Tutorial Big Data Analytics in PHM
2016 Conference of the PHM Society
Denver, CO

John Patanian
October 3, 2016

Imagination at work
TUTORIAL AGENDA

- Introduction
- Big Data and PHM Architecture
- Key Components of Apache Hadoop
- General Analytics Patterns (Streaming, Batch, Ad-Hoc)
- Tips and Tricks
- Sample Analysis – Using PHM 2008 Challenge Data Set
- Where to Go Next To Learn More
BIG DATA ANALYTICS

TUTORIAL GOALS

After this tutorial, you should be able to…

• Describe briefly components like Kafka, Hive, HDFS, MapReduce, Hive.

• Understand Streaming, Batch, and Interactive PHM Use Cases for PHM

• Understand differences in writing deploying analytics for the desktop vs. the Hadoop Cluster

This is not …

• A tutorial on Deep Learning

• A detailed tutorial on programming in Python.
Business Case For Big Data Analytics

How important is Big Data analytics relative to other priorities in your company?

<table>
<thead>
<tr>
<th>Industry</th>
<th>Top/highest priority</th>
<th>Within the top three priorities</th>
<th>Within the top five priorities</th>
<th>Not a priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aviation</td>
<td>61%</td>
<td>29%</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>45%</td>
<td>45%</td>
<td>6%</td>
<td></td>
</tr>
<tr>
<td>Power Generation</td>
<td>31%</td>
<td>63%</td>
<td>6%</td>
<td></td>
</tr>
<tr>
<td>Power Distribution</td>
<td>28%</td>
<td>56%</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>Oil &amp; Gas</td>
<td>31%</td>
<td>56%</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>Rail</td>
<td>40%</td>
<td>47%</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>Manufacturing</td>
<td>42%</td>
<td>45%</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>Mining</td>
<td>24%</td>
<td>55%</td>
<td>18%</td>
<td></td>
</tr>
</tbody>
</table>

* GE / Accenture Industrial Internet Insights Report for 2015
The Business Case For Big Data Analytics

<table>
<thead>
<tr>
<th>Priorities: 1-3 years</th>
<th>Aviation</th>
<th>Wind</th>
<th>Power Generation</th>
<th>Power Distribution</th>
<th>Oil &amp; Gas</th>
<th>Rail</th>
<th>Manufacturing</th>
<th>Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase profitability through improved resource management</td>
<td>61%</td>
<td>71%</td>
<td>56%</td>
<td>59%</td>
<td>56%</td>
<td>67%</td>
<td>58%</td>
<td>55%</td>
</tr>
<tr>
<td>Gain a competitive edge</td>
<td>58%</td>
<td>55%</td>
<td>53%</td>
<td>69%</td>
<td>50%</td>
<td>50%</td>
<td>76%</td>
<td>48%</td>
</tr>
<tr>
<td>Improve environmental safety and emissions</td>
<td>39%</td>
<td>61%</td>
<td>50%</td>
<td>75%</td>
<td>59%</td>
<td>43%</td>
<td>52%</td>
<td>58%</td>
</tr>
<tr>
<td>Gain insights into customer behaviors, preferences and trends</td>
<td>58%</td>
<td>61%</td>
<td>47%</td>
<td>56%</td>
<td>38%</td>
<td>60%</td>
<td>70%</td>
<td>39%</td>
</tr>
<tr>
<td>Gain insights into equipment health for improved maintenance</td>
<td>55%</td>
<td>48%</td>
<td>34%</td>
<td>56%</td>
<td>47%</td>
<td>73%</td>
<td>67%</td>
<td>39%</td>
</tr>
<tr>
<td>Drive operational improvements and workforce efficiencies</td>
<td>42%</td>
<td>48%</td>
<td>41%</td>
<td>72%</td>
<td>44%</td>
<td>53%</td>
<td>55%</td>
<td>64%</td>
</tr>
<tr>
<td>Create new business opportunities with new revenue streams</td>
<td>45%</td>
<td>61%</td>
<td>34%</td>
<td>53%</td>
<td>47%</td>
<td>40%</td>
<td>52%</td>
<td>58%</td>
</tr>
<tr>
<td>Meet or exceed regulatory compliance</td>
<td>32%</td>
<td>39%</td>
<td>41%</td>
<td>63%</td>
<td>50%</td>
<td>33%</td>
<td>39%</td>
<td>39%</td>
</tr>
</tbody>
</table>

* GE / Accenture Industrial Internet Insights Report for 2015
PHM at the core of IOT for Industrial Equipment

2016 Annual Conference of the Prognostics and Health Management Society

BIG DATA ANALYTICS

* GE / Accenture Industrial Internet Insights Report for 2015

Failure Histories
Geo-Spatial Tracking
Component Tracking
Model Database
OEM Model Reliability Info
RCM Analysis Info
Root Cause Analysis Info
Spare Part Analysis Info

RCM Analysis Info
Model Database
Component Tracking
Geo-Spatial Tracking
Failure Histories

Open Reliability Management
Open Anticipatory Logistics Management
Open Maintenance Management

RCM Analysis Info
Open Reliability Management
Open Anticipatory Logistics Management
Open Maintenance Management

Calibration & Config. Mgmt
Reactive Maintenance
Condition-Based Maintenance

Pre-Planned Work Packages
Work Order Tracking
MRO Materials
MRO Tools

Preventive Maintenance
Open Object Registry Management

MRO Labor
MRO Material

Source: Penn State ARL Lab
Analytic Patterns – Ad-hoc

• Interactive from a prompt, REPL environment (IDE) or Notebook
• Data & method exploration, data vending, etc., simple BI tasks. Users want an desktop-like experience.
• Typical expected response in sub-seconds to a few seconds.
Analytic Patterns - Streaming

• Low latency data refresh, analytics processing, end-to-end response
• Typically almost everything in memory.

• Need Results in seconds or milliseconds

  – Decision Support
  – Operator Guidance
  – Control or Human In the Loop
Analytic Patterns - Batch

- Scheduled or Event Driven
- Slower Response Times are Acceptable
- Large Volumes of Data
- Can be used to develop pre-defined query results for serving BI presentation layer.
CBM and Big Data Architectures

Advisory Generation (AG)
Prognostics Assessment (PA)
Health Assessment (HA)
State Detection (SD)
Data Manipulation (DM)
Data Acquisition (DA)

* source mimosa.org
Common Analytic Tasks

Data Exploration:
• Visual Exploration
• Descriptive Statistics
• Correlation Analysis
• Domain Specific Analyses

Often a very interactive, iterative process. Ideal to have the same level of interactivity regardless of data size.
Common Analytic Tasks – Model Development

- Train many models on the same set of features.
- Build multiple models with different features.
- Run a DOE on a single mode with different “knobs”.
- Map by operating regime and train one or more models per regime.

These operations are time consuming on a desktop because code are most often written in one or more for loops … a great opportunity for Model Parallelism.
Common Analytic Tasks

Data Preparation:

• Loading data set(s)
• Filtering
• Merging, Joining, Aggregating, Selecting
• Dealing with Data Quality
  • Missing Values, Outliers, Input errors
  • Imputation
• Feature Calculation, Extraction

Many “Embarassingly Parallel” operations directly translate between desktop and cluster
What is Apache Hadoop?

**Definition:** An open-source software ecosystem to store and process data that is too big for one device or server.

- Hadoop scales across tens to thousands of *commodity* servers that don’t share memory or disk space.
- Hadoop manages hardware reliability through redundancy and software.
- Processing happens close to the data whenever possible.
What is Apache

Hadoop: Two Foundational Components

• **Hadoop Distributed File System**: Resilient, high-throughput clustered storage.

• **MapReduce**: Distributed, fault-tolerant resource management & scheduling with a scalable data programming abstraction.
What is Apache Hadoop?

**Key Components to Know for Analytics**

**HDFS:** The scalable fail-safe distributed file system. Designed to store large amounts of data (TB to PB) and scale to many users.

**Hive:** Original SQL on Hadoop. Define schema on distributed files (text, CSV, compressed) in HDFS. (Similar components are Impala, Drill, HAWQ).

**MapReduce:** Legacy data processing architecture. Main API methods are Map, Reduce. Originally only batch. Latest enhancements are manageable for some interactive use.

**Spark:** Latest generation data processing architecture in Hadoop. Offers 10-100x speed improvements over MapReduce, API in Java, Scala, Python, and R. Handles batch, streaming and interactive.

**Sqoop:** Utility for Moving data between external databases, Hive, and HDFS.
What is Apache Hadoop?

**GOVERNANCE INTEGRATION**

- Data Lifecycle & Governance
  - Falcon
  - Atlas

**Data workflow**

- Sqoop
- Flume
- Kafka
- NFS
- WebHDFS

**TOOLS**

- Hue
- Zeppelin
- Ambari User Views

**DATA ACCESS**

- Batch (Map Reduce)
- Script (Pig)
- SQL (Hive)
- NoSql (HBase, Accumulo, Phoenix)
- Stream (Storm)
- Search (Solr)
- In-Mem (Spark)
- Others (HAWQ, Partners)

- YARN: Data Operating System

- HDFS: Hadoop Distributed File System

Hadoop® Reference Architecture

Lambda architecture designed to support batch, streaming and BI

* source mapr.com
Where this does infrastructure work for PHM?

- Centralized Monitoring
- System Level, Fleet-Level Analysis and Operational Guidance
- Maintenance Recommendations
- Integration of pre-processed data from edge devices.
- Combined Data Sets

- High Frequency data set transfer (latency due to bandwidth is a bottleneck).
- Better scenario is edge processing of raw data into features.
Key Components
The ubiquitous Word Count Example

Translation of complex operations into Map and Reduce Operations is non-trivial
Apache Hive Overview

- Built on top of Hadoop
- Data stored in HDFS
- Hive compiles SQL-like HiveQL into MapReduce jobs (Hive 1.0), executes in parallel and returns results.

- Still makes up the vast majority of MapReduce job executions.
- Because of the overhead of MapReduce, even small jobs take at least a minute to return results, making it difficult for interactive use.
- No INSERT OR UPDATE

- Hive 2.0 Performance gains by replacing MapReduce as the Execution engine.
- JDBC and ODBC drivers are available to integrate with BI and desktop analysis tools.
Apache Sqoop

Used for moving data between databases, HDFS and HIVE. Works with any database with a JDBC Driver.

Example SQOOP Command

# -P command will prompt the user for a password at the prompt
# -m 1 does sequential import of the data (not require a primary key) (this uses a single mapper)
# --hive-import will automatically create a hive table

```
sqoop import
   --connect 'jdbc:sqlserver://1.23.456.78;DatabaseName=PHMDemo' \
   --table DataChallenge \
   --fetch-size=10000 \
   --username '<EnterUserName>' \
   -P \
   -m 1 \
   --hive-import
```
Apache Kafka is a distributed streaming platform. It lets you...

• Publish and subscribe to streams of data like a messaging system.
• Store streams of data in a distributed, replicated cluster.
• Process streams of data in real-time.
• Typically used at front of Lambda architecture.
Key Components – Data Processing

**Apache Spark™** is a fast and general engine for large-scale data processing.

APIs in Java, Python, R, and Scala

- **Spark SQL**
- **Spark Streaming**
- **MLlib** (machine learning)
- **GraphX** (graph)

**Brief History of Spark**
2002 – MapReduce @ Google
2004 – MapReduce paper
2006 – Hadoop @ Yahoo
2008 – Hadoop Summit
2010 – Spark paper
2011 – Hadoop 1.0 GA
2014 – Apache Spark top-level
2014 – 1.2.0 release in December
2015 – 1.3.0 release in March
2015 – 1.4.0 release in June
2015 – 1.5.0 release in September
2016 – 1.6.0 release in January
2016 – 2.0.0 Release in July

**Graph**

**Interest over time**

United States. 2004 - present
Key Components – Data Processing

Much more extensive API than MapReduce AND at a higher level of abstraction

Spark Context
DataFrame
Spark SQL
Resilient Distributed Dataset (RDD)
Spark MLLib
from pyspark import SparkContext
from pyspark.sql import HiveContext
sc = SparkContext()

database_names = sqlContext.sql("SHOW DATABASES") # No data is transferred
database_names.collect() # Data is transferred when an action method is called.
database_names.take(2)

# The Result below is a data frame
phm_data = sqlContext.sql("SELECT * FROM PHMDATA WHERE SETTING1 > 0 AND SETTING2 > 10")
A hands on example

Processing the 2008 PHM Data Challenge Data Set
See github repository for Jupyter Notebooks


This data set is not very big, but is used for demonstration purposes.
Tips and Tricks

• When data fits into memory, don’t use a distributed processing paradigm.

• Understand what you lose by sampling your data set.

• Common scenario: Filter, column selection, feature calculations and extraction done using Big Data processing, often data set may no longer be big.
Tips and Tricks

• Transition between desktop to cluster execution is often challenging, still requires a change of thought process.
• Some algorithms don’t readily translate to a distributed version.
• Process is greatly simplified if your code is written into modular functions (Don’t write everything in one big function). If operation can be applied to each element of a row or column, or to an entire row or column, translation is straight-forward.
• Thought process about how to enable a calculation on a distributed framework is similar to thought process of how to vectorize an operation.
• Iterative operations are easier with Spark than with MapReduce, but still not a direct translation from Desktop to cluster.
How to Learn More

• DataBricks (i.e. the creators of Spark) have free accounts that enable execution on a small cluster running within AWS.

• The edX MOOC platform offers a series of online courses, ranging from Introductory to Advanced Machine Learning on Spark. These series of courses were created by Databricks Download a version of Hadoop or Spark and run on your Desktop. The Hortonworks Sandbox is a good Hadoop setup with Images available for several virtual machines. Your datasets will be limited to the size of your computer, but you will learn the API.

• Recommended approach is to take a problem you have already solved and see if you can replicate it in Spark.
Demo of Sample Non-Spark Analysis

https://github.com/patanijo/PHM2016/blob/master/Sample_Analysis_Notebook_Non_Spark.ipynb

**Homework**: Repeat this analysis using Spark (on a local machine or an available cluster). Solutions will be posted later this week on the same repository.
Backup
Instructions for Installing Apache Spark Standalone on OS/X

Follow Instructions Here:

https://medium.com/data-science-cafe/apache-spark-1-6-0-setup-on-mac-os-x-yosemite-d58076e8064e?swoff=true#.qqpb3kikb

After following instructions execute the following command:

echo "127.0.0.1 $HOSTNAME" | sudo tee -a /etc/hosts

Command to start with IPYTHON
Instructions for Installing Apache Hadoop on Windows

Recommended to Install on a VM Such as VirtualBox or VMWare

https://www.virtualbox.org/wiki/Downloads

For example link to HortonWorks Sandbox downloads:

http://hortonworks.com/downloads/#sandbox

This installs a complete stack of Hadoop tools.
Pandas to Spark

Converting between pandas and Spark DataFrames:

https://databricks.com/blog/2015/08/12/from-pandas-to-apache-sparks-dataframe.html

https://medium.com/@chris_bour/6-differences-between-pandas-and-spark-dataframes-1380cec394d2#.5eme3r4lw

https://lab.getbase.com/pandarize-spark-dataframes/