Hands-on Introduction to Apache Hadoop and Spark

Douglas Eadline

(Part 1)

BASEMENT SUPERCOMPUTING

Presenter

Douglas Eadline
deadline@basement-supercomputing.com
@thedeadline

- HPC/Hadoop Consultant/Writer
- http://www.basement-supercomputing.com

Outline Day 1

- **Segment 1:** Intro and Overview of Hadoop/Spark (40 mins)
- **Segment 2:** Using HDFS (25 mins)
- Break 10 minutes
- **Segment 3:** Running and Monitoring Hadoop Apps (35 mins)
- **Segment 4:** Using Apache Pig (20 mins)
- Break: 10 minutes
- **Segment 5:** Using Apache Hive (30 mins)

Outline Day 2

- **Segment 6:** Running Apache Spark (35 mins)
- Break: 5 minutes
- **Segment 7:** Running Apache Sqoop (30 mins)
- **Segment 8:** Using Apache Flume (20 mins)
- Break: 10 minutes
- **Segment 9:** Introduction and Example Analytics Application using Apache Zeppelin (40 mins)
- Break: 5 minutes
- **Segment 10:** Wrap-up/ Resources/Where to Go Next (20 mins)

Recommended Approach To Class

- Course covers a lot of material!
- Designed to get you started ("hello.c" approach)
- Sit back and watch the examples
- All examples are provided in a notes file
- I will refer to file throughout the class (cut and paste)
- The notes files are available for download along with some help on installing software (URL provided at end of this class)

User Programming Environment

There are four options (URLs provided at end of course):

1. Use the Linux Hadoop Minimal Sandbox virtual machine that can run under VirtualBox or VMWare.
2. Use the Hortonworks Sandbox, a full featured Hadoop/Spark virtual machine that runs under Docker, VirtualBox, or VMWare.
3. Install software from Apache.org (instructions provided) Best for Linux machines/laptops
4. Use a resource available to you: local cluster/cloud

Course Resources

Download all class notes and data files used in the lessons.

Includes directions on how to download and run the Linux Hadoop Minimal Sandbox virtual machine.

https://www.clustermonkey.net/download/Hands-on_Hadoop_Spark

Cluster Used for Class

Limulus™ Desk-side Hadoop/Spark Cluster

- Four/eight motherboards (1p, i5/i7/E3)
- 16/48 cores (48/96 threads)
- 256-512 GB RAM
- 1-32TB HDFS (SSD), 12-20TB HDD
- 1/10 GbE internal
- Hortonworks 2.7/CentOS Linux installed
- Single wall plug (low power)
- Cool and quiet for office/lab/classroom

- http://www.basement-supercomputing.com

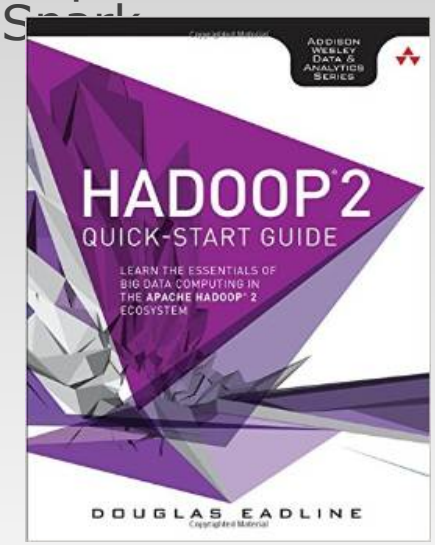
BASEMENT SUPERCOMPUTING

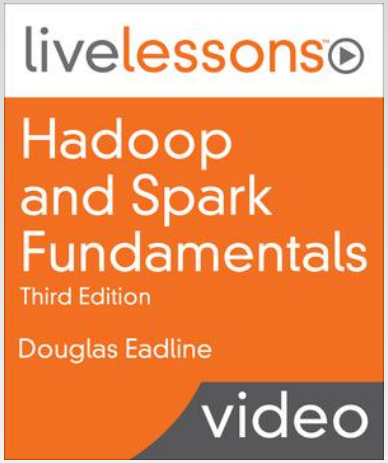
Supporting Materials

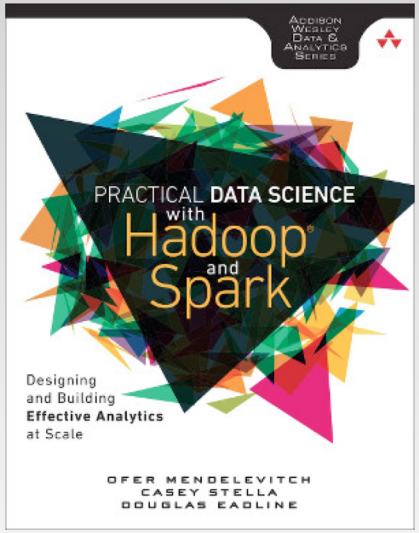
Hadoop 2 Quick-Start: http://www.clustermonkey.net/Hadoop2-Quick-Start-Guide

Hadoop Fundamentals: https://www.safaribooksonline.com/library/view/hadoop-and-spark/9780134770871

Practical Data Science with Hadoop and Spark: http://www.clustermonkey.net/Practical-Data-Science-with-Hadoop-and-Spark







BASEMENT SUPERCOMPUTING

Segment 1

Quick Overview of Hadoop and Spark

BASEMENT SUPERCOMPUTING

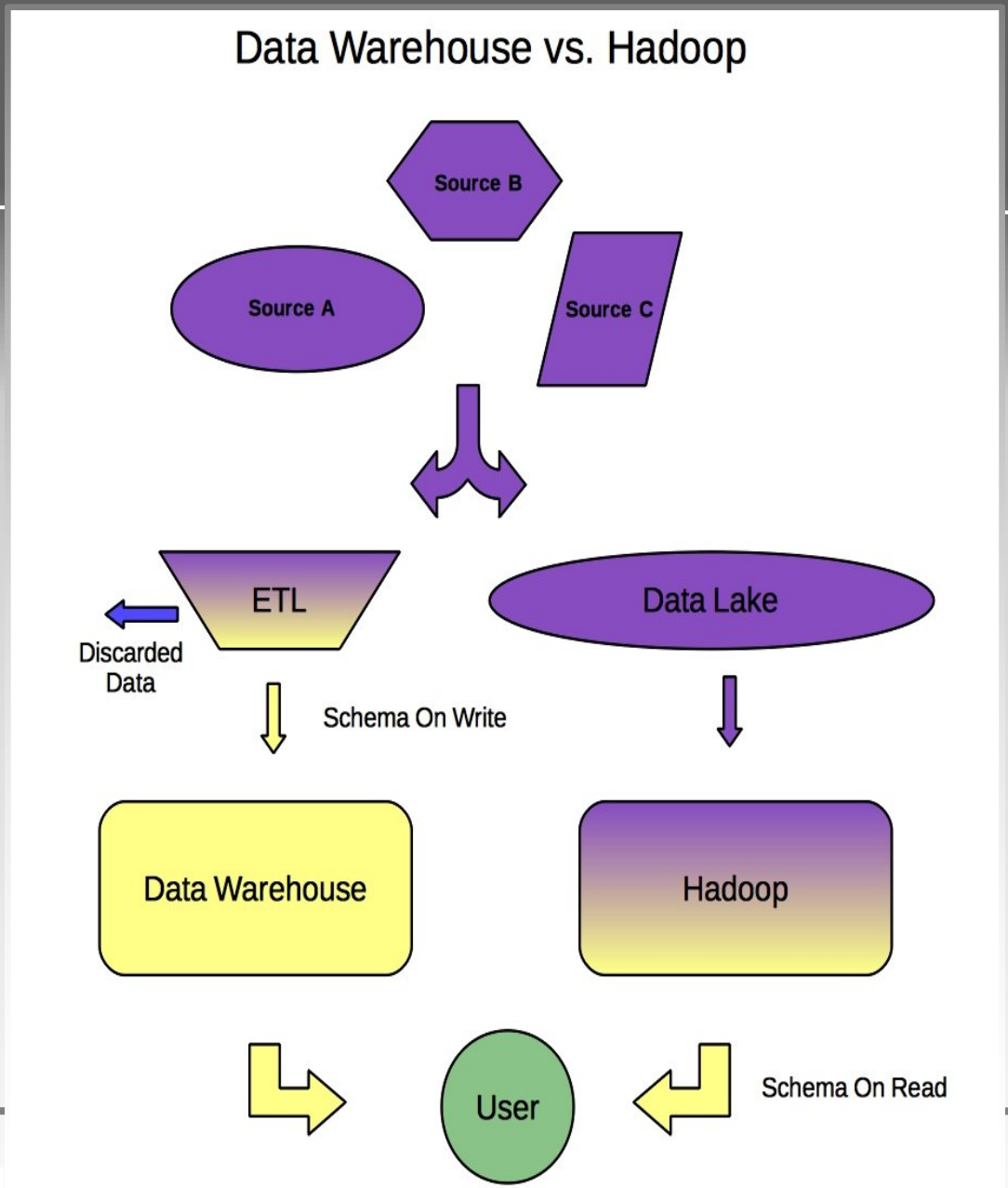
Big Data Analytics

- Three V's (Volume, Velocity, Variability)
- Data that are large (lake), growing fast (waterfall), unstructured (recreation) - not all may apply.
- May not fit in a "relational model and method"
- Can Include: video, audio, photos, system/web logs, click trails, IoT, text messages/email/tweets, documents, books, research data, stock transactions, customer data, public records, human genome, and many others.

Hadoop Data Lake

Data Warehouse applies "schema on write" and has an Extract, Transform, and Load (ETL) step.

Hadoop/Spark applies "schema on read" and the ETL step is part of processing. All input data are placed in the lake in raw form.

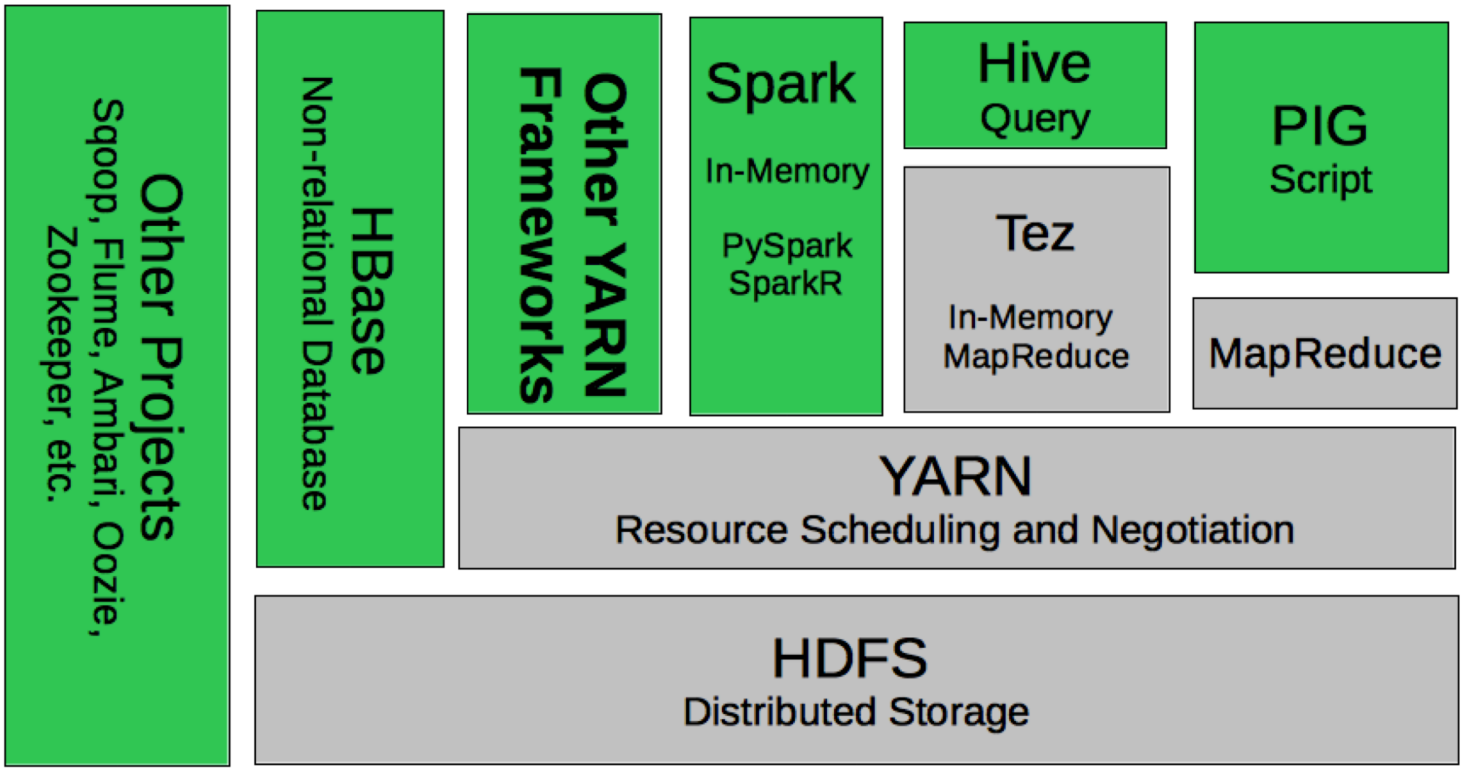


Defining Hadoop (Version 2+)

A PLATFORM for data lake analysis that supports software tools, libraries, and methodologies

- Core and most tools are open source (Apache License)
- Written in Java, but not exclusively a Java platform
- Primarily GNU/Linux, Windows versions available
- Scalable from single server to thousands of machines
- Runs on commodity hardware and in the cloud
- Application level fault tolerance possible
- Multiple processing models and libraries (Frameworks)
- Version 3 released in Dec 2017, no major change for users

Hadoop Components



Hadoop Core Components

- **HDFS – Hadoop Distributed File System.** Designed to be a fault tolerant streaming file system. Default block size is 64 MB vs. 4 KB for Linux ext4. Runs on commodity servers, master "NameNode" and multiple "DataNodes." Data are replicated on multiple servers.
- **YARN – Yet Another Resource Negotiator.** Master scheduler and resource allocator for the entire Hadoop cluster. User jobs ask YARN for resources (containers) and data locality. Provides dynamic resource allocation(run-time). Runs on commodity servers, master "ResouceManager" and multiple "NodeManagers."
- **MapReduce/Tez – YARN Application.** Provides classic MapReduce and optimized MapReduce processing.
- **Cluster servers (nodes) are usually both DataNodes and NodeManagers (Hyperconverged) Move processing to data.**

Apache Components We Will Use

- **Apache Pig** is a high-level "scripting" language for creating MapReduce programs used with Hadoop. Good for ETL.
- **Apache Hive** is an SQL compatible language for data summarization, ad-hoc queries, and the analysis of large datasets
- **Apache Spark** is an easy to use language for writing analytics applications that includes MapReduce, SQL queries and other tools.
- **Apache Sqoop** is a tool for transferring bulk data between Hadoop and relational databases.
- **Apache Flume** is a distributed service for efficiently collecting, aggregating, and moving large amounts of log data.
- **Apache Zeppelin** is web-GUI front end for creating data analytics applications

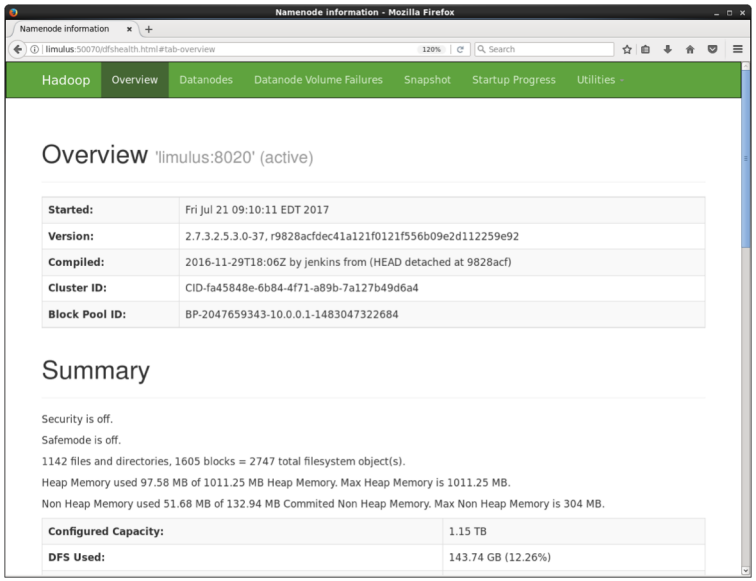
YARN Application Framework: Spark

- **Apache Spark Core -** Spark Core is the underlying general execution engine for spark
- **Spark SQL -** Spark SQL is a component on top of Spark Core that introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data.
- **Spark Streaming -** Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics.
- **MLlib (Machine Learning Library) -** MLlib is a distributed machine learning framework
- **GraphX -** GraphX is a distributed graph-processing framework on top of Spark. It provides an API for expressing graph computation.

Questions ?

Segment 2

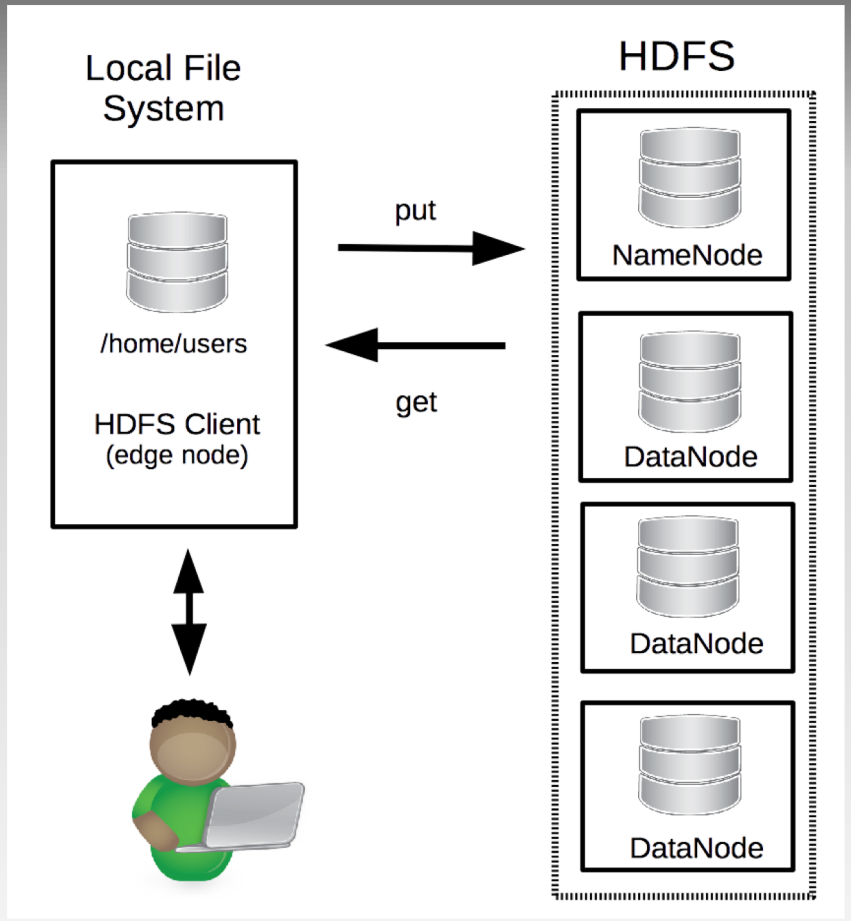
Using the Hadoop Distributed File System



BASEMENT SUPERCOMPUTING

How the User "Sees" HDFS

- HDFS is a separate file system from the host machine
- Data must be moved to (put) and from (get) HDFS
- Hadoop processing happens in HDFS



BASEMENT SUPERCOMPUTING

HDFS Commands

All user HDFS commands start with

```
$ hdfs dfs
```

For instance:

```
$ hdfs dfs –ls
```

Will list files. To get all possible commands:

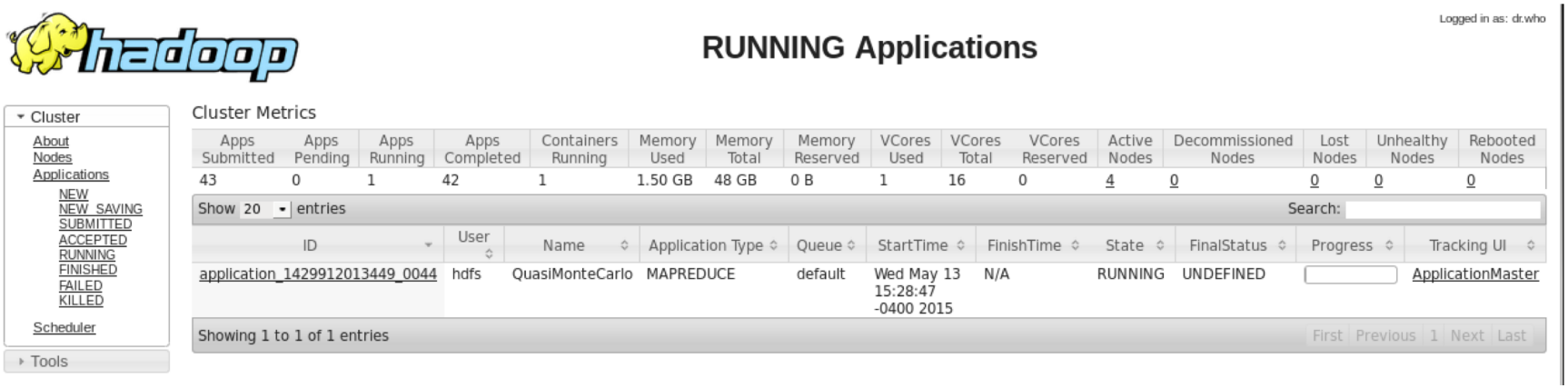
```
$ hdfs dfs
```

User hdfs is “root” for HDFS. Only user hdfs can perform administrative tasks.

10 Minute BREAK

Segment 3

Running and Monitoring Hadoop Applications



BASEMENT SUPERCOMPUTING

What Is Map Reduce?

Map Reduce Is a Simple Algorithm

(and can run in parallel due to no side-effects)

```
grep something | wc –l
```

- Distinct steps and one-way communication
- Map then Reduce
- Uses (key, value) pair
- Works great in many cases (even for some HPC applications), but it is not the only algorithm for Hadoop version 2. It is the basis for many tools including Pig and Hive.

Segment 4

Using Apache Pig

https://pig.apache.org

BASEMENT SUPERCOMPUTING

Apache Pig Background

Apache Pig is a platform for analyzing large data sets. Pig programs are amenable to substantial Map-Reduce parallelization, which in turns enables them to handle very large data sets. Pig's language layer currently consists of a textual language called Pig Latin

With Pig it is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks are easy to write, understand, and maintain. Optimization is available through Apache Tez (it runs fast).

10 Minute BREAK

Segment 5

Using Apache Hive

https://hive.apache.org

BASEMENT SUPERCOMPUTING

Apache Hive Background

Apache Hive provides facilitates for reading, writing, and managing large distributed datasets using using SQL syntax. Hive provides tools to enable easy access to data via SQL. Query execution can be done via Tez (optimized MapReduce) or Apache Spark.

Hive provides standard SQL functionality, including many of the later SQL:2003 and SQL:2011 features for analytics. Hive is not designed for online transaction processing (OLTP) workloads, but is partially ACID compliant (*INSERT, UPDATE*, and *DELETE)*

Questions ?

Continue Tomorrow

BASEMENT SUPERCOMPUTING