Outline of Hadoop

Background, Core Services, and Components

David Schwab

Synchronic Analytics

https://SynchronicAnalytics.com

Nov. 1, 2018

Hadoop Topics

- Hadoop's Purpose and Origin
- Hadoop's Architecture
  - Minimum Configuration
  - Practical Configurations
- Hadoop Components
  - Hadoop Distributed File System (HDFS)
  - Resource Manager (YARN)
  - MapReduce
  - Hive, Sqoop, Spark, and so on…

SYNCHRONIC ANALYTICS

Hadoop Overview

- Hadoop is an architecture for quickly processing very large data sets using commodity hardware:
  - "Quickly": fast enough to be useful
  - "Very Large": Billions of records
  - "Commodity hardware": existing networking servers and equipment
- Hadoop works by:
  - Storing large data sets multiple files for fast loading.
  - Processing these files in parallel for fast compute.
  - Distributing workloads across multiple servers for redundancy and failover

SYNCHRONIC ANALYTICS

Hadoop's Origin Story

Hadoop was first described in a 2003 white paper by the engineer Doug Cutting. By 2006, others had expanded his ideas, and Hadoop 1.0 was released.

The search-engine war provided inspiration:

- Google not yet dominant; competitors such as Yahoo!, Excite, Cha-Cha (!) and AskJeeves (!!) still active.
- Fast, accurate search demands processing very big data sets
- Relational databases are too slow, even in parallel
- Hadoop proposed as a solution

What Hadoop is Not

- Big Data
  - It is one of several Big Data technologies
  - Other technologies may be more appropriate for certain use cases (e.g. fast compute)
- Cloudera Manager, HortonWorks, etc.
  - These are third-party solutions that can make Hadoop more manageable
    - Provide integrated suite of proven components
    - Provide web console for cluster management, job scheduling, etc.
  - The core Apache Hadoop packages are open-source
- Hive, Solr, Spark, etc.
  - These are other Big Data technologies
    - some require Hadoop (e.g. Hive)
    - others can be used with several architectures (e.g. Spark, Solr)

SYNCHRONIC ANALYTICS

Minimum Configuration

- To configure a minimum viable cluster, you need the following
  - Three Linux servers
    - All interconnected w/ root password-less ssh
    - SELinux & iptables disabled
  - A Java distribution
    - Oracle Java 1.7.0_45 is latest tested by Apache
    - https://wiki.apache.org/hadoop/HadoopJavaVersions
  - The Apache Hadoop packages
    - hadoop-x.x.x.targ.gz (e.g. Hadoop-2.9.1.tar.gz) (345M)
    - http://www.apache.org/dyn/closer.cgi/hadoop/common/

SYNCHRONIC ANALYTICS

Minimum Configuration: Environment

- Hadoop-env.sh: Primary Hadoop environment
  - Set Java options
    - JAVA_HOME, classpath, heap size
  - Set configuration/logging folders
- Yarn-env.sh: Resource Manager environment
  - Similar to Hadoop-env.sh, but just for YARN
    - Java options
    - Configuration/logging

SYNCHRONIC ANALYTICS

Minimum Configuration: Site

- Core-site.xml
  - NameNode URI (defaults to hdfs://localhost:9000)
- Hdfs-site.xml
  - NameNode/DataNode directories
    - Local path where HDFS will store physical files
  - File replication
- Yarn-site.xml
  - YARN URI

SYNCHRONIC ANALYTICS

Minimum Configuration: Node

- List hostnames of all nodes in cluster in slaves file
  - karamazov.dmitri.local
  - karamazov.ivan.local
  - karamazov.alyosha.local
- Hadoop daemons will run all commands concurrently on each node.

Hadoop Daemons

- Two daemons control HDFS and YARN, the core Hadoop processes
  - Hdfs-daemon
  - Yarn-daemon
- Hadoop startup
  - /opt/hadoop/sbin/start-dfs.sh
  - /opt/hadoop/sbin/start-yarn.sh

A Practical Hadoop Cluster

- Nodes are enterprise servers
- NameNode & Secondary NameNode
  - One server each
  - Often RAID1 (though not required)
  - Minimal data storage/compute
- DataNodes
  - One server per node
  - Holds data and service-level metadata (e.g. Hive MetaStore)
  - Some services, such as Impala, require dedicated nodes

Sample Configurations

Per Node	Nodes	Cores	RAM (TB)	Drive (TB)
4 cores	15	60	3.8	360
256 GB	21	84	5.4	504
24 TB	29	116	7.4	696

SYNCHRONIC ANALYTICS

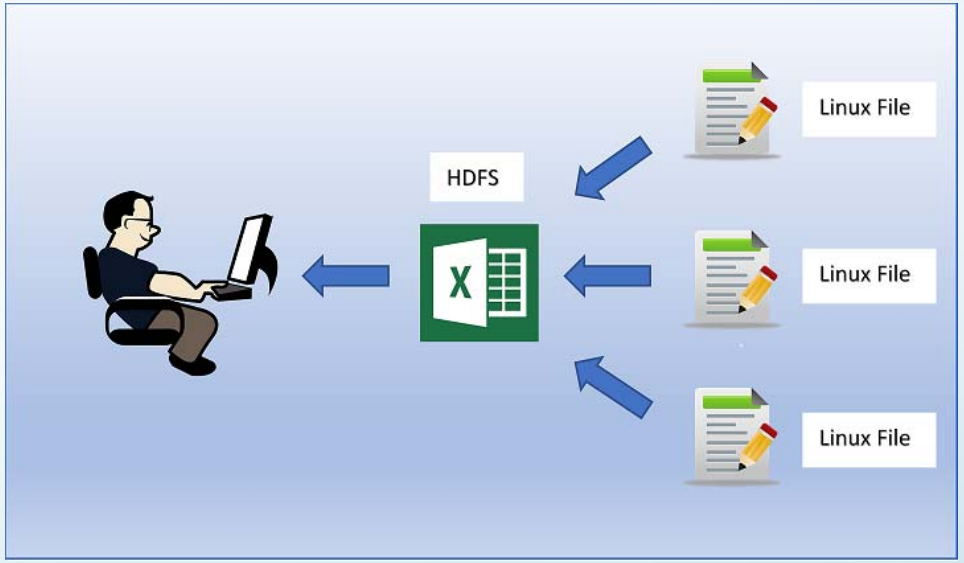
Hadoop Overview

- Hadoop is an architecture for quickly processing very large data sets using commodity hardware:
- Storing large data sets multiple files
  - HDFS (Hadoop Distributed File System)
- Processing these files in parallel
  - MapReduce algorithm

Core Services: HDFS

The Hadoop Distributed File System (HDFS):

- A logical file system that overlays the existing Linux file system
- Large data sets stored as multiple Linux files. . .
- . . .but processed as a single (HDFS) file



SYNCHRONIC ANALYTICS

Core Services: HDFS

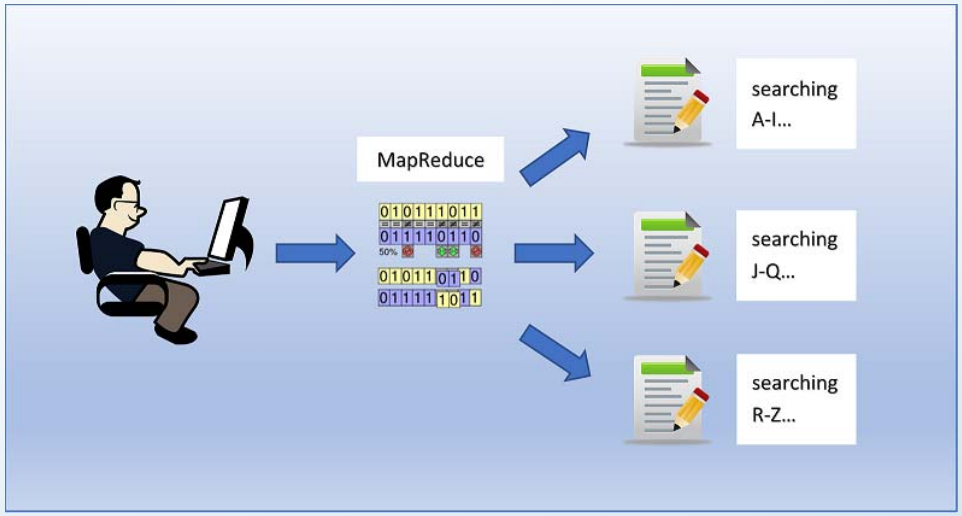
- HDFS access is through the hdfs-daemon.
  - Usage: hdfs dfs <command>
  - Run from Linux CLI
  - Ex. hdfs dfs –ls /user/john lists all files in the /user/john folder
- POSIX-based; commands very much like Linux
- All permissions set locally

SYNCHRONIC ANALYTICS

Core Services: MapReduce

MapReduce is an algorithm that processes large datasets in parallel:

- The program is a Java JAR file
- This is executed against each file block in HDFS
  - The map part applies a function to each file
  - The reduce aggregates the maps to yield the output



SYNCHRONIC ANALYTICS

Core Services: MapReduce

- MapReduce programs may be executed with the *hadoop jar* command:
  - Usage: hadoop jar <jar> [mainClass] args…
  - Run from Linux CLI
  - Ex. hadoop jar wordcount.jar infile outfile
- MapReduce programs may also be executed with YARN:
  - Allows YARN to manage program resources
  - Ex. yarn jar wordcount.jar infile outfile

Core Services: MapReduce

MapReduce Example: Word Count

- Input: Hadoop configuration files
- Output: Total word count

```
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
```

Sample Input File

SYNCHRONIC ANALYTICS

Core Services: MapReduce

MapReduce Example: Mapping Step

- Mapper creates key,value pairs
  - <word>, 1
  - <word>, 1
  - <word>, 1

Licensed	1
under	1
the	1
Apache	1
License	1
Version	1
2.0	1
the	1
"License"	1
you	1
may	1
not	1
use	1
this	1
file	1

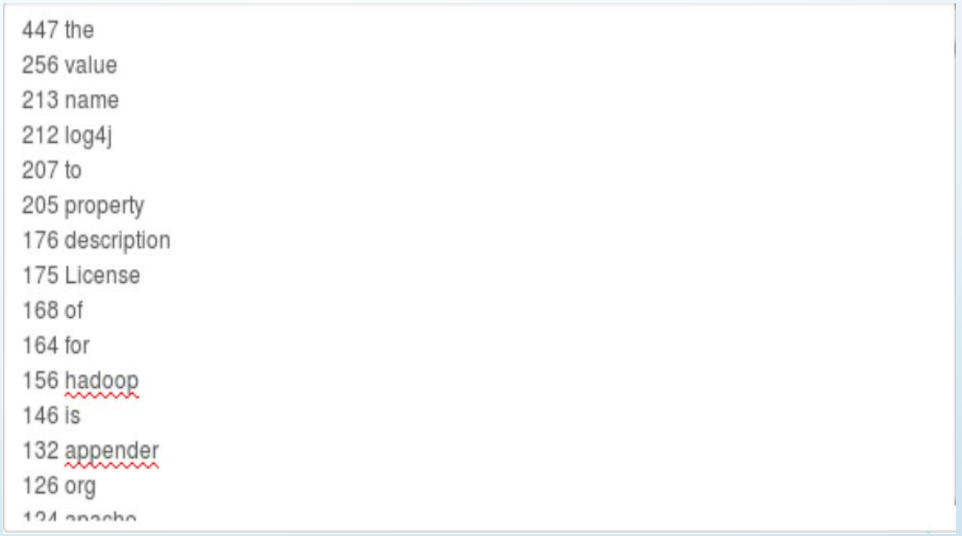
Map Output File

SYNCHRONIC ANALYTICS

Core Services: MapReduce

MapReduce Example: Reducing Step

- Reducer aggregates key, value pairs
  - <the, 447>
  - <value, 256>
  - <name, 213>



Reducer Output File

SYNCHRONIC ANALYTICS

Core Services: MapReduce

- MapReduce can implement many common algorithms:
  - Summarizing data
    - Category counts
    - Category statistics
  - Full-text search
  - Custom business logic

SYNCHRONIC ANALYTICS

Core Services: MapReduce

- MapReduce can implement many common algorithms:
  - Summarizing data
    - Category counts
    - Category statistics
  - Full-text search
  - Custom business logic
- However, implementation can be challenging
  - Programmed in Java using Hadoop API
  - Writing the program is often not straightforward

SYNCHRONIC ANALYTICS

Hadoop Use Cases

- Data Consolidation
  - More efficient use of resources
  - More robust architecture (fewer "moving parts")
  - Less cost overall from hardware & licenses
- Data Capture
  - Expand available pool of data
  - Real-time data capture and updates
- Data Analysis
  - Insight from all your data
  - Discover unexpected correlations and trends

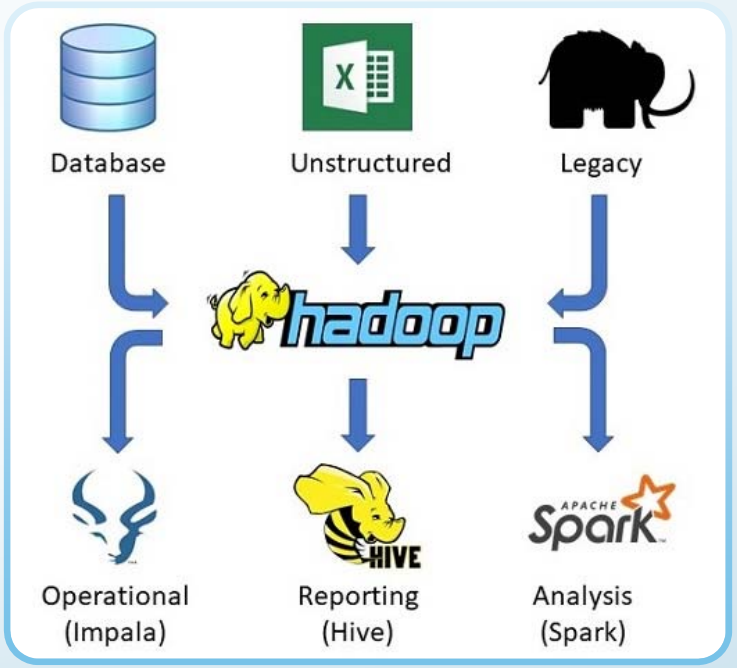
SYNCHRONIC ANALYTICS

Hadoop Use Cases

DATA CONSOLIDATION

Consolidating data through Hadoop makes your business more efficient and more robust at less cost.

- **Efficient**: Streamlined workflows with fewer bottlenecks
- **Robust**: Fewer "moving parts" and built-in redundancy
- **Unburdened** from legacy hardware and license costs



SYNCHRONIC ANALYTICS

Data Consolidation in Accounts Receivable

- A financial clearinghouse must route client data according to transaction type, but different transactions were processed by different systems.
- They built a Hadoop data lake to store transaction data and a Hive data warehouse containing a master routing table.
- Now transactions from several legacy systems are resolved in one place.

Hadoop Components Used
HDFS data lake
Hive data warehouse
Sqoop (data ingestion)

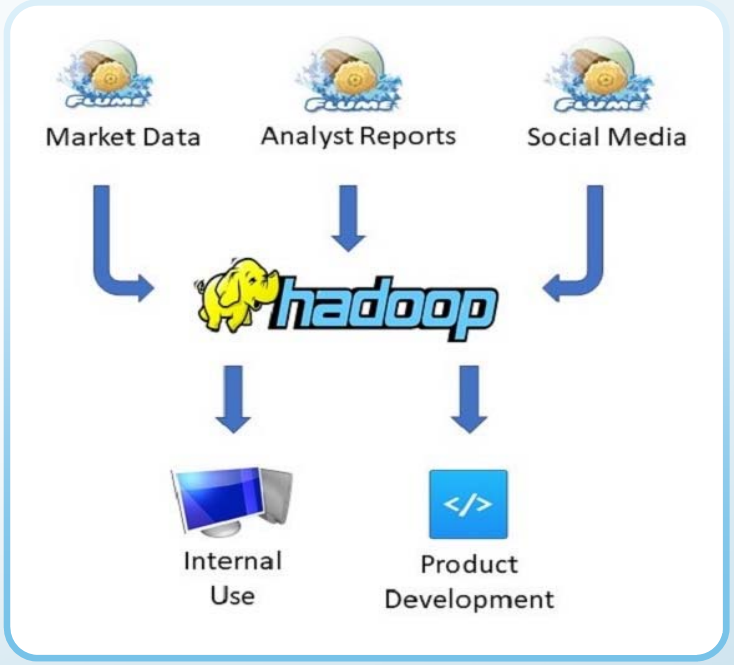
SYNCHRONIC ANALYTICS

Hadoop Use Cases

DATA CAPTURE

Hadoop lets you capture and analyze very large, diverse data sets, including real-time data.

- **Capture** TB of data from web pages, external databases, and data streams
- **Update** internal data stores in real time
- **Analyze** data with machine learning for real-time response



SYNCHRONIC ANALYTICS

Data Capture in Home Improvement

- A home-improvement retailer wanted their customers to find the right hardware.
- They used Hadoop to capture customer position and a smartphone app to direct the customer to the right product.
- In trial stores, customers were more likely to select the right hardware, leading to fewer returns, increased customer satisfaction, and lower costs overall.

Hadoop Components Used
HDFS data lake
Flume stream capture (position)
Hbase stream storage
Spark data analysis (product selection)

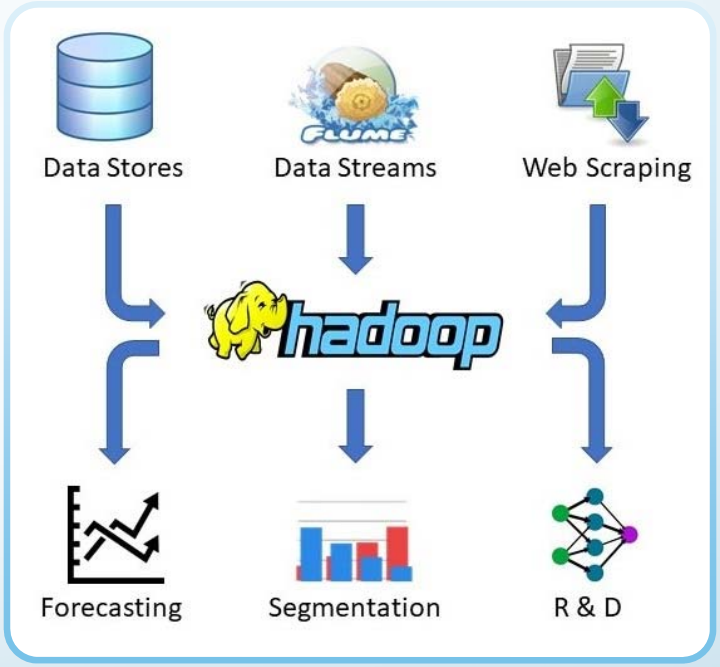
SYNCHRONIC ANALYTICS

Hadoop Use Cases

DATA ANALYSIS

Hadoop lets you analyze all your data to discover new insights and better serve your clients.

- **Analyze** your data as a whole, rather than in separate silos.
- **Discover** hidden trends and opportunities.
- **Create** new products from multiple sources



SYNCHRONIC ANALYTICS

Data Analysis in the Courtroom

- A law firm needed to predict whether clients should settle or proceed to court.
- They used Hadoop and machine learning to suggest the best strategy for the client
- In early testing, clients found that "standing firm" was often more profitable.

Hadoop Components Used
HDFS data dake
Hive data warehouse (case data)
Flume stream capture (other predictors)
Spark data analysis (recommendation)

SYNCHRONIC ANALYTICS

Hadoop Components, Briefly

- HDFS: the file system.
  - Stores all data
  - Enables distributed processing of very large data sets.
- Hive: the SQL query engine.
  - Query data in HDFS using a SQL-like syntax.
  - Overlay data with tables for reporting.
  - Hadoop's data warehouse
- Impala: the real-time query engine.
  - Queries in real-time
  - Works with existing Hive tables
  - Extremely fast, but a resource hog

SYNCHRONIC ANALYTICS

Hadoop Components, Briefly

- Sqoop: Imports databases
  - Easily import SQL databases.
  - Performs incremental updates.
  - Fast and scalable; require tuning.
- Flume: Captures streams
  - Capture data feeds, mouse clicks, web logs, etc.
  - Data typically stored in HBase
- Hbase: NoSQL database
  - NoSQL key-value database for streams and massive aggregation
  - Full-audit history and rule-based access
  - Requires careful integration; very powerful if done correctly.

SYNCHRONIC ANALYTICS

Apache Spark

Spark provides customizable machine learning at scale to existing Hadoop installations.

Benefits

- Standard and cutting-edge algorithms
- Supervised and unsupervised learning
- Very fast, even with huge data sets

Challenges

- Replaces certain Hadoop components; planning a must!
- Learning curve
  - Programmed in Scala, R, or Python
  - Users must understand algorithms; nothing predefined

SYNCHRONIC ANALYTICS

Implementation Options

Apache packages:

- Open source; no up-front cost
- Install semi-standardized, but still challenging
- Components (e.g. Hive, Sqoop) installed separately; greater flexibility, but can become dependency nightmare
- Limited web management; most interaction by Terminal
- Admins must know Linux; developers must use it.
- Gives you core toolset and little else.

SYNCHRONIC ANALYTICS

Implementation Options

Third-party vendor:

- Several vendors offer Hadoop suites:
  - Cloudera
  - Hortonworks
  - MapR
- Provides component integration; resolves (most) dependencies
- Includes web management; learning curve much faster
- Drawbacks are cost, lock-in, and little control over updates/end-of-life.

SYNCHRONIC ANALYTICS

Questions?

SYNCHRONIC ANALYTICS