
# CPSC 436C

# Cloud Computing for Data Science

*Big Data*

Maryam R.Aliabadi

mraiyata@cs.ubc.ca

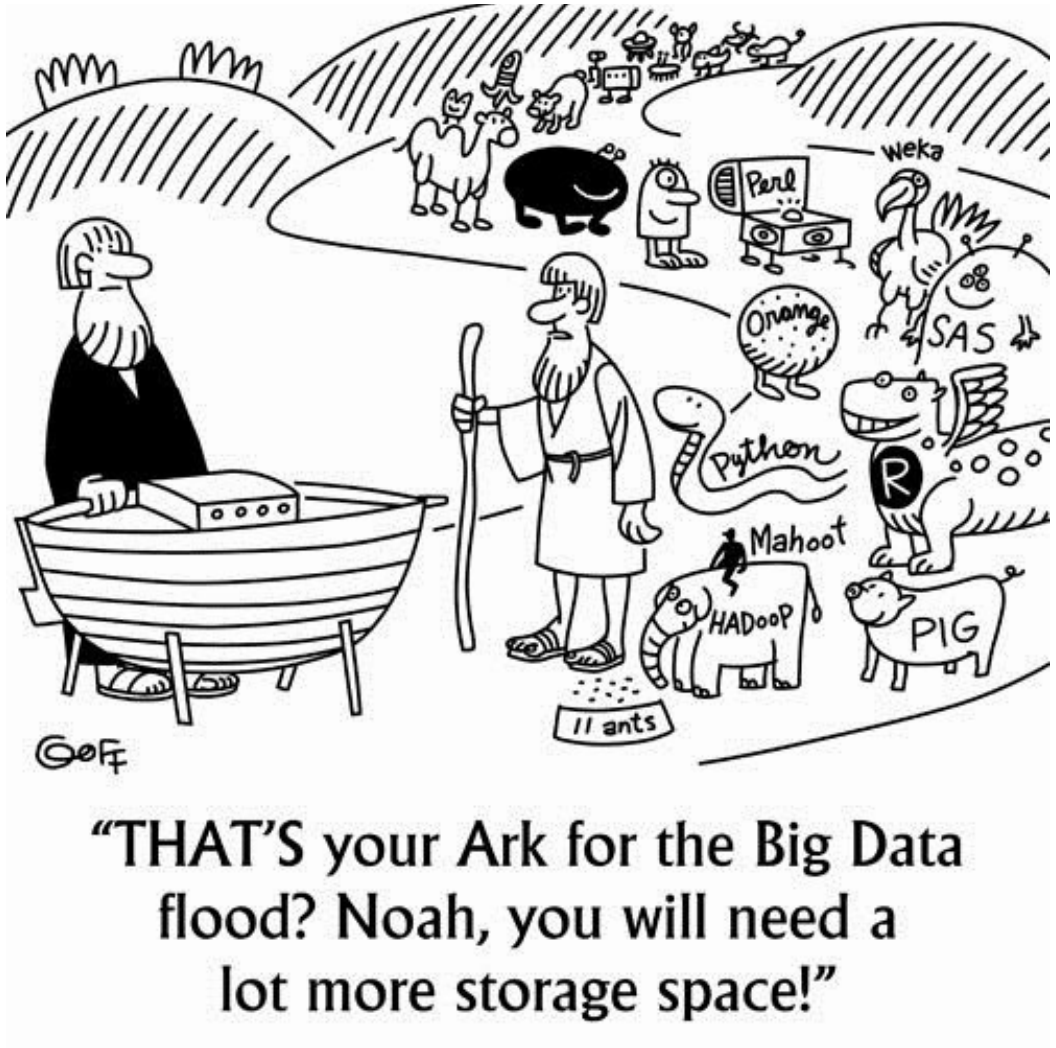
Fall 2023

# Last Week‘s Review

- Virtualization
- Virtualization types
- VM categories
- Partitioning
- VM Live migration
- How to launch a VM?

# Today's Topics

- Big data definition
- Big data properties
- Big data sources
- Big data analytics stack




"THAT'S your Ark for the Big Data flood? Noah, you will need a lot more storage space!"

# Big Data

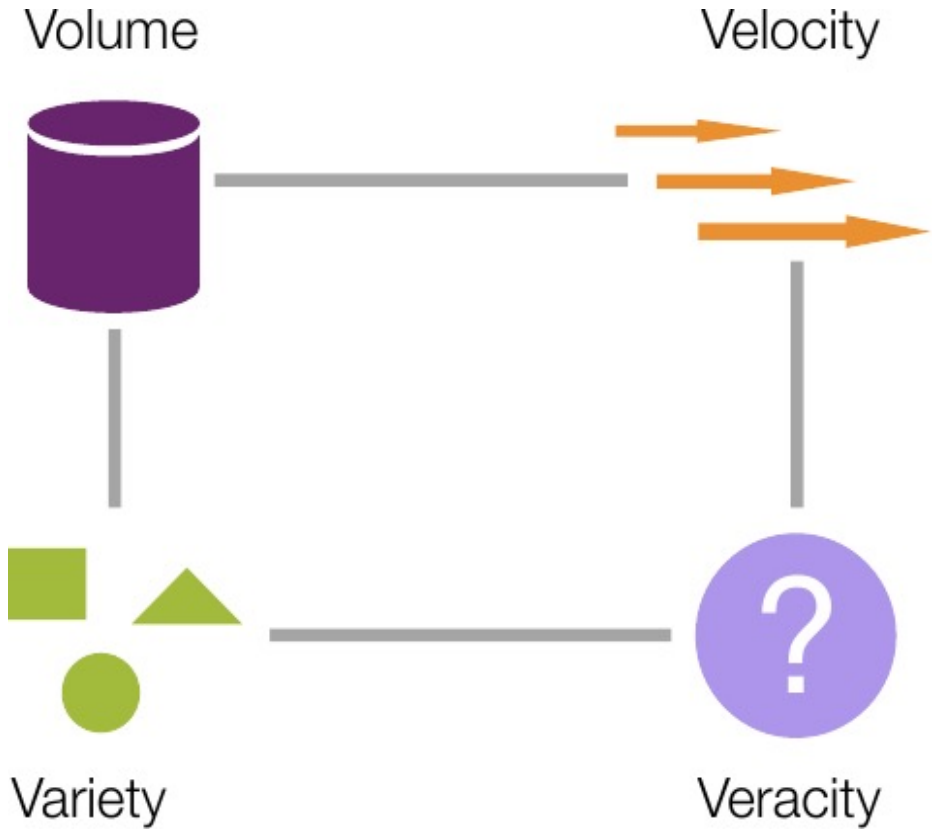
- Big Data refers to datasets and flows large  enough that has outpaced our capability to store, process, analyze, and understand.

small data


big data

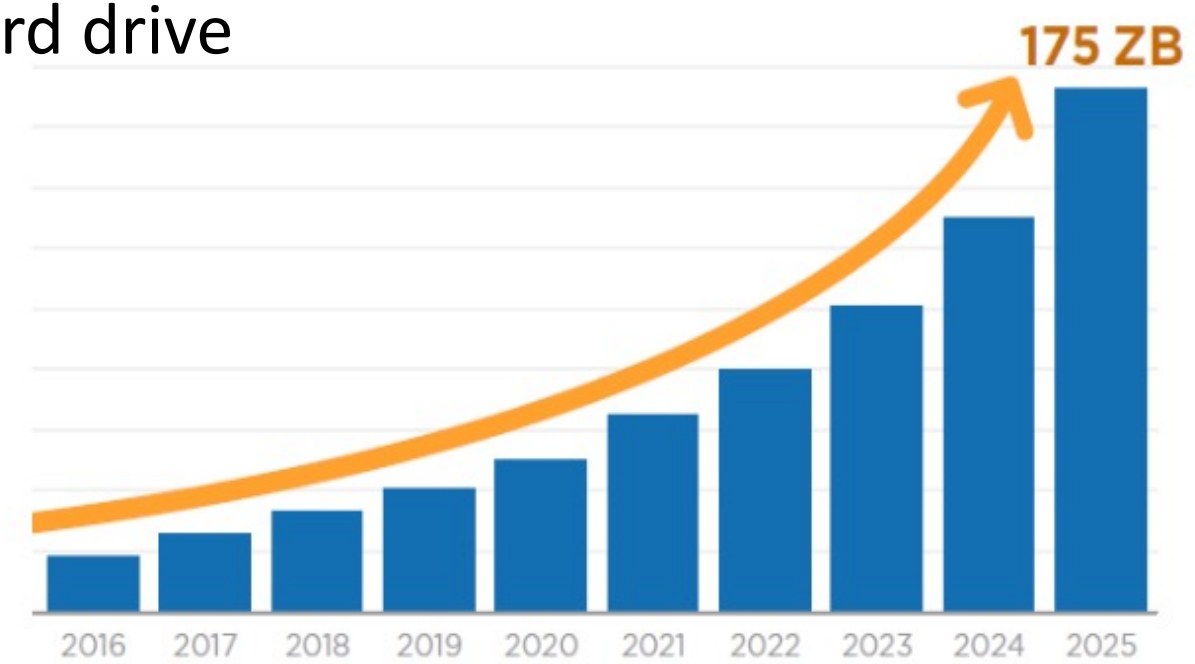
# Four Attributes of Big Data

- Volume: data size
- Velocity: data generation rate
- Variety: data heterogeneity
- Veracity: data quality



# Volume

- More data than fits in a computer's RAM
- More data than fits on a single hard drive
- Facebook's 2+billion users



# Velocity

- Rapid generation of new data
- Nearly 200 million emails are sent each minute of each day
- Nearly 5 billion videos are watched on YouTube every day

# Variety

- Data in many formats
- Videos, photos, audio
- GPS coordinates
- Social network connections

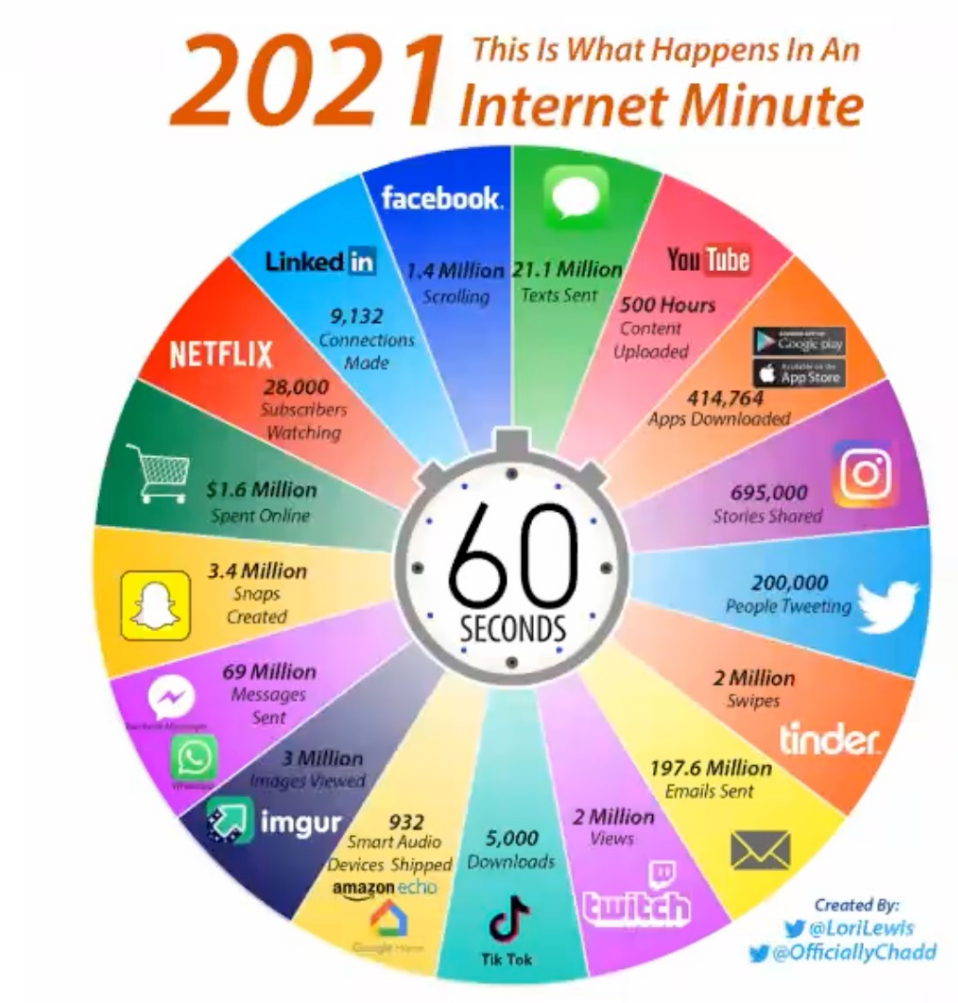
# Veracity

- Data reliability and trustworthiness
- Important to make informed decisions or draw meaningful insights.
- Data quality processes, data validation procedures, and data governance practices are required to maintain and improve the accuracy and trustworthiness of their data.

# Where does big data come from?

# Big Data Market Driving Factors

- Social Media
  - Social media begets more social media
  - Posts get liked
  - Images get tags
  - Followers share content



# Big Data Market Driving Factors

- Internet of Things (IoT)
  - IoT sensors
  - Smart homes
  - Smart grids
  - Self-driving cars

- More than 65 billion devices were connected to the Internet by 2010, and this number exceeded 230 billion by 2020.

∗"The Internet of Things Is Coming" [John Mahoney et al., 2013]

# How to store and process big data?

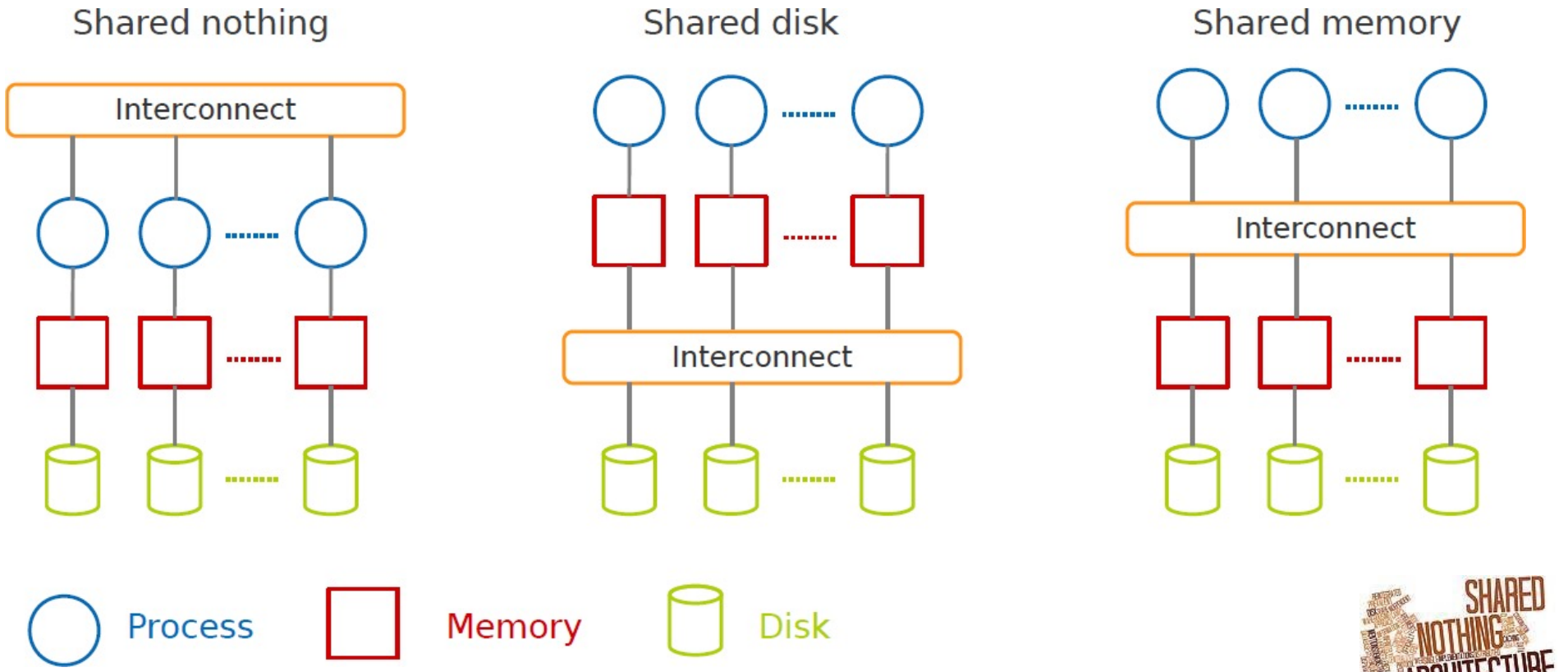
# Scale Up vs. Scale Out

- Scale up or scale vertically: adding resources to a single node in a system.
- Scale out or scale horizontally: adding more nodes to a system.

# Scale Up vs. Scale Out

- Scale up: more expensive than scaling out.
- Scale out: more challenging for fault tolerance and software development.

# Taxonomy of Parallel Architectures





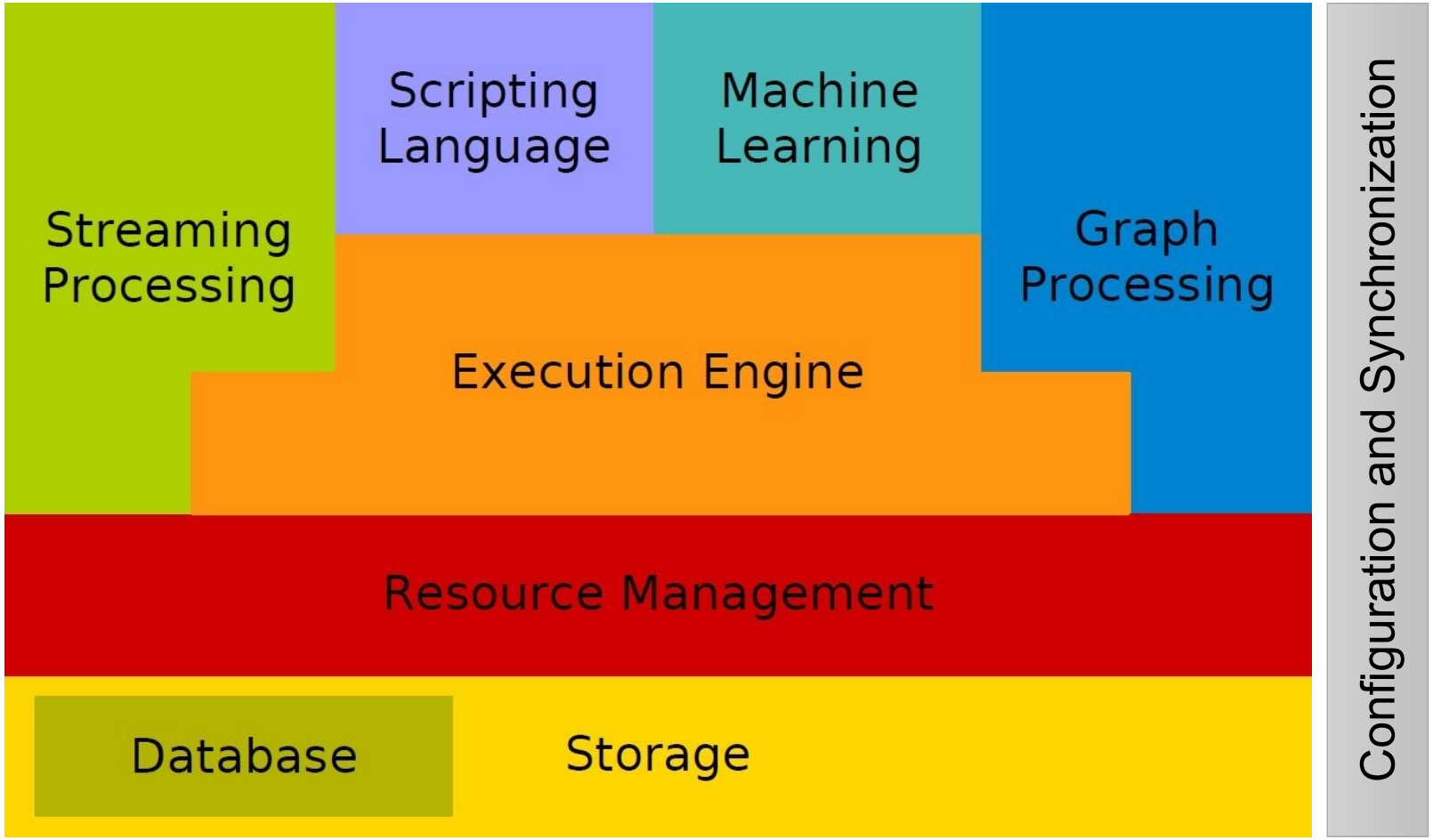


DeWitt, D. and Gray, J. "Parallel database systems: the future of high performance database systems". ACM Communications, 35(6), 85-98, 1992.

# Big Data Tools and Frameworks

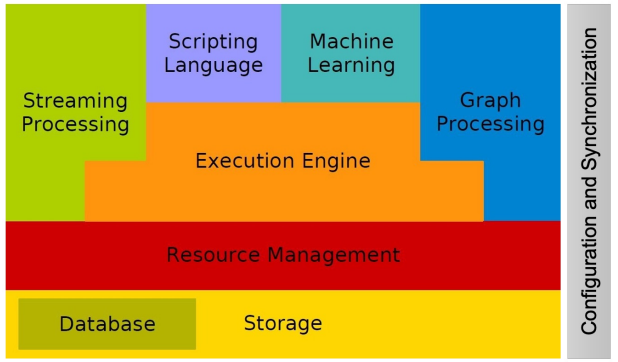
- Two main types of tools:
  - Data Store
  - Data Processing

# Big Data Analytics Stack



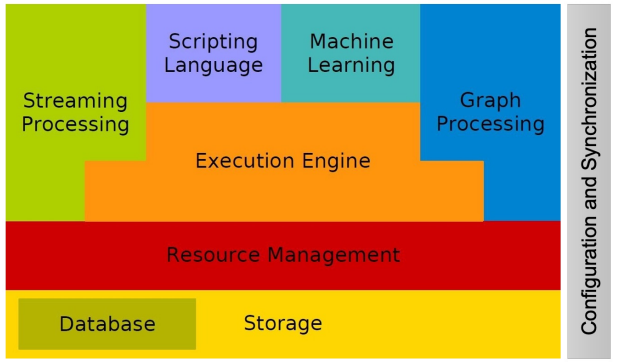
# Big Data – Storage (File systems)

- Traditional file-systems are not well-designed for large-scale data processing systems.
- Efficiency has a higher priority than other features, e.g., directory service.
- Massive size of data tends to store it across multiple machines in a distributed way.
- HDFS/GFS, Amazon S3, ...



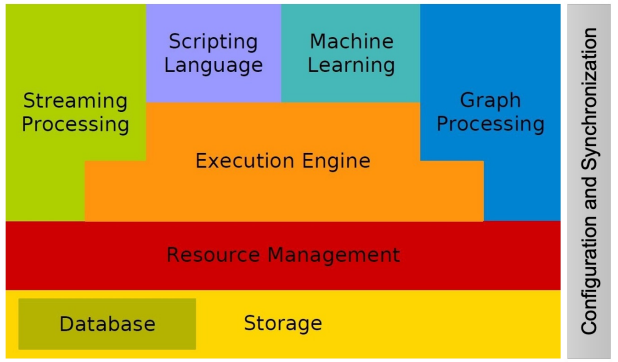
# Big Data - Database

- Relational Databases Management Systems (RDMS) were not designed to be distributed.
- NoSQL databases relax one or more of the ACID properties:
  - BASE (Basically Available, Soft state, Eventually consistent).
- Different data models: key/value, column-family, graph, document.
- NoSQL database exeamples: Hbase/BigTable, Dynamo, Scalaris, Cassandra, MongoDB, Voldemort, Riak, Neo4J, ...



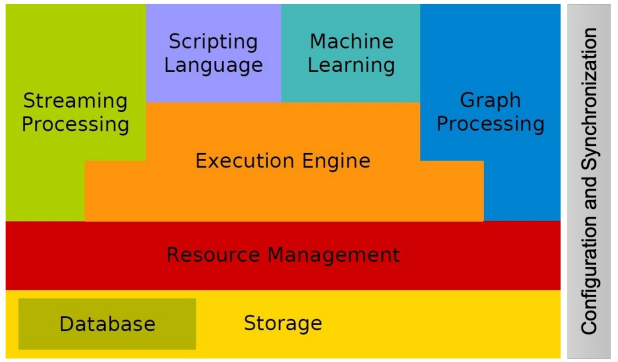
# Big Data – Resource Management

- Different frameworks require different computing resources.
- Large organizations need the ability to share data and resources between multiple frameworks.
- Resource management share resources in a cluster between multiple frameworks while providing resource isolation.
- Mesos, YARN, Borg, …



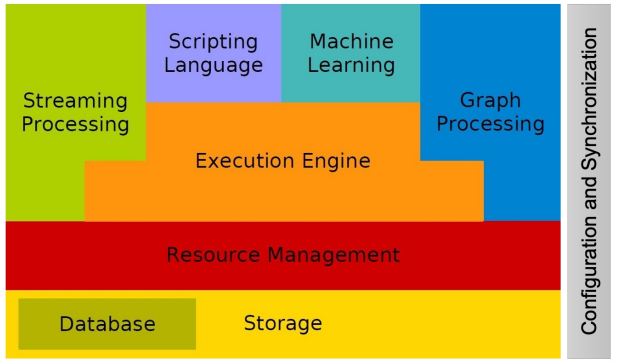
# Big Data- Execution Engines

- Scalable and fault tolerance parallel data processing on clusters of unreliable machines.
- Data-parallel programming model for clusters of commodity machines.
- MapReduce, Spark, Stratosphere, Dryad, Hyracks, ...



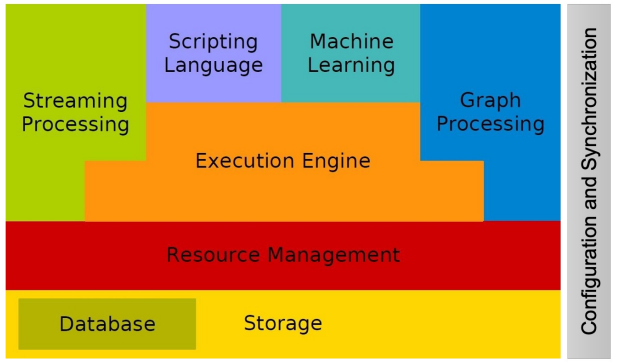
# Big Data – Query/Scripting Languages

- Low-level programming of execution engines, e.g., MapReduce, is not easy for end users.
- Need high-level language to improve the query capabilities of execution engines.
- It translates user-defined functions to low-level API of the execution engines.
- Pig, Hive, Shark, Meteor, DryadLINQ, SCOPE, ...



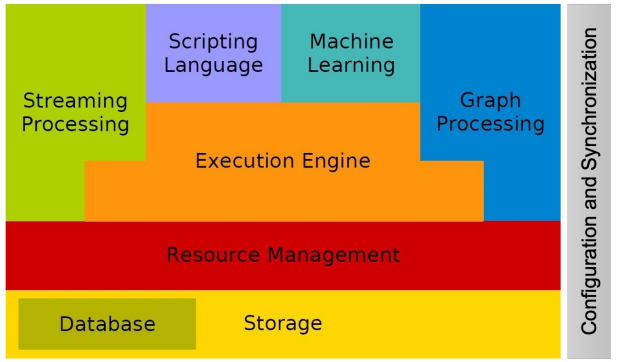
# Big Data: Graph Processing

- Many problems are expressed using graphs: sparse computational dependencies, and multiple iterations to converge.
- Data-parallel frameworks, such as MapReduce, are not ideal for these problems: slow
- Graph processing frameworks are optimized for graph-based problems.
- Pregel, Giraph, GraphX, GraphLab, PowerGraph, GraphChi, …



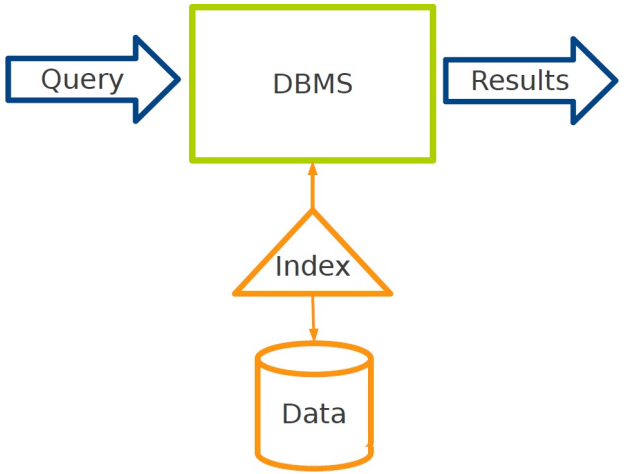
# Big Data – Machine Learning

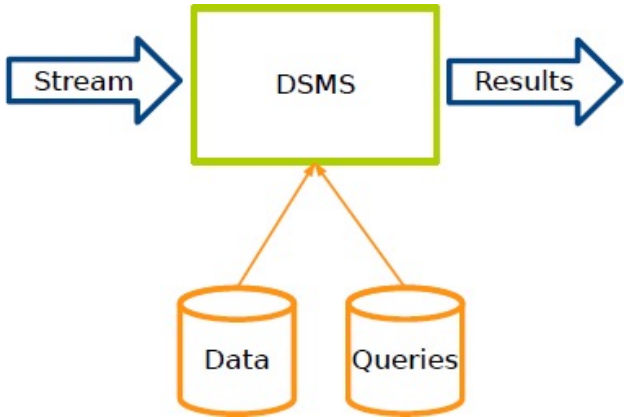
- Implementing and consuming machine learning techniques at scale are difficult tasks for developers and end users.
- There exist platforms that address it by providing scalable machine-learning and data mining libraries.
- Mahout, MLBase, Tensorflow, ...

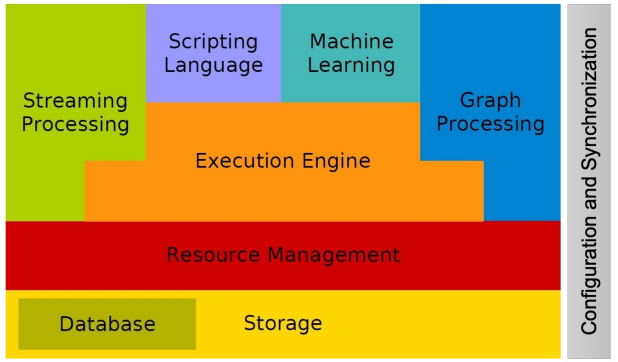


# Big Data – Stream Processing

- Providing users with fresh and low latency results.
- Database Management Systems (DBMS) vs. Data Stream Management Systems (DSMS)
- Storm, S4, SEEP, D-Stream, Naiad, ...

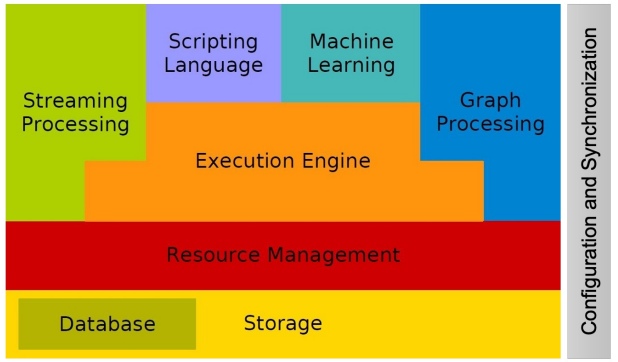


# Big Data – Configuration and Synchronization

- A means to synchronize distributed applications accesses to shared resources.
- Allows distributed processes to coordinate with each other.
- Zookeeper, Chubby, ...



# Recap

- Big data definition
- Big data properties
- Big data sources
- Big data analytics stack

# Next Topic: Data Store

# Module 1 Quiz

- https://join.iclicker.com/XXOJ

# Resource Utilization

1. Compare the resource utilization efficiency of virtual machines (VMs), containers, and serverless functions. In which scenario do you expect the most efficient resource usage, and why?
   a) VM
   b) Container
   c) Serverless
   d) All of them
   e) None of them

# Isolation and Security

2. How does containerization technology, such as Docker, achieve process isolation compared to virtual machines and serverless? What one is more secure?

a) VM
b) Container
c) Serverless
d) All of them
e) None of them

# Portability

3. Compare the concept of portability of virtual machines in contrast to containers and serverless functions. What service does bring portability to cloud deployments?

a) VM
b) Container
c) Serverless
d) All of them
e) None of them

# Scaling and Cost

4. Compare how VMs, containers, and serverless computing handle auto-scaling based on workload demand. Which approach is likely to be more cost-effective for a highly variable workload?

a) VM
b) Container
c) Serverless
d) All of them
e) None of them

# Cold Starts and Latency

5. Which computing service has the minimum "cold starts" and latency when responding to requests?

a) VM
b) Container
c) Serverless
d) All of them
e) None of them