MapReduce

Frank Burkholder

(modified by Jon Courtney)

galvanize

- Distributed Computing
- Hadoop Ecosystem
- Hadoop MapReduce

Learning Objectives

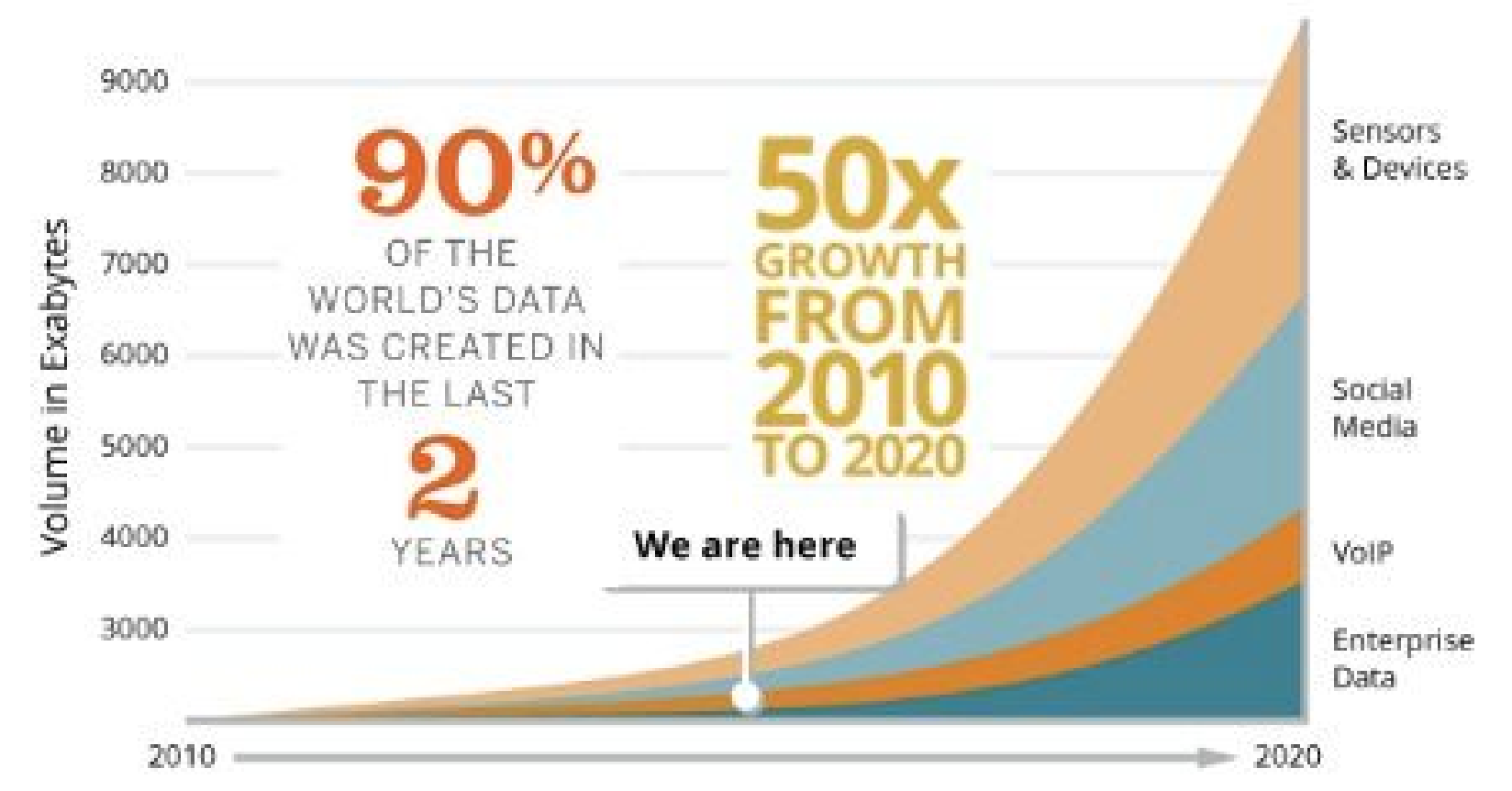
galvanize

- Describe what distributed computing is
- List 4 major components of the Hadoop Ecosystem
- Define **HDFS** and **MapReduce**.
- Track and explain computation through the MapReduce workflow
- Explain what a generator is
- Assignment: Use Python's `mrjob` package to perform a MapReduce job locally, then get on to an EC2 instance in Amazon Web Services (AWS) and use ElasticMapReduce to perform the same analysis in the cloud
    - Make an IAM account, make AWS access keys, specify an instance, generate ssh keys, ssh to connect to your instance, scp files from your laptop to your instance
- Getting comfortable on EC2 is one major objective of this course

Big Data

Growth of Data

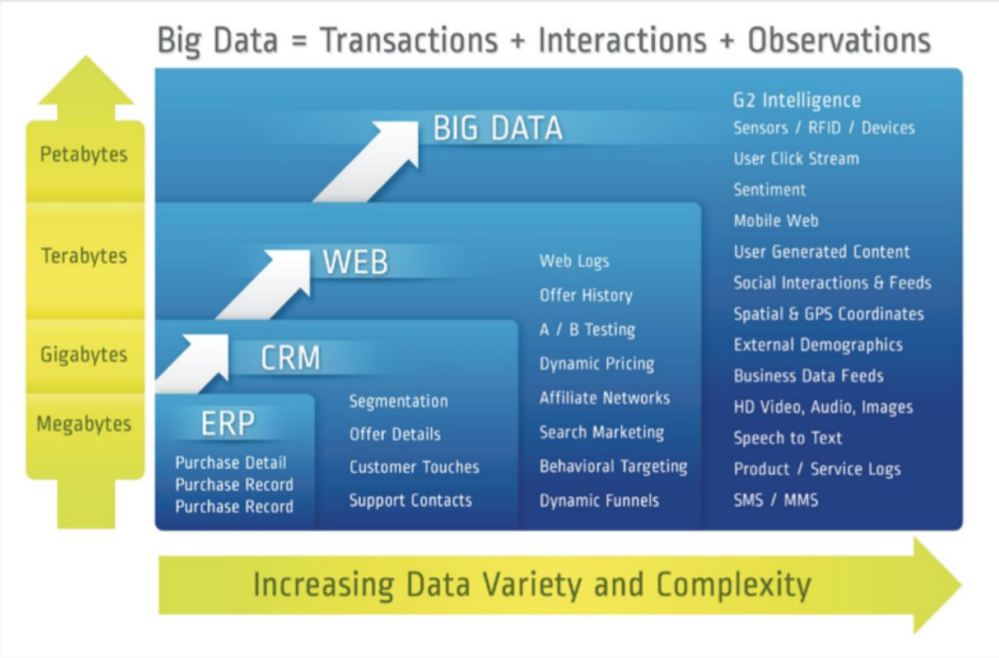
galvanize



https://www.promptcloud.com/blog/want-to-ensure-business-growth-via-big-data-augment-enterprise-data-with-web-data

Big Data

- Data so large that it cannot be stored on one machine.
- Can be
  - Structured: highly organized, searchable, fits into relational tables
  - Unstructured: no predefined format, multiple formats
- Often described as 3 Vs: (high volume, velocity, and variety)
- Two possible solutions to Big Data:
  - Make bigger computers (scale up)
  - Distribute data and computation onto multiple computers (scale out)



ERP: Enterprise resource planning
CRM: Customer relationship management

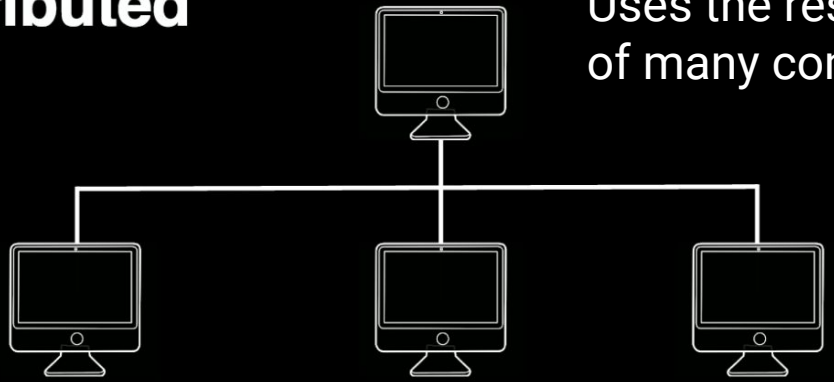
Local vs. Distributed Computing

galvanize

Local

Uses the resources of 1 computer

Distributed



Uses the resources of many computers

Distributed vs local computing, which is better?

galvanize

Let's say we have a problem that requires faster computation and more storage than any one commercial grade computer….

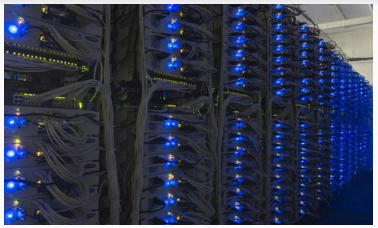
Why use a server rack
(look at all those cables!)
Distributed example: scale out

instead of

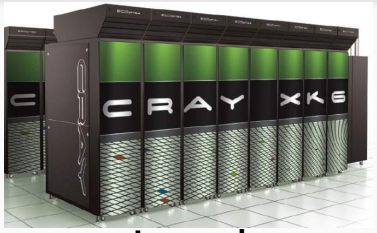
The Amazing Cray XK6
SUPERCOMPUTER
Local example: scale up

Where do Distributed vs. Local advantages lie?

galvanize



Distributed



Local

	Distributed	Local
Number of cores and amount of storage	Tie	Tie
Speed of communication between cores and storage		✓
Ability of cores to work on the same task		✓
Scalability (resource needs have changed)	✓	
Availability (if one thing fails is whole system down?)	✓	
Initial cost	✓	
Operation and maintenance	?	?

When to think about using distributed computing

galvanize

Size of data	Analysis tools	Data storage	Examples
< 10 GB	R/Python	Local: can fit in one machine's RAM	Thousands of sales figures
10 GB - 1 TB	R/Python with indexed files (key, record)	Local: fits on one machine's hard drive	Millions of web pages
> 1 TB	Hadoop, Spark, Distributed Databases	Distributed: Stored across multiple machines	Billions of web clicks, about 1 month of tweets

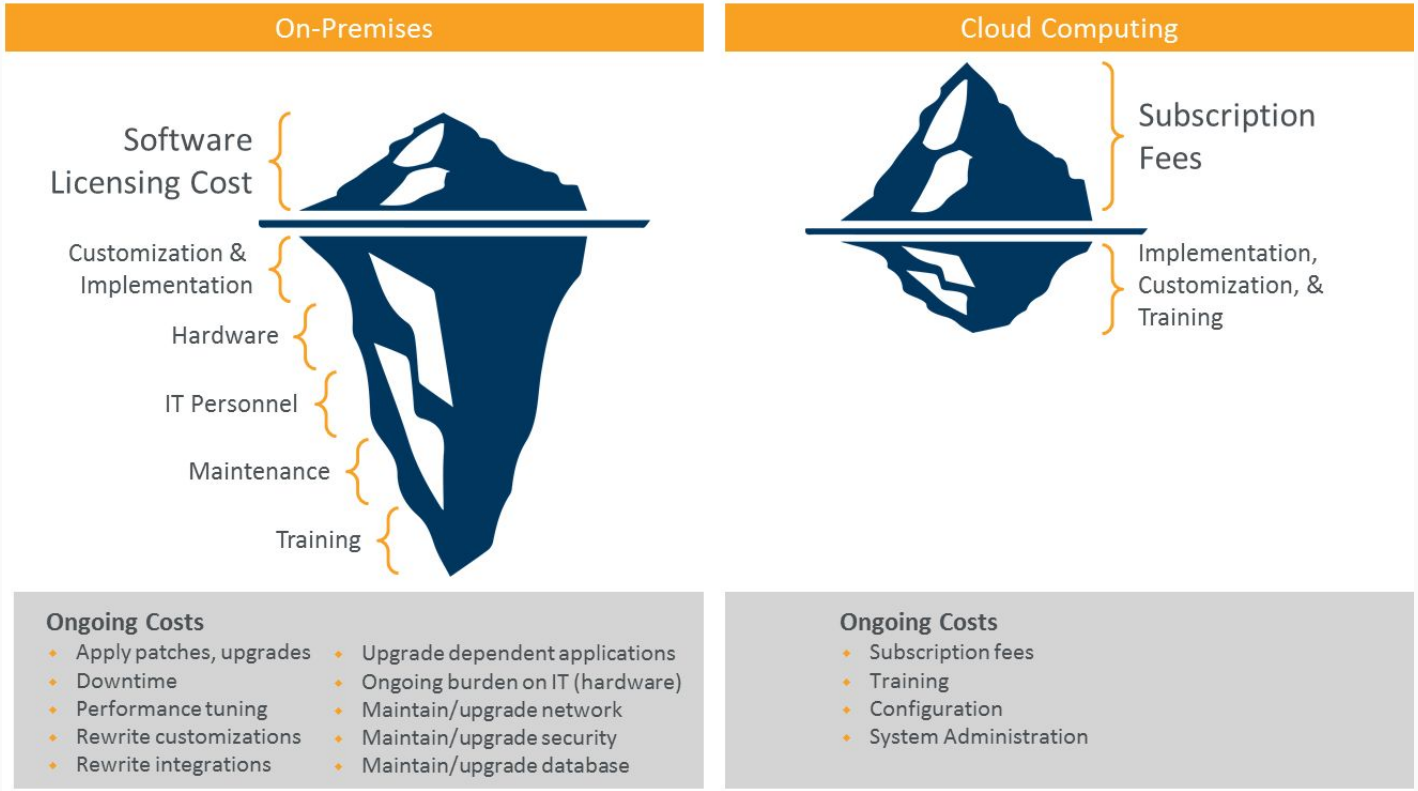
Of course, for the purposes of this class and project you are free to consider using Hadoop or Spark (later in curriculum) on much smaller datasets to get experience and potentially list it on your resume. *Just be advised: test it locally on a subset of the data first.*

Tangent: On-premise vs. the Cloud

galvanize

<u>On-premise</u>:
Software and/or hardware that are installed on the premises of the company that uses them.

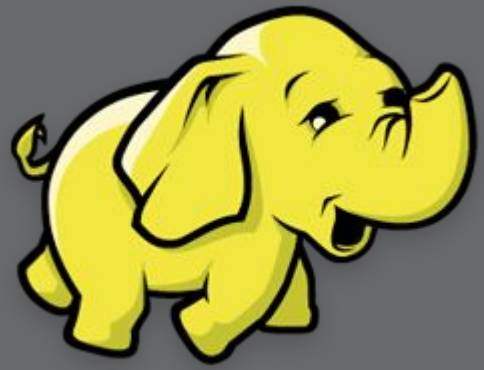
<u>Cloud</u>:
Software and/or hardware installed at a remote facility and provided to companies for a fee.



https://www.adeaca.com/blog/cloud-erp-lower-total-cost-of-ownership/#iLightbox[gallery11765]/0

Distributed computing (specifically: Hadoop)

galvanize



Apache Hadoop

galvanize

- **Hadoop** (the full proper name is Apache™ Hadoop®) is an open-source, **distributed computing framework** that was created to make it easier to work with big data.
- It provides a method to access and **process data that are distributed among multiple clustered computers**.
- "Hadoop" typically refers to four core components, though sometimes it refers to the ecosystem (next slide). The four components:
  - Hadoop Distributed File System (**HDFS**) - Manages and provides access to **distributed data**.
  - Hadoop YARN - Provides framework to schedule and manage jobs across the cluster.
  - Hadoop **MapReduce** - YARN-based parallel processing system for large datasets. MapReduce provides **computation on distributed data**.
  - Hadoop Common - A set of utilities that support the other three core modules.

See http://www.bmc.com/guides/hadoop-introduction.html

The Hadoop Ecosystem

See: https://hadoopecosystemtable.github.io/

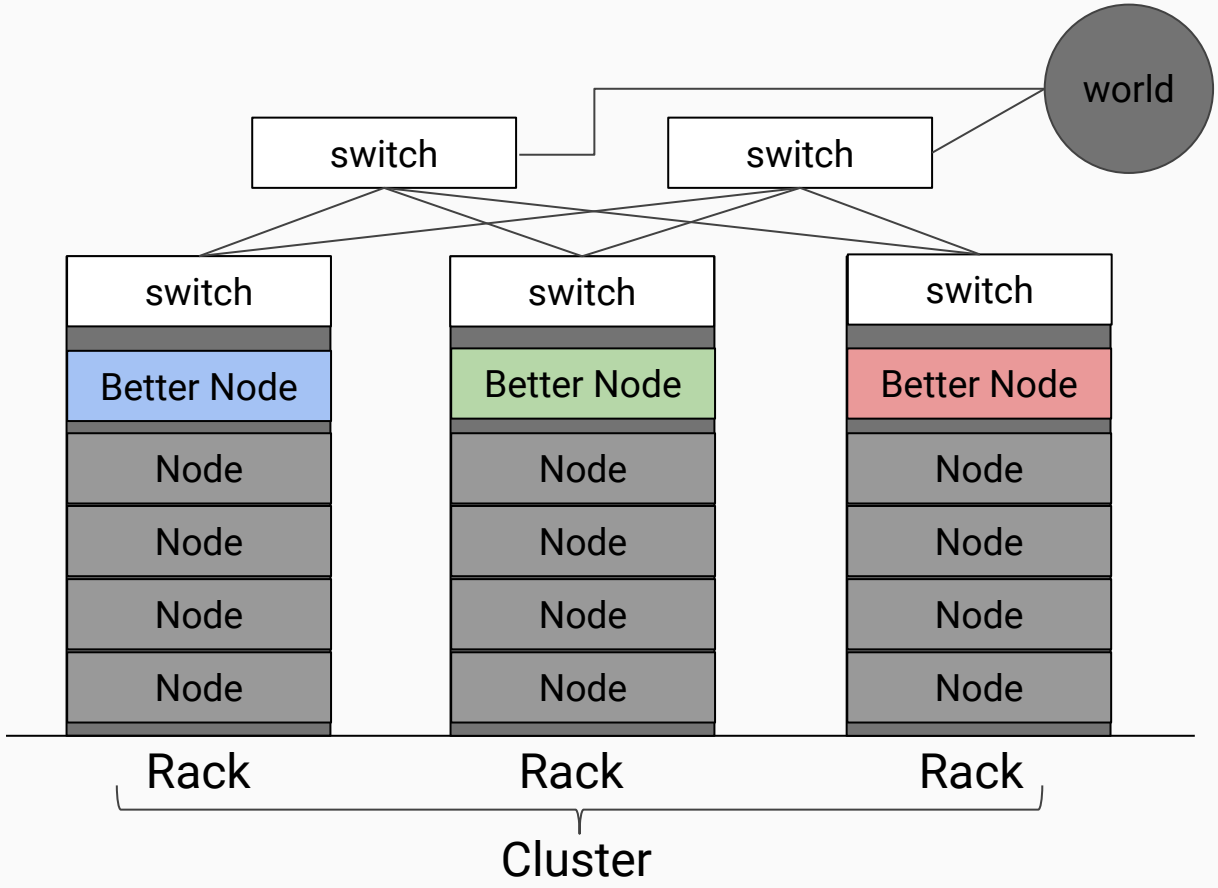
Headings:

- Distributed File System (HDFS is one)
- Distributed Computing (See MapReduce, Spark)
- SQL-on-Hadoop (See Hive)
- NoSQL databases
- NewSQL databases (!!)
- ...and many other headings

Reference:
Server Farms and Hadoop

Server farm (where Hadoop runs)

galvanize



A node (server), per Cloudera

galvanize

Typical Node Specifications

Here are the recommended specifications for DataNode/TaskTrackers in a balanced Hadoop cluster:

- 12-24 1-4TB hard disks in a JBOD (Just a Bunch Of Disks) configuration
- 2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz
- 64-512GB of RAM
- Bonded Gigabit Ethernet or 10Gigabit Ethernet (the more storage density, the higher the network throughput needed)

"Better" Node Specifications

Here are the recommended specifications for NameNode/JobTracker/Standby NameNode nodes. The drive count will fluctuate depending on the amount of redundancy:

- 4–6 1TB hard disks in a JBOD configuration (1 for the OS, 2 for the FS image [RAID 1], 1 for Apache ZooKeeper, and 1 for Journal node)
- 2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz
- 64-128GB of RAM
- Bonded Gigabit Ethernet or 10Gigabit Ethernet

Should use "enterprise class" components

HDFS software: DataNodes and the NameNode

galvanize

DataNode software

- DataNode installed on the nodes (servers) whose responsibility is to store and compute.
- It manages storing and locating blocks (chunks) of 64 MB of data on the node.
- It communicates with and responds to requests from the NameNode, including the “heartbeat.”
- Can communicate with other DataNodes (e.g. to copy data) and the Client.

NameNode software

- Tracks where data blocks are stored in the cluster.
- Interacts with client applications.
- Is the potential single point of failure for the entire HDFS. This is why there is a backup NameNode and its construction and components are “enterprise class”.
- Manages backing up of data blocks (generally stored in 3 different nodes in different racks).

MapReduce software: TaskTracker and JobTracker

galvanize

TaskTracker software

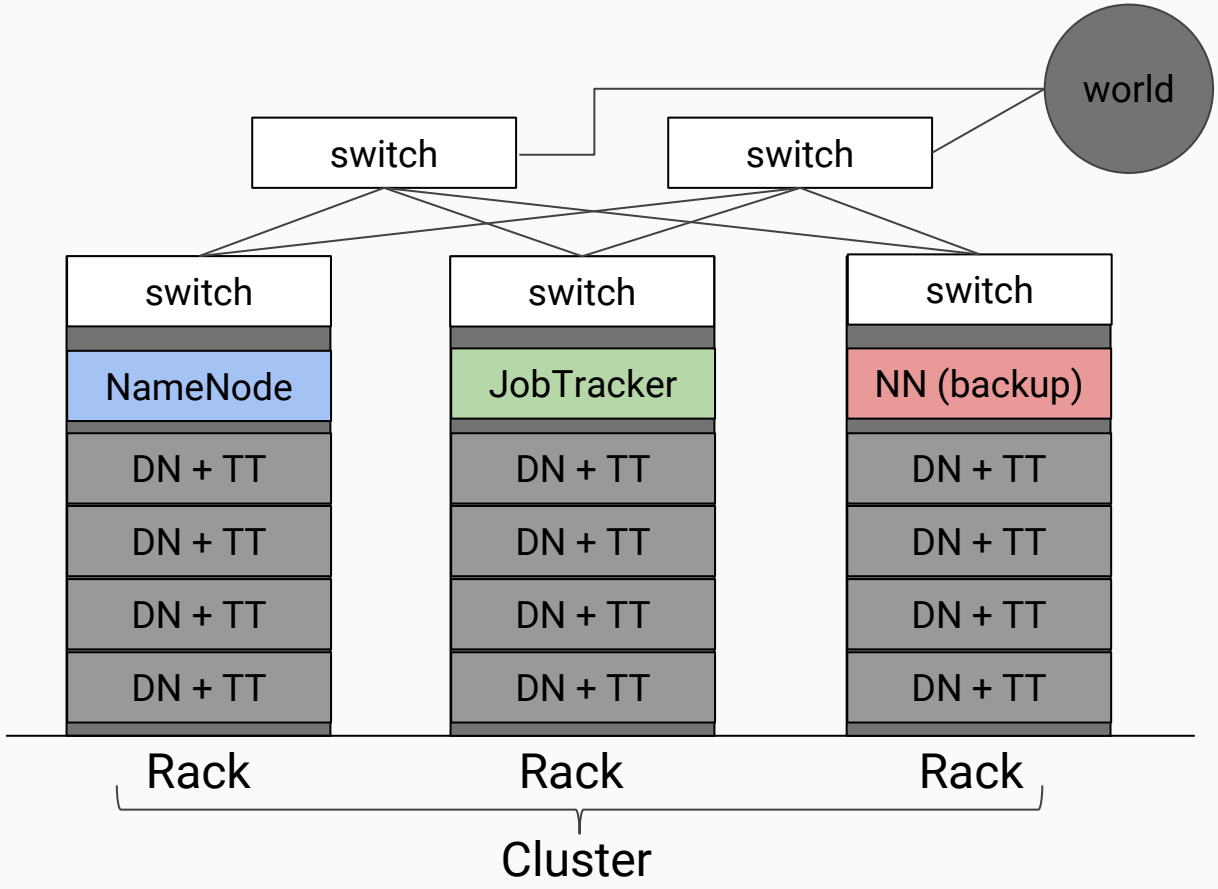
- Installed on nodes with the DataNode software.
- Performs the map, shuffle and sort, and reduce operations.
- Monitors status of these operations and reports progress to JobTracker. Also sends a “heartbeat” to JobTracker to indicate that it’s functioning properly.
- Can communicate with other TaskTrackers.

JobTracker software

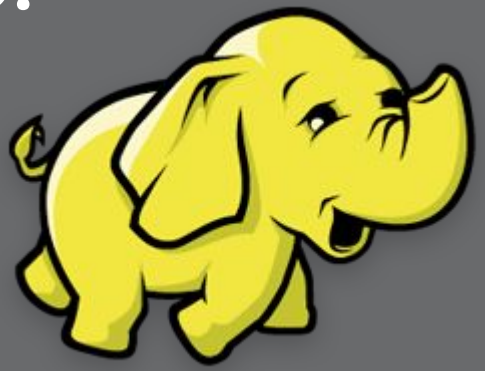
- Coordinates data processing.
- Interacts with NameNode to determine where data is stored.
- Will schedule a different TaskTracker if a TaskTracker doesn’t submit a “heartbeat” or has corrupt data.
- Communicates with the Client.
- Just like the NameNode, the JobTracker is given “enterprise” hardware.

Bring it all together: HDFS and MapReduce

galvanize



Distributed computing data processing software: Hadoop MapReduce



galvanize

MapReduce - the general idea and steps

galvanize

- **Send the computation to the data rather than trying to bring the data to the computation.**
- Computation and communication are handled as (key, value) pairs.
- In the “map” step, the **mapper** maps a function on the data that transforms it into (key, value) pairs. A **local combiner** may be run after the map to aggregate the results in (key, local aggregated values).
- After the mapping phase is complete, the (key, value) or (key, local aggregated value) results need to be brought together, sorted by key. This is called the “shuffle and sort” step.
- Results with the same key are all sent to the same MapReduce TaskTracker for aggregation in the **reducer**. This is the “reduce” step.
- The final reduced results are communicated to the Client.

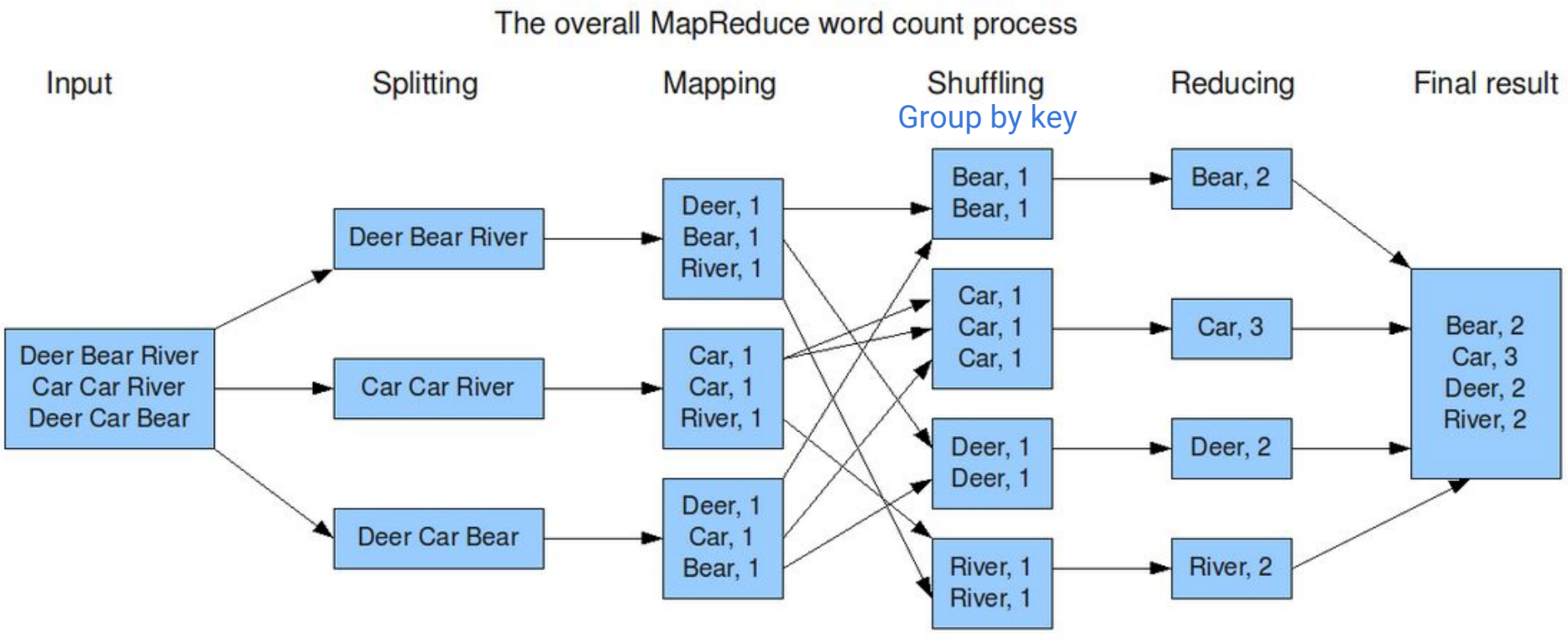
MapReduce: the general idea and steps

galvanize

- It's really Divide and Conquer:
  - 1) Split one large tasks into many smaller sub-tasks that can be solved in parallel on different nodes (servers)
  - 2) Solve these tasks independently
  - 3) Recombine the results of the sub-tasks for the final results.

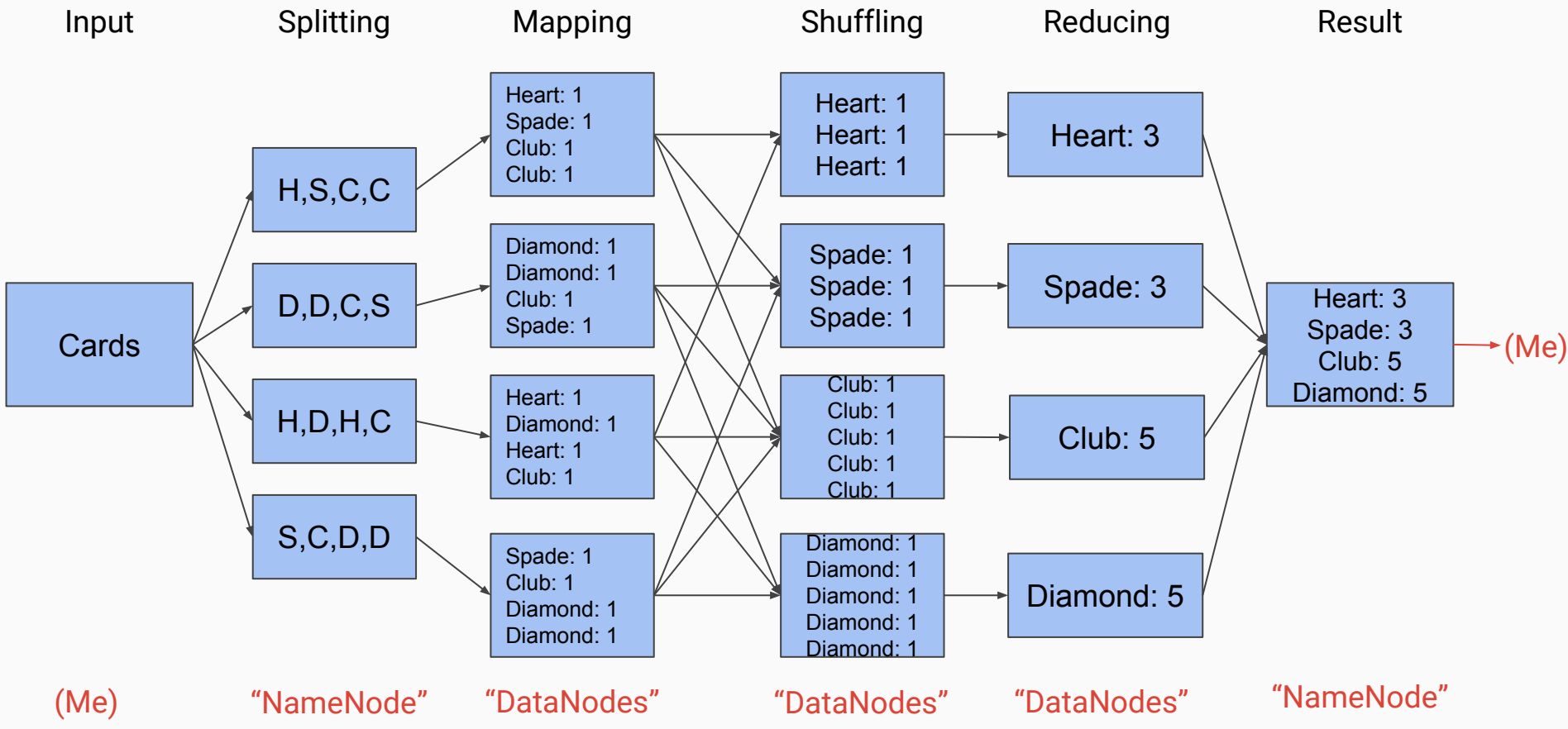
- The types of problems MapReduce is especially suited for:
  - Count (morning assignment), sum, avg, sort, graph traversal and analysis (optional afternoon assignment)
  - Some machine learning algorithms

MapReduce Illustration: Word Count



MapReduce Exercise: Counting Cards by Suit

galvanize



Take Aways

galvanize

- Hadoop - open-source framework made to handle big data through distributed computing.
- HDFS - data management component of Hadoop
  - NameNode - keeps track of where data is, makes sure it's backed up
  - DataNode - stores the data
- MapReduce - computation component of Hadoop
  - JobTracker - coordinates jobs, communicates with client
  - TaskTracker - performs computations on local data
    - A local mapper maps a function on data, perhaps using local combiner, then sends the results somewhere to be reduced
    - Data is handled as (key, value) pairs
    - All computations written to hard disk (for redundancy, but slow)
- Many other components in Hadoop Ecosystem

Breakout: generators_and_mrjob.ipynb

galvanize

References

- https://en.wikipedia.org/wiki/Apache_Hadoop
- https://wiki.apache.org/hadoop/