Introduction to Big Data and Data Science (CSCE 5300 Section 001)*

Yunhe Feng

Assistant Professor, Department of Computer Science and Engineering

17th October, 2024

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

Quiz 3

- Closed-book in-person Quiz
- 5 Questions: 1 point for each question
- Quiz time: 2:35 pm - 3:00 pm

1. Hadoop Distributed Computing

2. Assignment

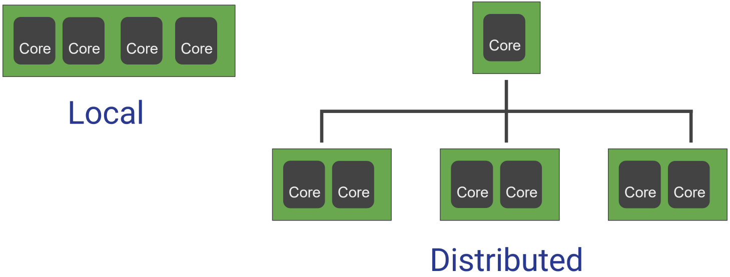
Hadoop Distributed Computing

- Local versus Distributed Systems
- Explanation of Hadoop, MapReduce, and Spark

Big Data: What if Data Exceeds RAM

- We have worked with data that can fit in to RAM of a local computer
- What can we do if we have a larger set of data?
  - Try using a SQL database to move storage onto hard drive instead of RAM
  - Or use a distributed computing environment, that distributes the data to multiple machines/Nodes

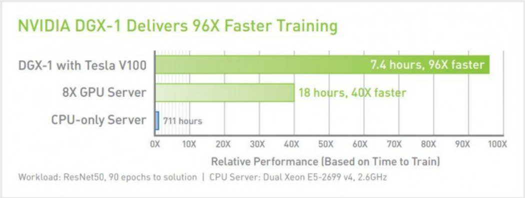
Local versus Distributed



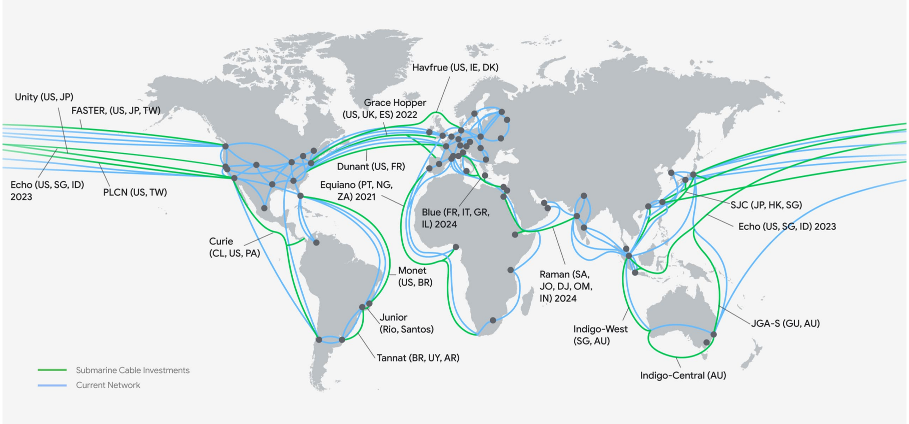
Local versus Distributed

- A local process will use the computational resources of a single machine.
- A distributed process has access to the computational resources across a number of machines connected through a network.

Local Computing: NVIDIA DGX-1 with V100 GPU



Distributed Computing: Goolge Cloud Sever Locations



Local or Distributed Processing?

- Which computing architecture is better?
  - A single node with several processor cores?
  - Or multiple nodes each with smaller set of cores?

Parallel Computing VS Distributed Computing

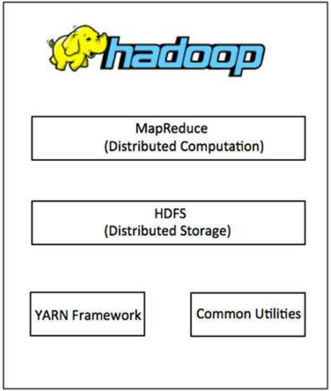
Aspect	Parallel Computing	Distributed Computing
Definition	Multiple processors within a single machine work simultaneously on the same task.	Multiple independent machines work together to solve a task over a network.
Hardware Architecture	Multi-core processors or shared memory systems within one machine.	A network of independent computers, each with its own memory and processor.
Communication	Uses shared memory for communication.	Communicates over a network (e.g., message passing).
Scalability	Limited by the number of processors/cores in one machine.	Scales horizontally by adding more machines.
Examples	GPU computing, scientific simulations, matrix operations.	Cloud computing (AWS, Hadoop), web applications, large-scale simulations.
Use Cases	High-performance tasks that benefit from shared memory, like image processing and simulations.	Large-scale distributed tasks, like big data processing and web services.

Apache Hadoop

- A a framework that allows for the **distributed processing** of **large data sets** across clusters of computers using simple programming models.
- Designed to **scale up** from single servers to thousands of machines, each offering local computation and storage
- Rather than rely on hardware to deliver high-availability, the library itself is designed to **detect and handle failures** at application the layer.

Hadoop Environment

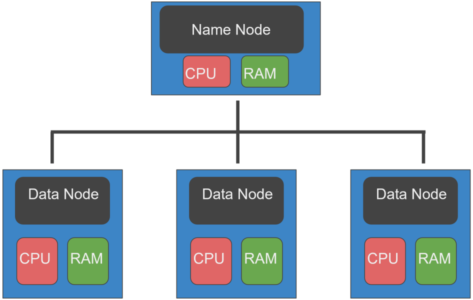
- Hadoop MapReduce: Processing/Computation layer
- Hadoop Distributed File System (HDFS): Storage layer
- Hadoop YARN: a framework for job scheduling and cluster resource management
- Hadoop Common: Java libraries and utilities required by other Hadoop modules



Hadoop Environment

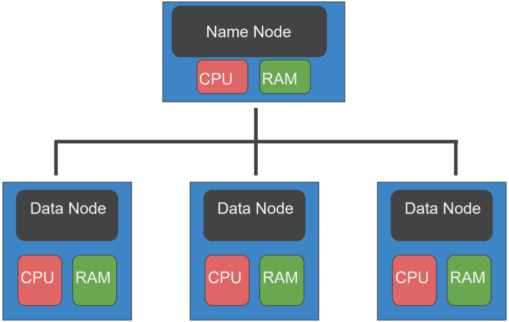
- Hadoop is a way to distribute very large files across multiple machines.
- It uses the Hadoop Distributed File System (HDFS)
- HDFS allows a user to work with large data sets
- HDFS also duplicates blocks of data across nodes for fault tolerance
- Hadoop computing on is based on MapReduce Algorithm and distributed data via client/server or master/slave model

Distributed Storage - HDFS



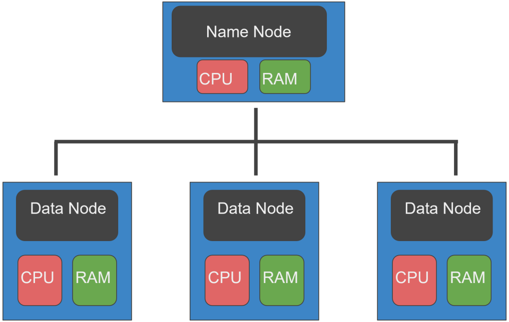
Block - Redundant Distributed Storage

- HDFS uses blocks of data, with a size of 128 MB by default
- Each of these blocks is replicated three times
- The blocks are distributed in a way to support fault tolerance



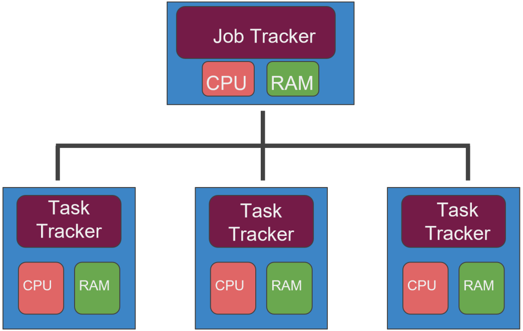
HDFS is Fault Tolerant

- Multiple copies of a block prevent loss of data due to a failure of a node.
- Smaller blocks provide more parallelization during data processing.

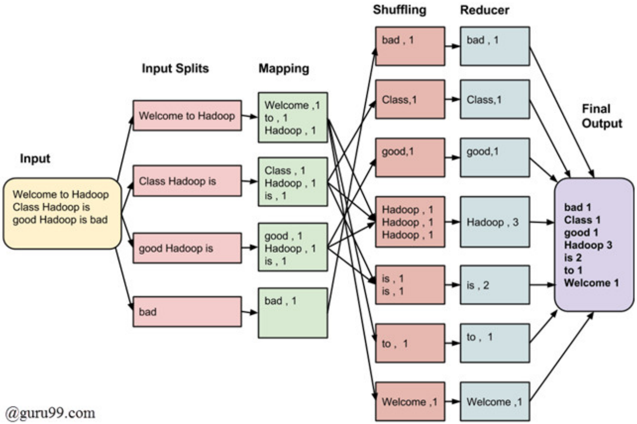


MapReduce Algorithm

- MapReduce is a way of splitting a computation task to a distributed set of files (such as HDFS)
- It consists of a Job Tracker and multiple Task Trackers
- Job Tracker sends code to run on the Task Trackers
- Task trackers allocate CPU and memory for the tasks and monitor the tasks on the worker nodes



MapReduce - Example



Covered So far: Hadoop Computing

- Hadoop uses HDFS to distribute large data sets and multiple copies for fault tolerance.
- Uses MapReduce and master/slave algorithm for comptuation on distributed data

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

Spark vs MapReduce

- You can think of Spark as a flexible alternative to MapReduce
- Spark can use data stored in a variety of formats
  - Cassandra
  - AWS S3
  - HDFS
  - And more

Spark vs MapReduce

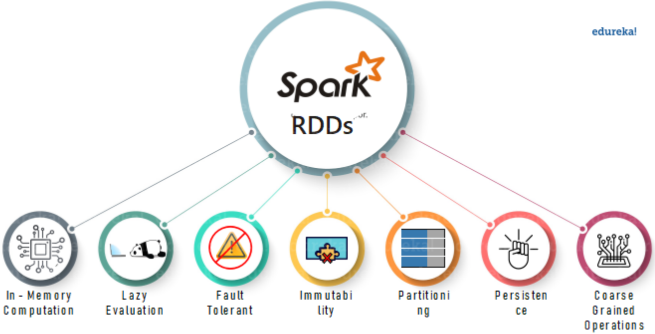
- MapReduce requires files to be stored in HDFS, Spark does not.
- Spark also can perform operations up to 100x faster than MapReduce
- So how does it achieve this speed?

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

Spark vs MapReduce

- MapReduce requires files to be stored in HDFS, Spark does not.
- Spark also can perform operations up to 100x faster than MapReduce
- So how does it achieve this speed?
  - In-Memory Computing
  - RDDs
  - DataFrames

Recap - Spark Resilient Distributed Dataset (RDD)



- RDD: a programming abstraction that represents an immutable collection of objects that can be split across a computing cluster
- Operations on RDDs: can also be split across the cluster and executed in a parallel batch process

Map, Filter and Reduce in Pyhton

See `https://book.pythontips.com/en/latest/map_filter.html`

1. Hadoop Distributed Computing

2. Assignment

Assignment-7 (4.0 pts.)

- Practice Map, Filter and Reduce functions in Pyhton (4 pts.)

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)