Curso de extensão em Data Science

GERÊNCIA DE INFRAESTRUTURA PARA BIG DATA

Prof. Tiago Ferreto – tiago.ferreto@pucrs.br

ESCOLA POLITÉCNICA

HADOOP

Big Data and Hadoop History

- Early 2000s – emergence of storage and processing methodologies ("Big Data") from search engine providers – principally Google and Yahoo!
- Search engine providers
  - First group to face with Internet scale problems – how to process and store indexes of all of the documents in the Internet
- Timeline
  - 2003 – Google releases "The Google File System" whitepaper – scalable and fault tolerant file system
  - 2004 – Google releases "MapReduce: Simplified Data Processing on Large Clusters" – large scale processing
  - At the same time – Doug Cutting starts working on a web indexing project called Nutch
    - Inspired by Google papers, he decides to incorporate the storage and processing principles, which are later moved to a new project called Apache Hadoop
  - 2006 – Hadoop is born as an open source project under the Apache Software Foundation

Big Data and Hadoop History

- At the same time as the Hadoop project was created, other technology innovations were happening (data deluge)
  - Rapid expansion of ecommerce
  - Birth and rapid growth of the mobile Internet
  - Blogs and user-drive web content
  - Social media
- Besides Hadoop, it also led to the emergence of other projects, such as Spark, Messaging Systems, NoSQL, etc

What is Hadoop?

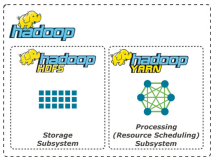
- Hadoop is a **data storage and processing platform**
- Central concept → **data locality**
  - Data locality refers to the processing of data where it resides (computation is sent to the data, rather than computation requesting data from its location - e.g. DBMS)
- Big data makes it really difficult to move data around
- Hadoop enables processing large datasets locally on the nodes of a cluster
  - Uses a **shared nothing approach**
  - Each node independently processes a subset of the entire dataset without needing to communicate with one another

What is Hadoop?

- Hadoop is schemaless with respect to its write operations
  - **It is a *schema-on-read* system**
  - Can store and process a wide range of data, such as:
    - Unstructured text documents
    - Semi-structured JSON or XML documents
    - Well structured extracts from relational database systems
- In *schema-on-write* systems, such as relational databases, data is strongly typed and a schema is predefined and enforced in all operations
  - NoSQL databases are *schema-on-read* systems
- Pros: doesn't need indexes, statistics or optimization constructs
- Cons: datasets with larger size and replicated data
- Hadoop uses a **divide and conquer approach** to tackle problems applying data locality and shared nothing concepts

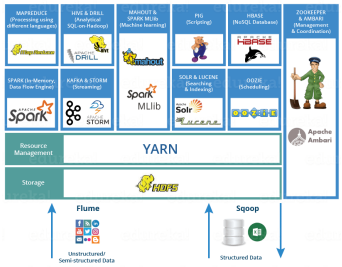
Hadoop –Components

- Hadoop Common – libraries and utilities needed by other modules
- HDFS (Hadoop Distributed File System) – distributed file-system for data storage
- Hadoop YARN (Yet Another Resource Negotiator) – platform for managing computing resources and scheduling applications
- Hadoop MapReduce – implementation of the MapReduce programming model



Hadoop Ecosystem Components

- Hadoop Ecosystem includes any other projects that interact or integrated with Hadoop in some way (Data ingestion, Data analysis, NoSQL Databases, Security, Management, Search, Machine Learning, Workflow coordination, etc)



Source: https://www.edureka.co/blog/hadoop-ecosystem

Commercial Hadoop Landscape

- 2008 – Cloudera
  - CDH (Cloudera Distribution of Hadoop)
- 2009 – MapR
  - Includes MapRFS – custom storage solution, adapted from HDFS
- 2011 – HortonWorks
  - HDP (Hortonworks Data Platform)
- Cloudera, Hortonworks, and MapR are considered "pure play" Hadoop vendors
  - Business models are founded upon Hadoop
  - Other vendors with proprietary distributions – IBM, Pivotal, Teradata
    - Hadoop is not their core business
- 2014 – ODPi (Open Data Platform Initiative) founded by Hortonworks – odpi.org
  - Goal: provide a consistent, commercial ready set of Hadoop core and selected ecosystem components
  - Includes other vendors: IBM, Teradata, EMC², SAS, Pivotal, Splunk, and others

Hadoop releases

- Hadoop 1 (2012)
  - First release - HDFS and MapReduce
- **Hadoop 2 (2013)**
  - Introduces YARN - optimizes resource allocation for various applications
- Hadoop 3 (2017)
  - Support multiple name nodes on HDFS (solves SPOF problem)
  - HDFS uses erasure coding – reduces storage overhead
  - Utilization of containers (Docker)
  - Supports the utilization of GPUs for processing (Deep Learning)

Typical Hadoop Use Cases

- **Data warehouse offload**
  - Utilization of Hadoop for long-term storage and ETL routines
  - Leverages the lower cost storage and distributing processing capabilities of Hadoop (in comparison to specialized Data Warehouse platforms)
  - Implementation of a Data Lake - storage of unstructured, unprocessed data prior to being staged of integrated into a data warehouse or exposed in a data mart

Typical Hadoop Use Cases

- **Event and complex event processing**
  - Associated with IoT (Internet of Things) - thousands of end points, including temperature sensors, RFID, NFC scanners, CCTV cameras, signalling systems, etc.
  - Involves the ingestion and processing of streaming data sources, such as sensor data, message data, or log data
  - Hadoop provides low-cost storage to accumulate millions of messages and a computational platform to mine volumes of messages to identify patterns or peculiarities.
  - Typically involves other Hadoop ecosystem platforms, such as Storm, Flume, Spark Streaming, Kafka, or others

Typical Hadoop Use Cases

- **Advanced Analytics (Data Science)**
  - Process of identifying patterns in data and building mathematical models to represent these patterns, which can be used to predict outcomes from subsequent data.
  - Advanced analytics encompasses the specific disciplines of data mining, predictive modelling, and machine learning, which are typically search-based, iterative, trial-and-error–based problem solving approaches
  - Hadoop enables all these approaches at scale

Hadoop Cluster Architecture Overview

- HDFS Cluster – Storage
- YARN Cluster – Processing

- Both clusters can be used independently. However, a Hadoop Cluster is when both clusters (HDFS and YARN) are configured on the same nodes.

HDFS Basic Concepts

- Set of Java processes to manage and orchestrate Hadoop's distributed filesystem

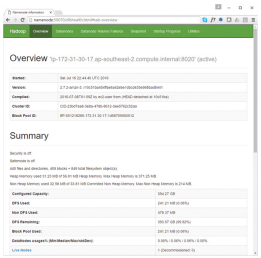
- Master-slave cluster architecture
  - Model of communication whereby one process has control over one or more other processes
  - In HDFS, master and slave processes are predesignated (static roles)
  - HDFS processes run as daemons/services on cluster nodes and are classified as master or slaves

HDFS Basic Concepts

- Files, Blocks, and Metadata
  - Files are comprised of blocks in HDFS
    - Block size is configurable (default: 128 MB)
  - Blocks are distributed and replicated across one or more nodes of the cluster
    - Goals: fault tolerance and enable parallel processing using local data
  - Filesystem metadata contains information about virtual directories, files and physical blocks that comprise the files

- NameNode
  - HDFS master node process → manages the filesystem metadata
  - Metadata is kept in memory for faster access
  - Uses journaling functions to assure durability and consistency of the filesystem's metadata
  - NameNode serves a web interface on port 50070

NameNode Web Interface

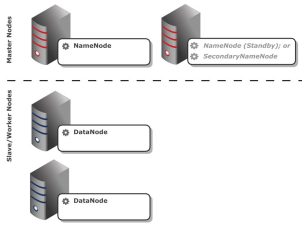


HDFS Basic Concepts

- Secondary NameNode and Standby NameNode
  - Optional processes
  - Expedite filesystem metadata recovery (Secondary NameNode)
  - Provide failover process (Standby NameNode)

- DataNode process
  - HDFS slave node daemon → run on one or more nodes of the HDFS cluster
  - Responsible for managing block storage and access for read/write data
  - Also responsible for block replication

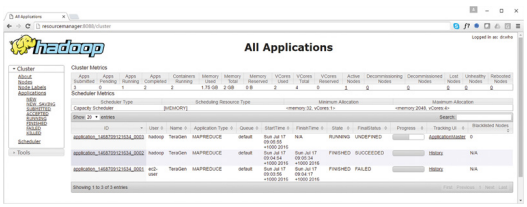
HDFS Cluster Overview



YARN Basic Concepts

- Uses a master-slave model like HDFS
  - ResourceManager and NodeManager
- ResourceManager
  - Master of the YARN Cluster
  - Responsible for scheduling applications and granting cluster compute resources
    - Tracks applications' status
  - Resources are represented as units called "containers" (combinations of CPU cores and memory)
    - Tracks available capacity in the cluster - allocates and releases containers to applications
  - For each application submitted, ResourceManager allocates the first container on an available NodeManager as a delegate process for the application called "ApplicationMaster"
  - ApplicationMaster negotiates further containers required for the application
  - ResourceManager serves a web interface on port 8088

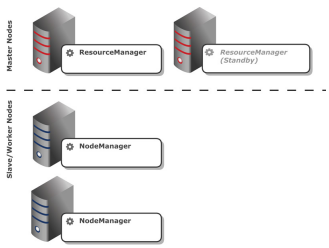
ResourceManager Web Interface



YARN Basic Concepts

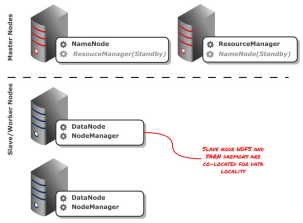
- NodeManager
  - Slave node daemon
  - Manages containers on the host
- ApplicationMaster
  - First container allocated by the ResourceManager to run on a NodeManager for an application
  - Determines resources required to run the application (based on the amount of data to be processed)
  - Orchestrates the allocation of resources to execute application stages (eg. Map and reduce stages)
    - ApplicationMaster requests resources from the ResourceManager on behalf of the application
    - ResourceManager grants resources on the same or other NodeManagers to the ApplicationMaster
  - Monitors progress of tasks, stages (groups of tasks executed in parallel), and dependencies

YARN Cluster Overview



Hadoop Cluster Architecture

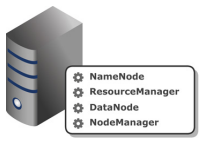
- Combines HDFS and YARN Clusters



Hadoop Cluster Deployment Modes

- Hadoop supports three deployment modes
  - LocalJobRunner (Standalone Operation)
    - Runs an application locally in a single JVM
    - Enables integrating the application with an IDE (e.g. Eclipse) for performing unit tests, debugging, and tracing
    - Uses the local filesystem instead of HDFS
  - Pseudo-Distributed
    - All Hadoop daemons are executed on separate JVMs on a single host
    - Simulates how a fully cluster would function
    - Useful for testing to simulate the interaction between components in a real cluster using a single machine
  - Fully Distributed Cluster
    - Executes master and slave daemons on distinct machines
    - Typical deployment mode for production systems

Pseudo-Distributed Mode



Hadoop Deployment

- Hadoop was originally developed for Linux
  - Windows-compatible distribution already exists, but may present compatibility issues with other Hadoop ecosystem projects
- Recommendations
  - Do not use LVM (Logical Volume Manager). It may restrict performance (especially on slave nodes)

Requirements (medium to large scale production Hadoop clusters)

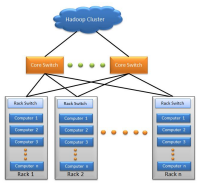
- **Master Nodes**
  - 16 or more CPU cores (preferably 32)
  - 128GB or more RAM (preferably 256GB)
  - RAID Hard Drive Configuration (preferably with hot-swappable drives) - for fault tolerance
  - Redundant power supplies - for fault tolerance
  - Bonded Gigabit Ethernet or 10Gigabit Ethernet

Requirements (medium to large scale production Hadoop clusters)

- **Slave Nodes**
  - 16-32 CPU cores
  - 64-512 GB of RAM
  - 12-24 1-4 TB hard disks in a JBOD Configuration
    - JBOD (Just a Bunch of Disks) – directly attached storage not in RAID configuration; each disk operates independently
    - RAID is not recommended for block storage on slave nodes!
      - Access speed is limited by the slowest disk in the array. JBOD has been proven to outperform RAID 0 for block storage by 30% to 50% in benchmarks conducted at Yahoo!.
  - Slave nodes support failures, therefore they do not require the same degree of fault tolerance that master nodes do
  - Storage capacity in slave nodes should not be fully allocated!
    - Failure of a node may impact the network due to replication of blocks

Networking Considerations

- Fully-distributed implementations consume significant amount of network resources
  - Control messages, status updates and heartbeats, block reports, data shuffling, and block replication
- On-premises deployment
  - Private subnets with dedicated switches
  - Redundant core and TOR (Top of Rack) switches
    - Utilization of STP (Spanning Tree Protocol) to avoid loops
- Hostname resolution between nodes
  - Forward and reverse DNS configuration or hosts file
- Time synchronization (NTP – Network Time Protocol)
  - Used by some components (e.g. Kerberos)



Software requirements

- Hadoop is almost entirely written in Java
- Requirements
  - Java Runtime Environment (JRE) 1.7 or above
  - Java Development Kit (JDK) 1.7 or above - for compiling MapReduce applications
- OpenJDK and Oracle versions supported

- Some versions may present problems!
- Site with tested JDKs
  - https://wiki.apache.org/hadoop/HadoopJavaVersions

- Other ecosystem projects may have other specific requirements
  - For instance, Apache Spark requires Scala and Python

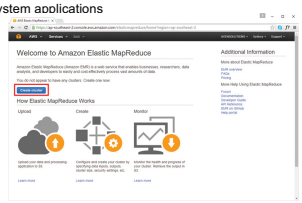
Installing Hadoop On-premises

- Non-Commercial Hadoop
  - Install from Apache builds – http://hadoop.apache.org

- Commercial Distributions
  - Cloudera (http://www.cloudera.com/)
    - Provides a Quickstart VM - http://www.cloudera.com/downloads.html
    - Cloudera Manager (proprietary software) – provides a management console and framework to deploy, manage and monitor Hadoop clusters
  - Hortonworks (http://hortonworks.com/)
    - Hortonworks Sandbox – VM with Hadoop pseudo-distributed mode - http://hortonworks.com/products/sandbox/
    - Hortonworks Data Platform (distribution with complete Hadoop stack and selected ecosystem components)
      - Uses Apache Ambari for deployment and management
  - MapR (https://www.mapr.com/)
    - MapR Sandbox – VM with Hadoop pseudo-distributed mode - https://www.mapr.com/products/mapr-sandbox-hadoop/download
    - MapR Control System (MCS) – central console to configure, monitor and manage MapR clusters

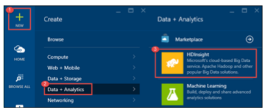
Installing Hadoop on the Cloud

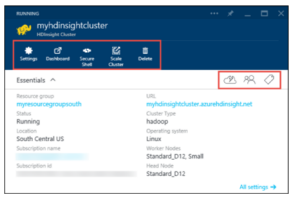
- Amazon Web Services
  - EMR (Elastic MapReduce) – Amazon's Hadoop-as-a-Service platform
  - Enables creating clusters with a specific number of nodes, node instance types (EC2), Hadoop distribution, and additional ecosystem applications
  - Uses S3 for reading/writing data
  - Provisioned on demand
  - https://aws.amazon.com/emr/



Installing Hadoop on the Cloud

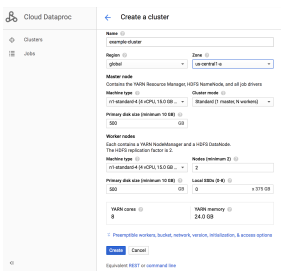
- Microsoft Azure HDInsight
  - https://azure.microsoft.com/en-us/services/hdinsight/





Installing Hadoop on the Cloud

- Google Cloud Dataproc
  - https://cloud.google.com/dataproc/



APACHE HADOOP - HANDS-ON