ISA 414 – Managing Big Data

Lecture 21 – Introduction to Hadoop

(Part I)

Dr. Arthur Carvalho

arthur.carvalho@miamioh.edu

MIAMI UNIVERSITY

Lecture Objectives

- Quick review of Homework 9

- Introduction to the Hadoop ecosystem

Lecture Instructions

- There are several new concepts today
  - Suggestion: actively take notes
  - Important keywords are highlighted in the slides

- Recall that several of such concepts will be in the final exam

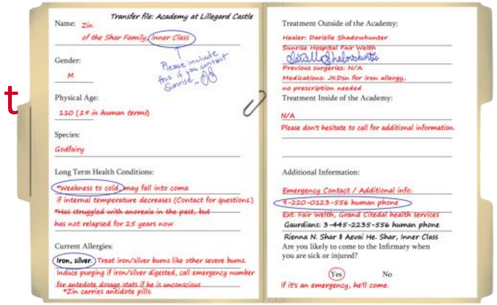
Agenda

- First part of the course: CRISP-DM
  - Managing big data projects
- Second part of the course: technologies/big-data enablers
  - Cloud computing and storage (previous 2 lectures)
    - IaaS and PaaS help with the required infrastructure
    - SaaS might help with the analysis
  - Hadoop framework (rest of the course)
- Before learning about Hadoop, we must first learn about two relevant concepts
  - Distributed storage
  - Distributed computations

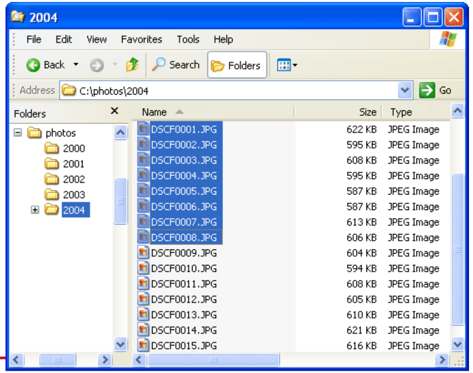
Distributed Storage

- Data are stored inside **files**
  - For our purposes, a file provides a way of storing and retrieving data/information
  - Different technologies
    - *E.g.*, paper records, computer files
  - Different formats
    - CSV, JSON, XML, XLSX, …

data about a patient



computer files

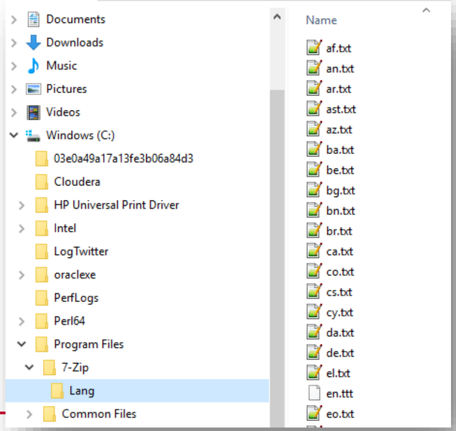


Distributed Storage

- How can we organize files?
  - Example: file cabinets organize paper-based files
    - File/folder organization/sorting is subjective

Distributed Storage

- How to organize computer files?
  - **(Computer) file system**
    - Managed by the operating system
      - Linux, Windows, Android,...
    - Often use file directories (like a file cabinet) and hierarchies (hierarchical trees)
      - Folders might contain subfolders
      - Files have exact addresses in the file system
        - **Paths** = branch of the hierarchical tree
        - *E.g.*, C:\carvalag\ISA414\Lecture19.pdf



Distributed Storage

- What about computer (digital) files? How can we store and organize them?
  - Series of bytes
  - Stored inside **storage devices** (*e.g.*, hard-drive disks, flash-based solid-state drives, …)
    - Non-volatile memory
    - More on this in future lectures

Distributed Storage

- Different computers have different storage capacities

- What happens when one runs out of storage space?
  - Remember that big data is often defined in terms of volume
- Should one just replace an old storage device with a new one?
  - Big hassle: transfer all the data to the new device
    - Think about an organization: potentially, hundreds of terabytes or even petabytes of data

Distributed Storage

- One way of tackling the previous problem is by storing data across multiple machines/storage devices
  - One can simply add a new machine or a storage device to a collection of machines when running out of storage
    - **Distributed storage**
    - No need to transfer data or replace old computers
  - How does one know where a certain file is?
    - Each machine has its own file system
    - The collection of machines has a **Distributed File System (DFS)**
      - Helps to store and index files across multiple machines

Distributed Storage

- Distributed storage tackles one issue related to big data
  - Namely the increasing need for storage space due to the volume aspect of big data
  - Summary of the main idea (we will elaborate on this later):
    - One can use the storage devices (*e.g.*, SSD devices) of many **commodity computers** to store data in a distributed fashion
      - "Many computers" = a **cluster** of computers
    - A *distributed file system* helps to organize and determine where each file is stored in a cluster (computer + file path)

Distributed Storage

- (Over) simplified example of a file system
  - Every file in the system is associated with a path

File	Path
Picture1.jpeg	C:/users/carvalag/pictures/Picture1.jpeg
data1.csv	C:/users/carvalag/data/data1.csv

- (Over) simplified example of a distributed file system
  - Every file in the system is associated with a path and storage device

File	Device ID	(Local) Path
noshow.csv	173.16.157.4	/user/carvalag/noshow.csv
data.csv	173.16.157.2	/user/smith2/data.csv

Distributed Computing

- Distributed storage does not tackle another problem associated with data volume
  - The increasing need for computational power

- Complex data-analytics tasks can often benefit from **parallel computation**

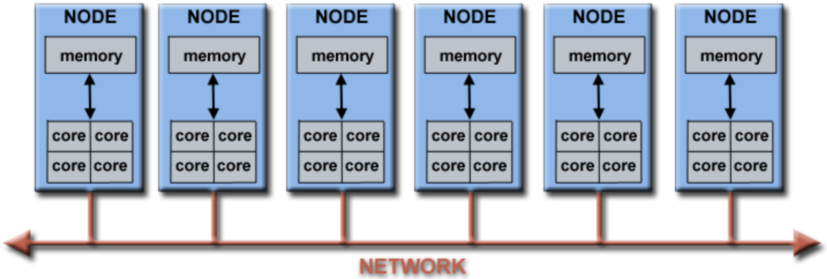
Distributed Computing

- Different ways of performing parallel computations
  1. Single **nodes** (computers)
     - Multi-core processors: single computing component (CPU) with two or more independent units (“cores”)
     - Relatively cheap and easy to program (threads)
       - For example, see the Python module threading

Distributed Computing

- Different ways of performing parallel computations
  2. Parallel computers (or "super computers")
     - Multiple CPUs:
       - Very large number of single computing nodes
       - Connected via some network (part of the machine)
       - Very expensive
         - From a few to hundreds of millions of dollars

Sunway TaihuLight
RAM: 1,310,000 GB
Storage: 20,000 TB
CPUs: 40,960
Cores: 10,649,600
Cost: $273 million



MIAMI UNIVERSITY

Distributed Computing

- Different ways of performing parallel computations
  3. **Commodity cluster**
     - Distributed computations across many relatively cheap (commodity) individual computers, each one having potentially many cores
     - Example: ISA 414 cluster (first request)
       - 5 Computer Nodes
         - 2 Nodes with 48 cores, 256 GB RAM each
         - 3 Nodes with 72 cores, 768 GB RAM each
         - 500 TB of shared storage capacity
       - Price tag: $116,351.64
         - Including service, racks, and other hardware

Distributed Computing

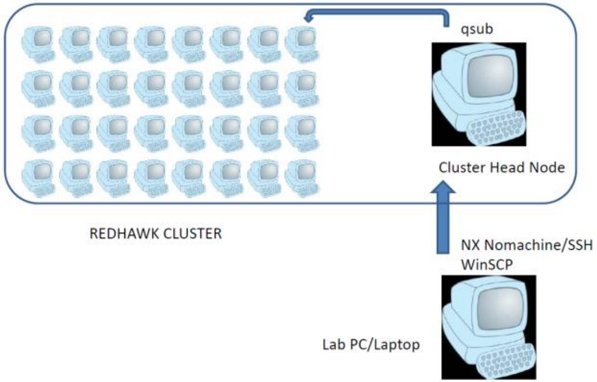
- Different ways of performing parallel computations
  3. **Commodity cluster**
     - Distributed computations across many relatively cheap (commodity) individual computers, each one having potentially many cores
     - Yahoo! Cluster (2010)
       - 3500 nodes. A typical cluster node has:
         - 2 quad core Xeon processors @ 2.5ghz
         - 4 hard disks (one terabyte each)
         - 16GB RAM

Source: paper "*The Hadoop Distributed File System*"

Distributed Computing

- Different ways of performing parallel computations

  3. **Commodity cluster**
     - RedHawks Cluster (https://www.miamioh.edu/research/research-computing-support/services/hpc-cluster/index.html)



Miami's current HPC cluster consists of:

- 2 login nodes – 24 cores, 384 GB of memory each. Machine names:
  - mualhplp01
  - mualhplp02
- 26 compute nodes – 24 cores, Intel Xeon Gold 6126 2.6 GHz processors, 96 GB of memory each. Machine names:
  - mualhpcp10.mpi-mualhpcp26.mpi
  - mualhpcp28.mpi-mualhpcp35.mpi
  - mualhpcp37.mpi
- 5 compute nodes - 24 cores, Intel Xeon Gold 6226 2.7 GHz processors, 96 GB memory each. Machine names:
  - mualhpcp42.mpi-mualhpcp45.mpi
  - mualhpcp47.mpi
- 2 large memory nodes – 24 cores, Intel Xeon Gold 6126 2.6 GHz processors , 1.5 TB of memory each. Machine names:
  - mualhpcp27.mpi
  - mualhpcp36.mpi
- 4 GPU nodes – 96 GB of RAM, 24 cores, Intel Xeon Gold 6126 2.6 GHz processors and each with 2 Nvidia Tesla V100-PCIE-16GB GPUs. Machine names:
  - mualhpcp38.mpi-mualhpcp41.mpi
- Shared storage system with approximately 30 TB of storage, expandable.

Distributed Computing

- Example of a top-of-the-line “commodity” computer
  - Cisco UCS C240 M4 Rack Server
    - 128 GB RAM
    - Dual Intel E5-2680v3 12-Core 2.50 GHz CPU
    - 2 disks, each on having 1TB HDD
    - NvidiaTesla K80 GPU
    - Price tag: $6,500

- Individual computers are stacked one on top of another in racks

MIAMI UNIVERSITY

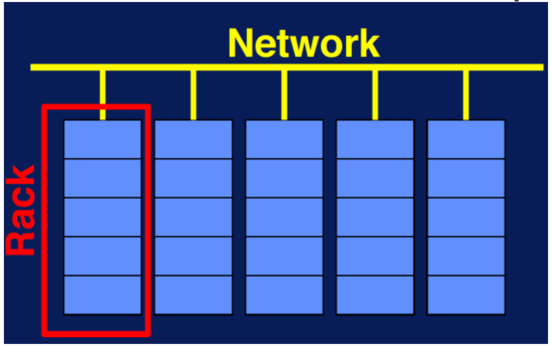
Distributed Computing

- Commodity cluster
  - Much cheaper than supercomputers
  - Less powerful
  - One can also have a cluster of old, very cheap computers

MIAMI UNIVERSITY

Distributed Computing

- Computer clusters
  - Many jobs and/or applications can run in parallel
    - Different machines
  - This tackle the problem with growing computational demand due to big data
    - Main idea: one can break down a demanding computation into pieces, which will be executed in parallel in different nodes



Distributed Storage and Computing

- Let's put things together now
  - A cluster of computers allows for:
    - Distributed storage
      - Requires a distributed file system
    - Distributed computing
      - Different computers can work on different (sub)tasks in parallel
  - Hence, a cluster of computers can solve some storage and computational challenges brought by big data
    - Big-data enabler
  - Commodity cluster means that the above can be done cheaply

Distributed Storage and Computing

- Let's put things together now
  - New computation paradigm: **move computation to data**
    - Different computers store different pieces of data
    - A task/job that needs access to a piece of data will be executed in the computer where the data are stored
  - Benefit: moving task/jobs require less bandwidth than moving data
    - *I.e.*, it does not mess up the network
  - We did the opposite in class
    - Move data (from a database server) to computation (Python code)
    - Assignment 3: we downloaded data from a MongoDB database

Distributed Storage and Computing

- How to manage a commodity cluster?
  - How to distribute data and computations across nodes?
  - Ideal storage operations:
    - Split volumes of data across nodes
    - Quickly retrieve distributed data
    - Enable the addition of more racks (nodes) without losing performance
    - Fault-tolerant
      - Replicate data partitions across nodes

Distributed Storage and Computing

- How to manage a commodity cluster?
  - How to distribute data and computations across nodes?
  - Ideal <u>computational</u> operations:
    - Scheduling many tasks at the same time running in different nodes
    - Automatic job restart when a node fail:
      - A rack (or individual computer) stop working
      - Network connection is lost

Distributed Storage and Computing

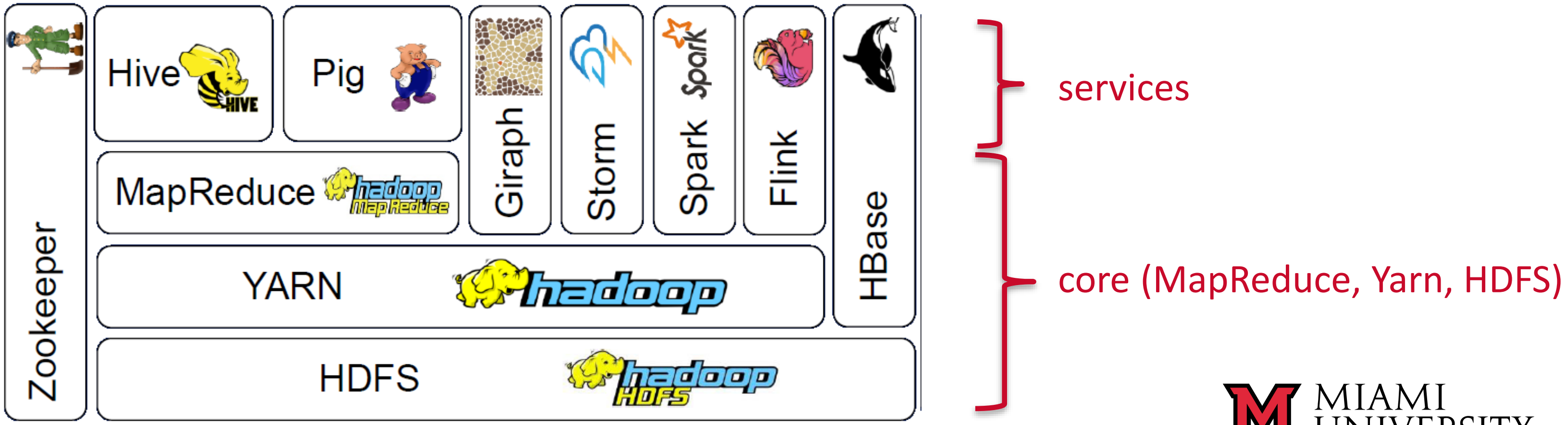
- **Hadoop**
  - Framework used for distributed storage and computing
    - *I.e.*, a tool that manages commodity clusters
    - Accomplishes all the ideal operations listed before
  - Distributed storage
    - Hadoop Distributed File System (HDFS)
  - Distributed computation
    - MapReduce
    - Spark
    - …

Distributed Storage and Computing

- Hadoop: timeline
  - 2004: MapReduce paper released by Google
    - Title: "*The Google File System*"
    - *Google File System* as distributed file system
    - MapReduce as distributed computing model
  - 2005: Yahoo! releases an open-source implementation of Google's framework called *Hadoop*
  - 2006: Apache continues to develop Hadoop
  - 2006 – present: many services built on top of core Hadoop
    - The *zoo: Hive*, *Pig*, *Giraph* … over 100+ services and counting

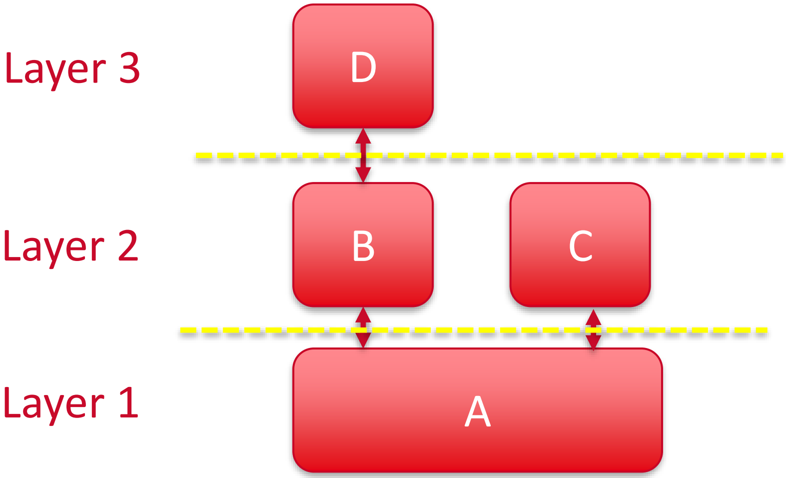
The Hadoop Ecosystem

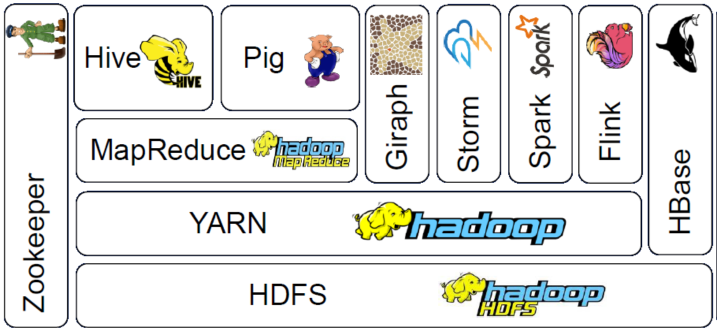
- Hadoop version 2
  - Simplified version
    - Many more services: Flume (log collector) Sqoop (data exchange), …



The Hadoop Ecosystem

- Hadoop version 2
  - Layer diagram (or stack)
    - A component uses the functionalities/capabilities of the layer below it

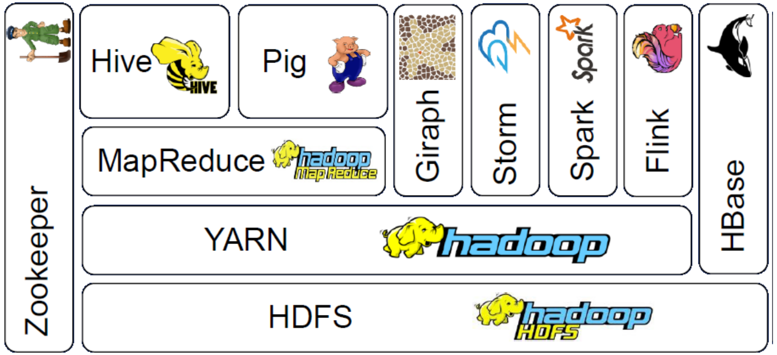




MIAMI UNIVERSITY

The Hadoop Ecosystem

- Hadoop version 2
  - Layer diagram (or stack)
    - A component uses the functionalities/capabilities of the layer below it

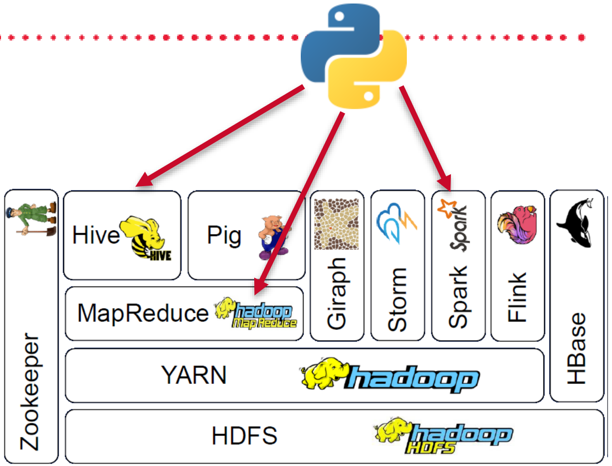


higher levels: interactivity

lower levels:
storage and task scheduling

The Hadoop Ecosystem

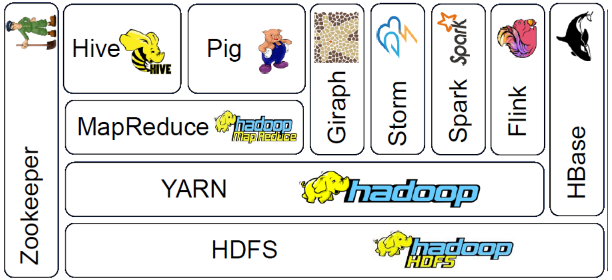
- Overview and agenda
  - HDFS (Lecture 22)
    - Hadoop Distributed File System
    - Scalable and reliable storage
  - Yarn (Lecture 22 - brief discussion)
    - Schedule jobs/task over HDFS storage
  - Spark (Lecture 23 and 24)
    - Built for real-time, in memory processing of data



MIAMI UNIVERSITY

The Hadoop Ecosystem

- Other services
  - Pig (created by Yahoo!)
    - Dataflow scripting
  - Giraph (created by Facebook)
    - Processing large graphs (social networks)
  - Storm/Flink (created by Twitter/Data Artisans)
    - Built for real-time, in memory processing of data
  - HBase (created by Facebook)
    - NoSQL database
    - Used by Facebook's messaging platform
  - Zookeeper (created by Yahoo!)
    - Manage services named after animals



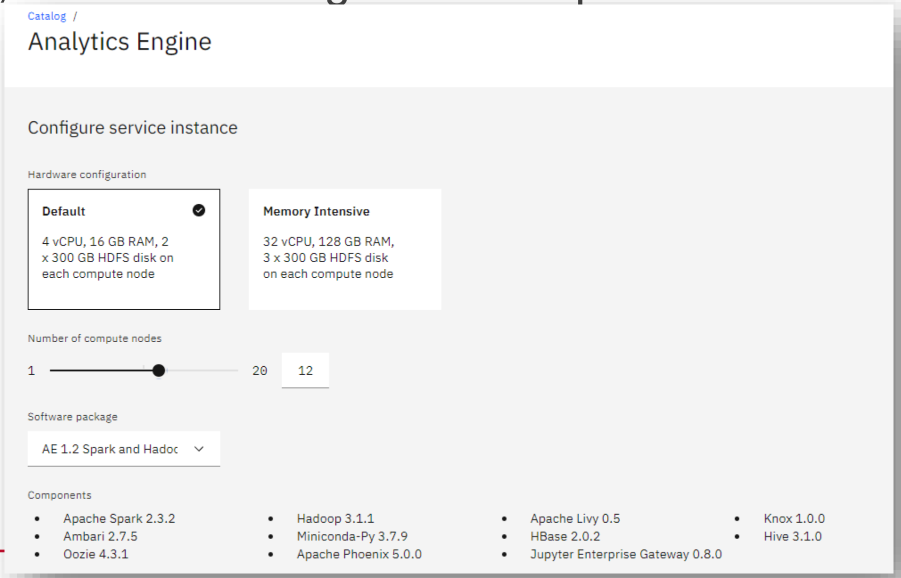
MIAMI UNIVERSITY

The Hadoop Ecosystem

- How to install Hadoop
  - All the previous tools are free and open-source (why?)
    - Large community for support
  - One can download and install each service/tool separately
    - Obviously, one must install lower-level services first (*e.g.*, HDFS, YARN) before installing higher-level services
    - Requires technical expertise (*e.g.*, advanced Linux/Unix skills)
  - Alternative #1: install a pre-built system (*i.e.*, stacks of these tools)
    - Cloudera, MAPR, Hortonworks
    - Offer commercial support for production environments

The Hadoop Ecosystem

- Alternative #2: cloud service (PaaS)
  - *E.g.*, you can have your own cluster of computers on IBM Cloud
    - Service name*: Analytics Engine*
    - Few clicks to add extra nodes (*i.e.*, increase storage and computational power) and Hadoop services

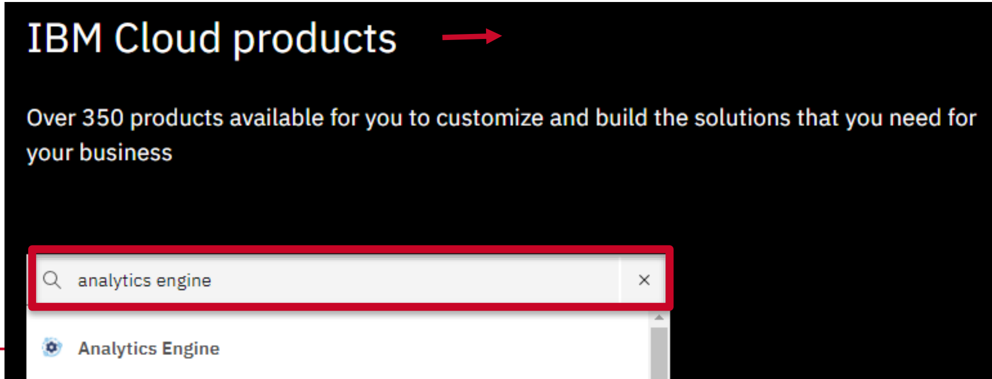


Demonstration

Creating a Hadoop Cluster on IBM Cloud

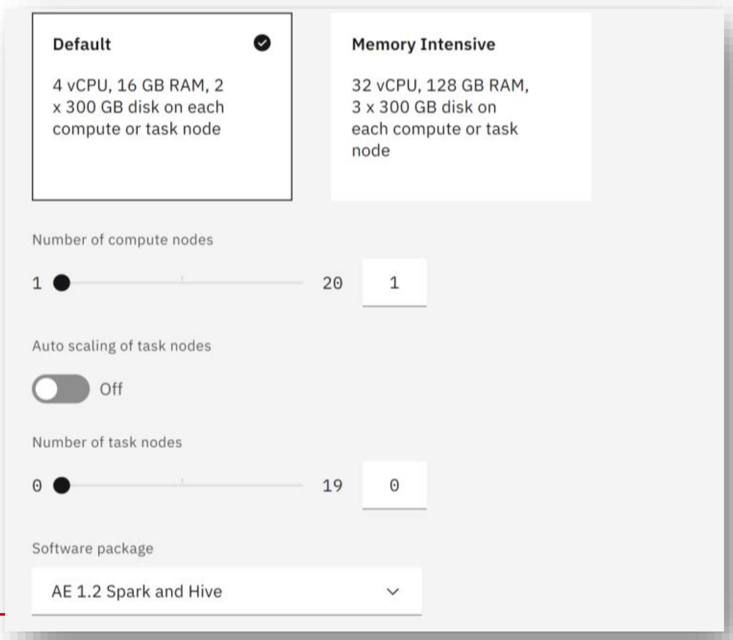
Hadoop on IBM Cloud

- After creating an account on IBM Cloud
  - Go to https://www.ibm.com/cloud
  - Log in
  - Select *Catalog -> Analytics*
  - Search for *“Analytics Engine”*



Hadoop on IBM Cloud

- Select the cloud's region
- Select the pricing plan, configure the cluster, and click on *Create*
  - It might take a few minutes for the cluster to be created



Hadoop on IBM Cloud

- Creating a cluster is incredibly easy
  - Difficult part: integrate a cluster with current business processes and in-house infrastructure

- We learn a few more details about Hadoop, HDFS, and Hadoop (PaaS) in our next class
  - Real-time demo with a cluster

Summary

- We learned how distributed storage and computing can tackle volume-related problems associated with big data
  - Hadoop = framework that manages distributed storage and computing
- Next lecture: we study the core of Hadoop
  - HDFS
  - Yarn (brief discussion)