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

TUTORIAL GOALS

After this tutorial, you should be able to…

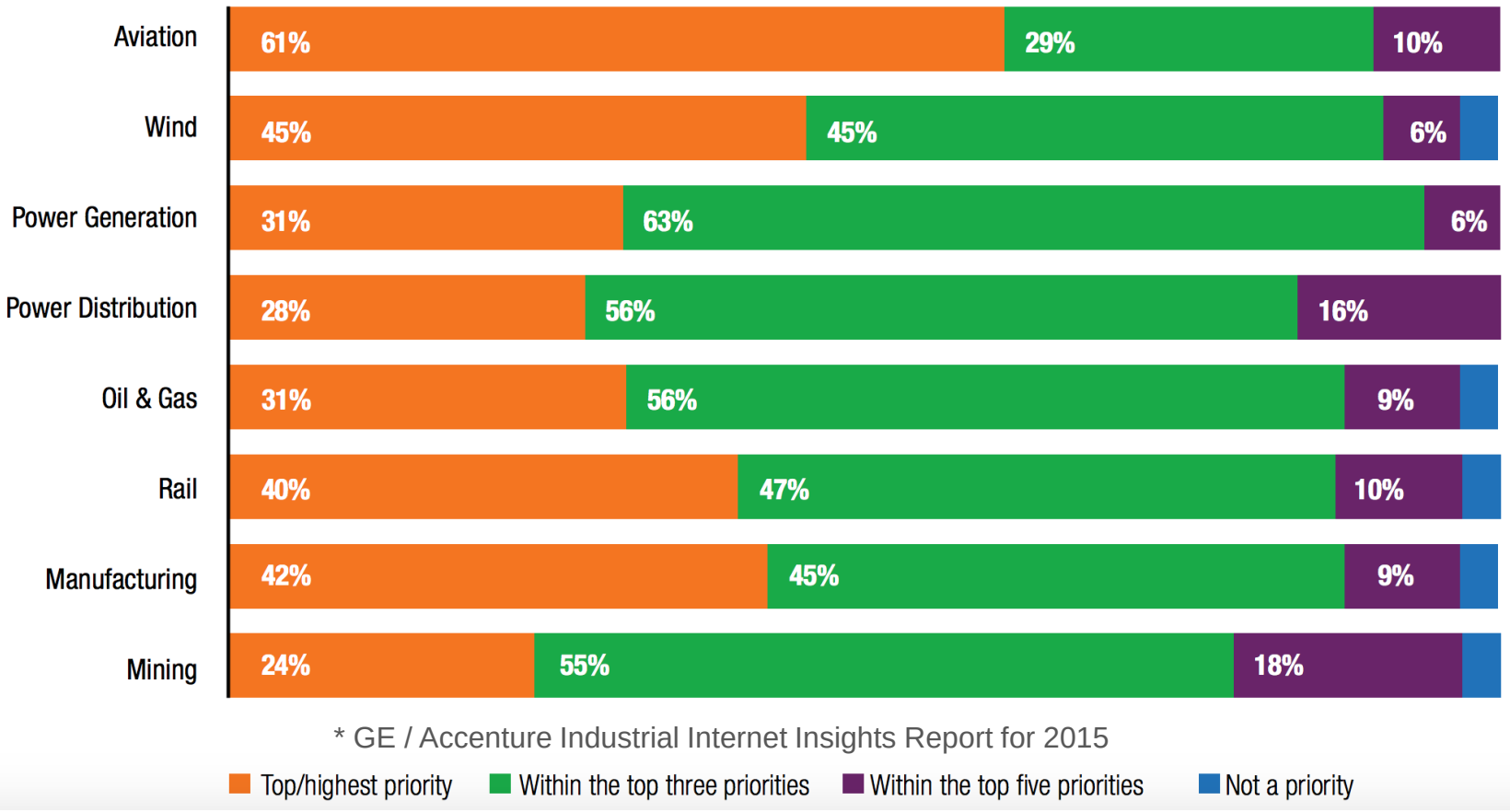
- Describe briefly components like Kafka, Hive, HDFS, MapReduce, Hive.
- Understand Streaming, Batch, and Interactive PHM Use Cases for PHM
- Understand differences in writing deploying analytics for the desktop vs. the Hadoop Cluster

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?



The Business Case For Big Data Analytics

Highest-ranked priorities

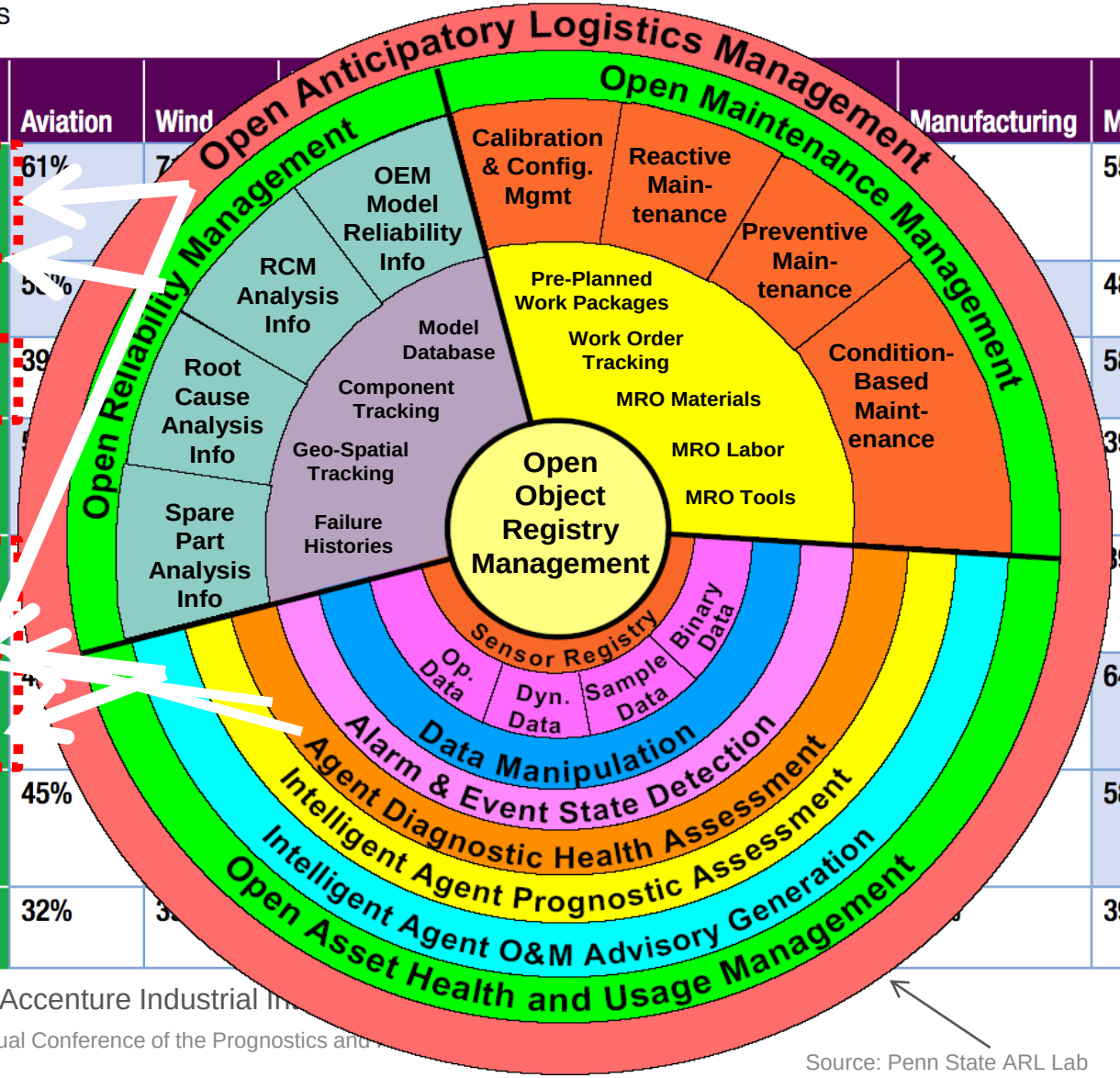
Priorities: 1-3 years	Aviation	Wind	Power Generation	Power Distribution	Oil & Gas	Rail	Manufacturing	Mining
Increase profitability through improved resource management	61%	71%	56%	59%	56%	67%	58%	55%
Gain a competitive edge	58%	55%	53%	69%	50%	50%	76%	48%
Improve environmental safety and emissions	39%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

* GE / Accenture Industrial Internet Insights Report for 2015

PHM at the core of IOT for Industrial

Highest-ranked priorities

Priorities: 1-3 years	Aviation	Wind	Manufacturing	Mining
Increase profitability through improved resource management	61%	7		55%
Gain a competitive edge	5 %			48%
Improve environmental safety and emissions	39			58%
Gain insights into customer behaviors, preferences and trends				39%
Gain insights into equipment health for improved maintenance				39%
Drive operational improvements and workforce efficiencies	4			64%
Create new business opportunities with new revenue streams	45%			58%
Meet or exceed regulatory compliance	32%	3		39%



Source: Penn State ARL Lab

* GE / Accenture Industrial In

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

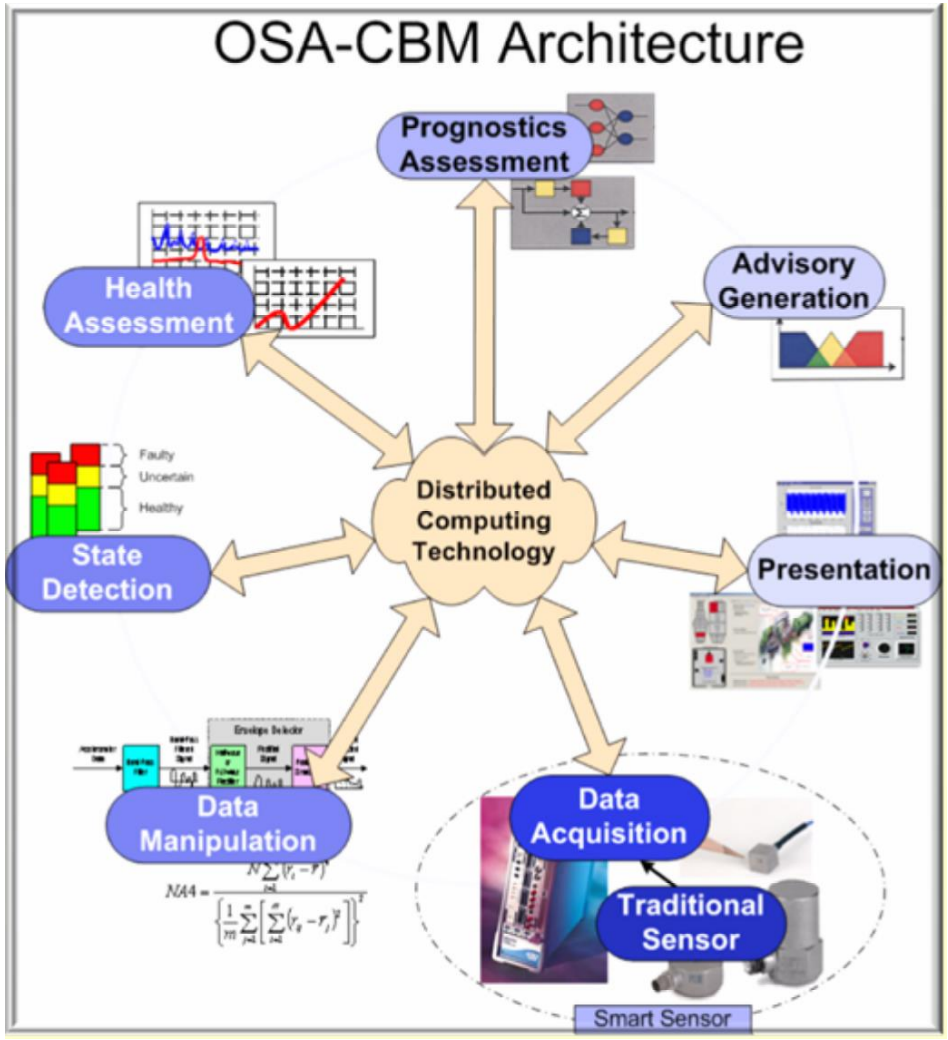
Analytic Patterns - Batch

- 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:

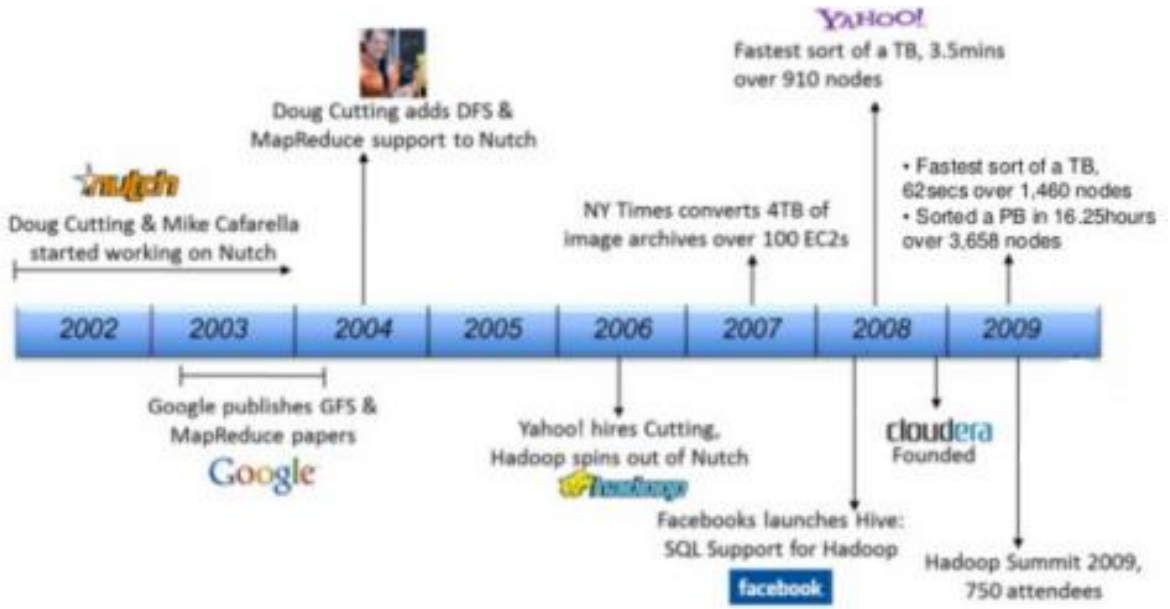
- Loading data set(s)
- Filtering
- Merging, Joining, Aggregating, Selecting
- Dealing with Data Quality
  - Missing Values, Outliers, Input errors
  - Imputation
- Feature Calculation, Extraction

Many “Embarassingly Parallel” operations directly translate between desktop and cluster

What is Apache Hadoop?

Definition: An 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 through redundancy and software.
- Processing happens close to the data whenever possible.



What is Apache 

Hadoop : Two Foundational Components

- **H**adoop **D**istributed **F**ile **S**ystem: 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.

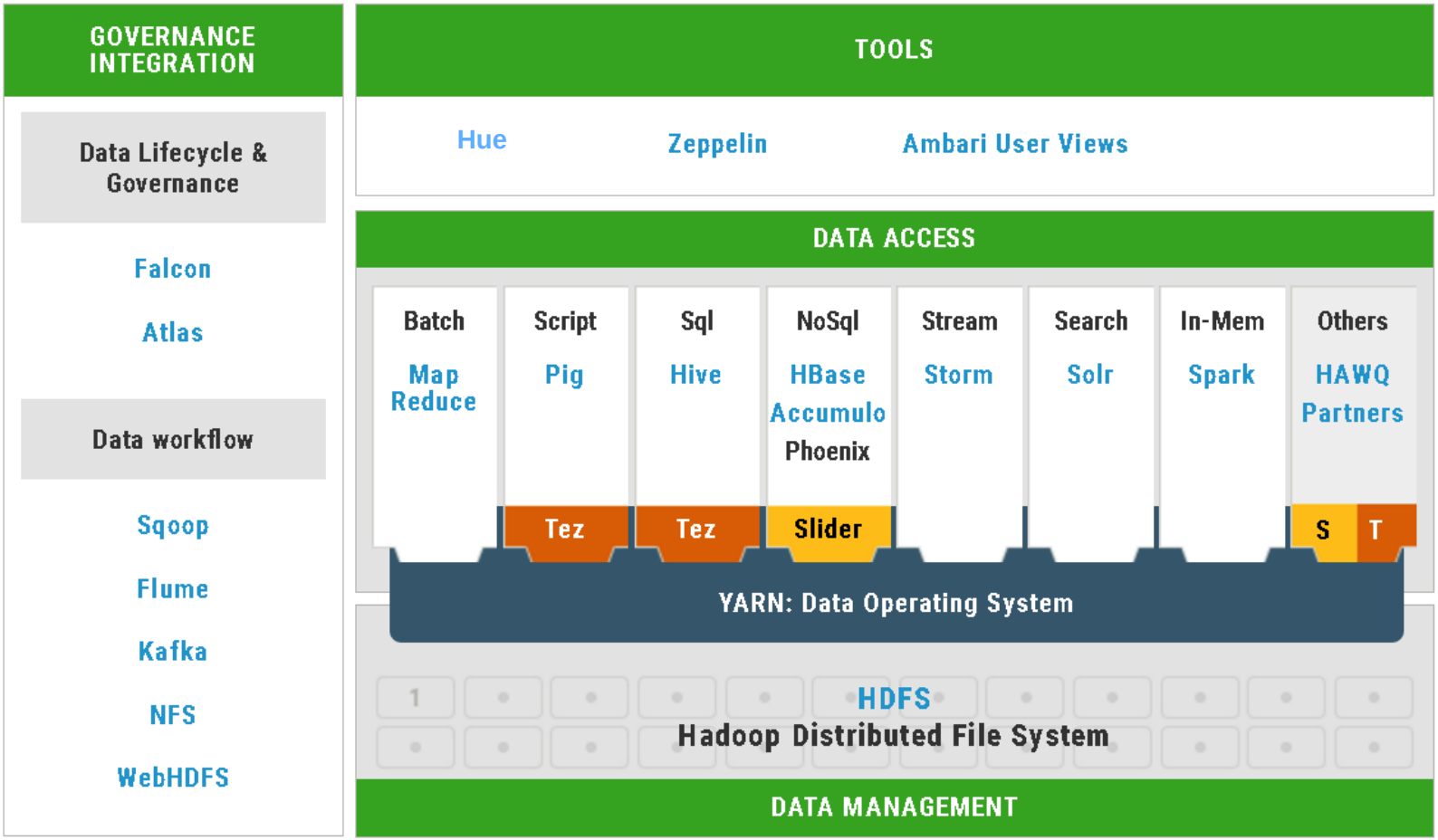
Hive: Original SQL on Hadoop. Define schema on distributed files (text, CSV, compressed) in HDFS. (Similar components are Impala, Drill, HAWQ).

MapReduce: Legacy data processing architecture. Main API methods are Map, Reduce. Originally only batch. Latest enhancements are managable 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

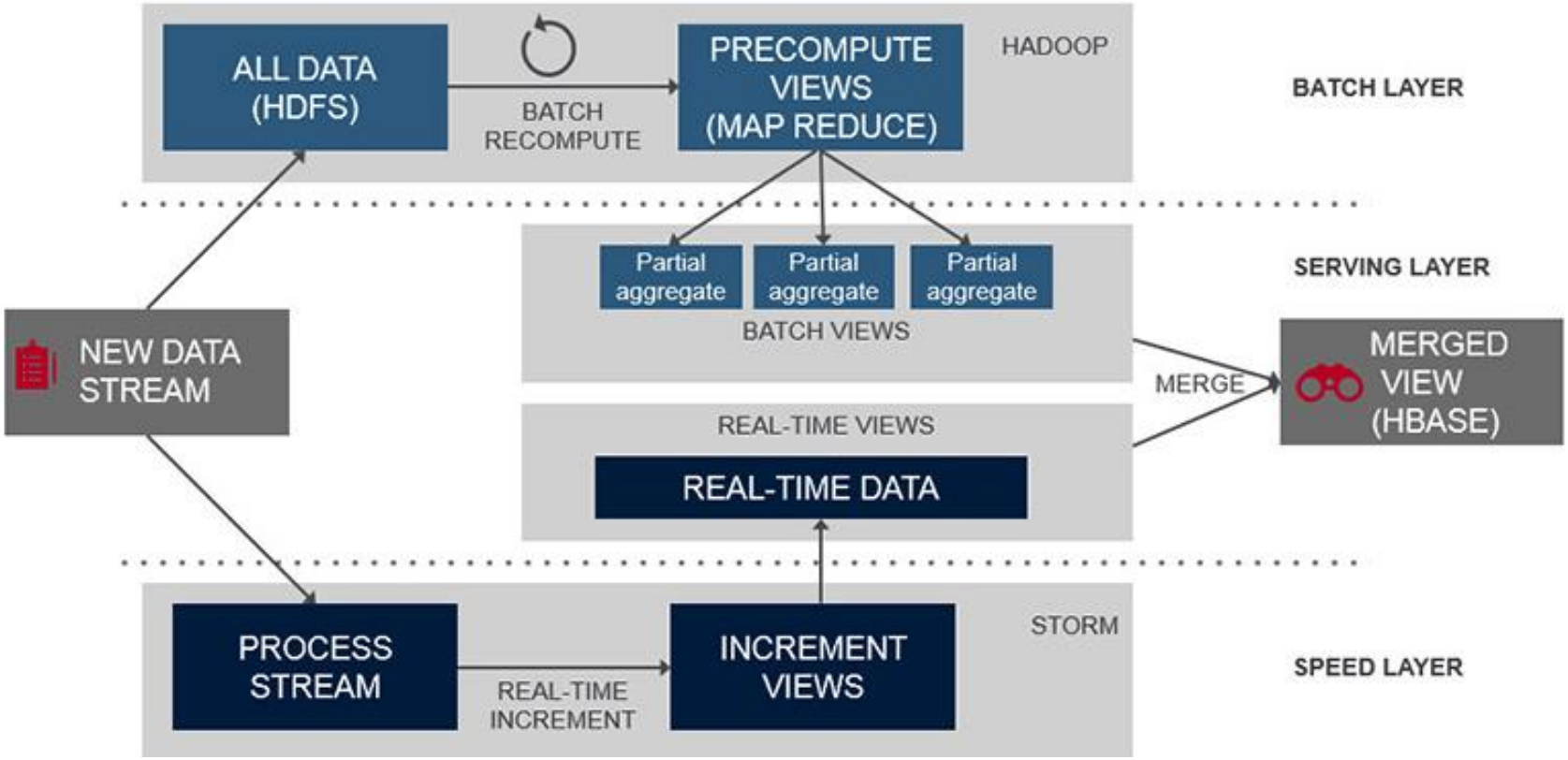
What is Apache Hadoop?



source: http://hortonworks.com/products/data-center/hdp/

Hadoop® Reference Architecture

Lambda architecture designed to support batch, streaming and BI



* source mapr.com

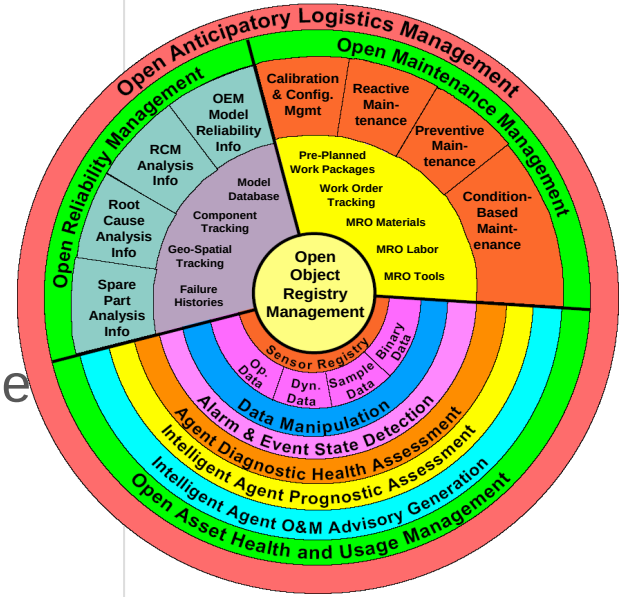
Where this does infrastructure work for PHM?

GOOD

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

MAYBE

- High Frequency data set transfer (latency due to bandwidth is a bottleneck).
- Better scenario is edge processing of raw data into features.

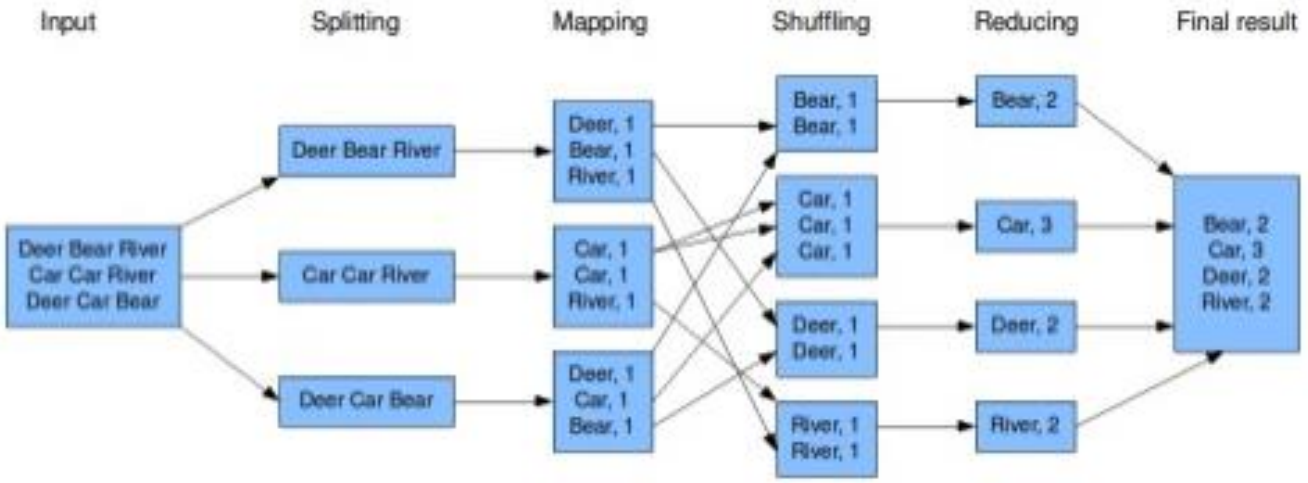


Key Components

Apache Map Reduce

Original Core Data Processing Engine of Hadoop

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.

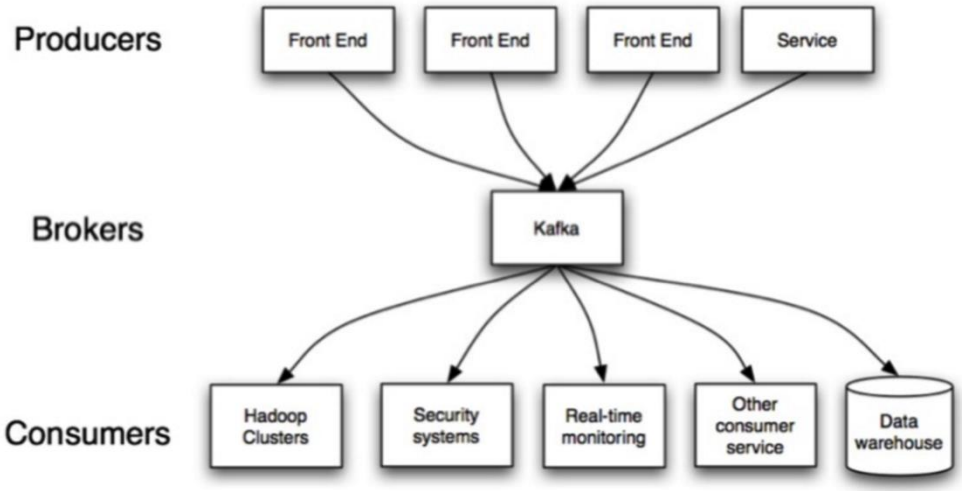
Example SQOOP Command

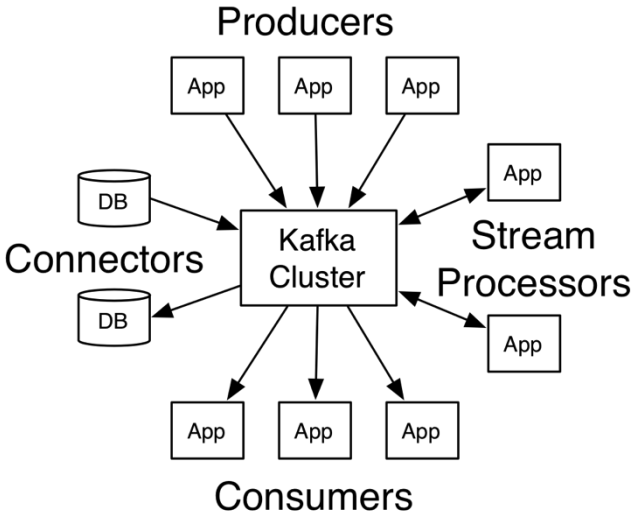
```
# -P command will prompt the user for a password at the prompt
# -m 1 does sequential import of the data (not require a primary key) (this uses a single mapper)
# --hive-import will automatically create a hive table

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

A distributed streaming platform.

Kafka decouples data-pipelines





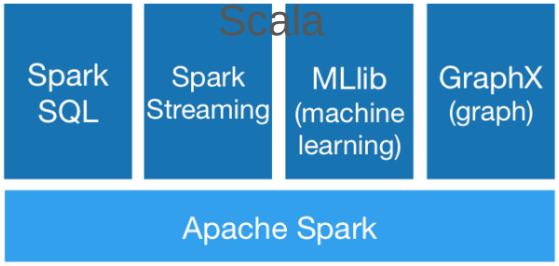
Apache Kafka is a distributed streaming platform. It lets you...

- 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

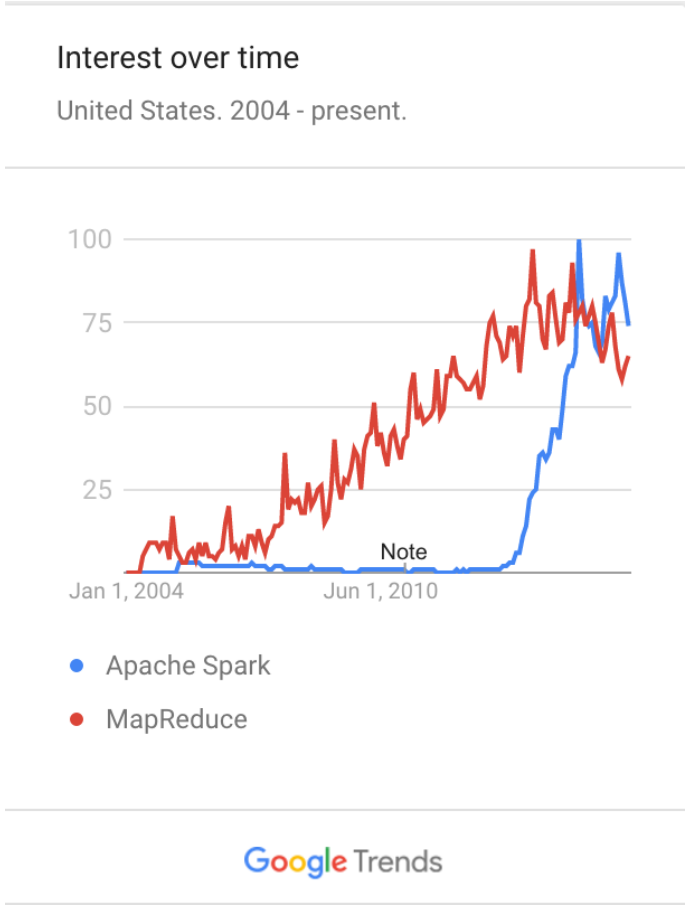
Apache Spark™ is a fast and general engine for large-scale data processing.

APIs in Java, Python, R, and Scala



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



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

Key Components – Data Processing

Spark SQL Example

```
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

https://github.com/patanijo/PHM2016

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/