Table of Contents

Big Data & hadoop Ecosystem 1

*The Need for Hadoop… 1*

*Take the Processing to the Data 2*

*The Hadoop Ecosystem & Hortonworks 2*

*What is HDFS? 4*

*YARN* 5

Installation 6

Configuration 7

Trobleshooting 7

High Availability 7

Security 7

# Big Data & hadoop Ecosystem

Hadoop is a framework used to handle big data for storing,processing and analyzing data that was previously ignored due to the limitations of traditional data management technologies.

## The Need for Hadoop…

* Hadoop fills several important needs in your data storage and processing infrastructure:
* Store and use all types of data: Allows semi-structured, unstructured and structured data to be processed in a way to create new insights of significant business value.
* Process all the data: Instead of looking at samples of data or small sections of data, organizations can look at large volumes of data to get new perspective and make business decisions with higher degree of accuracy.
* Scalability: Reducing latency in business is critical for success. The massive scalability of Big Data systems allow organizations to process massive amounts of data in a fraction of the time required for traditional systems.
* Commodity hardware: Self-healing, extremely scalable, highly available environment with cost-effective commodity hardware.

## Take the Processing to the Data

* This is what makes Hadoop so unique. A developer or analyst can write applications or use existing tools, such as Pig or Hive, to read data and analyze it in place, without it moving around your network.
* The tooling and APIs make it possible for all types of users to process data in Hadoop. For example Hive, one of the main frameworks in the Ecosystem, allows users to express their data analysis in natural SQL. Thus, the entry into Hadoop is very minimal.

## The Hadoop Ecosystem & Hortonworks

* The Hortonworks Data Platform, or HDP for short, is the only 100% open source data management platform for Apache Hadoop, and is the most stable and reliable Apache Hadoop distribution. It delivers the cost-effectiveness of Hadoop and the advanced services required for enterprise deployments. In this class, we will discuss all of them, but with some more depth for Pig, Hive, Sqoop, Oozie, MapReduce, as these are key frameworks.
* HDP can be installed and is supported on RedHat/CentOS Linux, Windows Server, Suse, and Oracle Linux.
* The key features of HDP include:
* High Availability: HA is now achieveable in HDP 2.1 without the use of an outside technology.
* Open Source Cluster Management: HDP includes Apache Ambari, the only open source operations tools that allows you to provision, manage and monitor a Hadoop cluster of any size.
* Metadata Services & HCatalog: HCatalog provides metadata services and a REST interface that provides an additional SQLlike interface to Hadoop.
* Data Integration Services: including Sqoop, Flume and WebHDFS.
* ODBC Done Right: Hive has a free high-performance ODBC driver that includes a SQL engine so you can interact with nearly every BI tool, including all SQL-92 interfaces.

## What is HDFS?

Data in Hadoop is stored on a filesystem referred to as HDFS - the Hadoop Distributed File System. With HDFS, data is broken down into chunks and distributed across a cluster of machines.

HDFS has the following characteristics:

* Primary storage system for Hadoop, it stores large files as small blocks Designed to be deployed on low-cost hardware
* Designed to scale easily and effectively (adding more nodes increases both storage space and computing throughput)
* Reliability - data is replicated so that disk failover is not only acceptable, but expected and handled seamlessly



**NOTE**: HDFS is the storage mechanism for Hadoop

## The Components of HDFS:

## NameNode

* The “master”node of HDFS
* Determines and maintains how the chunks of data are distributed across the DataNodes

## DataNode

* Stores the chunks of data, and is responsible for
* replicating the chunks across other DataNodes

## YARN

When many folks think about Hadoop, they are really thinking about two related technologies. These two technologies are the Hadoop Distributed File System (HDFS), which houses your data, and MapReduce, which allows you to actually do things with your data. While MapReduce is great for certain categories of tasks, it falls short with others. This led to fracturing in the ecosystem and a vari‐ ety of tools that live outside of your Hadoop cluster but attempt to communicate with HDFS.

In May 2012, version 2.0 of Hadoop was released, and with it came an exciting change to the way you can interact with your data. This change came with the introduction of YARN, which stands for Yet Another Resource Negotiator.

YARN exists in the space between your data and where MapReduce now lives, and it allows for many other tools that used to live outside your Hadoop system, such as Spark and Giraph, to now exist natively within a Hadoop cluster. It’s important to understand that Yarn does not replace MapReduce; in fact, Yarn doesn’t do anything at all on its own. What Yarn does do is provide a convenient, uni‐ form way for a variety of tools such as MapReduce, HBase, or any custom utilities you might build to run on your Hadoop cluster

# Installation

* Configure a local HDP repository
* Install ambari-server and ambari-agent
* Install HDP using the Ambari install wizard
* Add a new node to an existing cluster
* Decommission a node
* Add an HDP service to a cluster using Ambari

# Configuration

* Define and deploy a rack topology script
* Change the configuration of a service using Ambari
* Configure the Capacity Scheduler
* Create a home directory for a user and configure permissions
* Configure the include and exclude DataNode files

# Trobleshooting

* Restart an HDP service
* View an application’s log file
* Configure and manage alerts

# High Availability

* Configure NameNode HA
* Configure ResourceManager HA
* Create a snapshot of an HDFS directory
* Recover a snapshot

# Security

* Install and configure Knox
* Install and configure Ranger
* Configure HDFS ACLs