platfora
   - Cognos
   - Microstrategy

7. Analysis/Modeling Tools
   - SAS
   - Pentaho R?

8. Back up or copy data from HDFS to a redundant cluster for quick recovery
   * For future implementation TBD

*Use Spark Streaming and append to Hadoop output - Realtime events

HDFS

Teradata

filestack

RDBMS

CED/Claims

Clinical Data

Teradata

RDBMS
The master dataset is the only part of the Lambda Architecture that absolutely must be safeguarded from corruption. So for this reason the fault tolerance and redundancy of HDFS adds tremendous support. The master dataset is also referred to as “Raw Data” or “Bronze Data” in our reference architecture and wiki in general. Information created in the serving and speed layers is also referred to as “Silver Data.”

Credit Nathan Marz - Big Data
Hadoop and Big Data

• Big Data = Hadoop + Relational + other suitable task related technologies

• Hadoop is complementary
What we accomplished?

• Evangelized Hadoop

• Linked Hadoop to BI Tools

• A fail fast iterative analytics approach
Introduced Data Lake

- **Data Warehouse**
  - Data organized into a single consistent schema
  - Analysis done on the curated data

- **Data Lake**
  - Data in raw form (structured or unstructured)
  - Schema-less
  - All your data
  - Immutable

- **Data Swamp**
  - Danger! Your lake is a dumping ground for data
Data Lake vs. Warehouse

With a **data warehouse**, incoming data is cleaned and organized into a single consistent schema before being put into the warehouse...

...analysis is done directly on the curated warehouse data

With a **data lake**, incoming data goes into the lake in its raw form...

...we select and organize data for each need

Credit Martin Fowler
Real World Problems - Real-Time Analytics

• Today, most operations in our Big Data ecosystem are of a batch nature
  • What about data that arrives in real-time?
  • Need to do analysis proactively and not reactively
  • Harness data in real-time
  • Growing volumes of “continuous data” from many sources
  • From many sources to many target systems in different formats

• Systems/Applications have high throughput and low latency needs
  • Data ingestion at Hadoop scale

• Real-time data streams and eventing system

• Analytics in real-time on data that arrives in real-time

• Capture all interaction events
  • Real-time click stream/event data from portals
Kafka Benefits

- Near real-time access to data + a view of history = all data
- Kafka fits our existing Hadoop ecosystem
- Scales horizontally as needs grow
- Cloudera supports Kafka - Monitoring and Services
- Cost Efficient/Effective - Need a small modestly sized cluster of Kafka brokers to handle millions of messages per second (read/write)
- Supports Lambda architecture — foundation of our Data Lake
- Popular pattern for data movement and feeding multiple downstream systems using the same data pipe
- Supports Multi-datacenter deployments
What’s out there?

Stream Processors

STORM

samza

Spark

akka
Spark for Big Data

General purpose computational framework that substantially improves on mapreduce

Spark Core

- Spark SQL structured data
- Spark Streaming real-time
- MLlib machine learning
- GraphX graph processing

Easier
Rich APIs

Faster
In-Memory processing and caching

- Standalone Scheduler
- YARN
- Mesos

Powerful
Supports different workloads:
- Batch
- Streaming
- ML
- Graph
Flume is a reliable, distributed service for ingesting textual data, typically log files, into Hadoop. Spark Streaming enables the creation of real-time complex event processing architectures.
Blog on Tuning Spark Streaming App

- Onboard WebClick events into the cluster
  - Use Kafka as a message exchange medium
  - Use Flume Agents as Kafka producers
- Use Spark Streaming application to read Kafka queue asynchronously
  - Parse the semi-structured events
- Use Spark DataFrames to enrich the events with reference data from a large (125 million records) Apache Hive table.
  - The Hive table is cached in memory for faster operations
- Use Spark DataFrame API to write out processed information back into Hive Tables
- Use Apache Parquet as the storage format for the final data
- Use low level Restful api to Serve up as a JSON response over HTTP to enterprise services.
- Use Impala ODBC and Impyla to expose the data to Business Users
Blog on Tuning Spark Streaming App

Embrace the Most Important Change: \textit{Culture}

\textit{Democratize your data and reap the benefits!}
Mix Your Paradigms

Data Driven Solutions + FP

“Functional Programming: I came for the concurrency, but I stayed for the Data Science”
Dean Wampler
Questions?