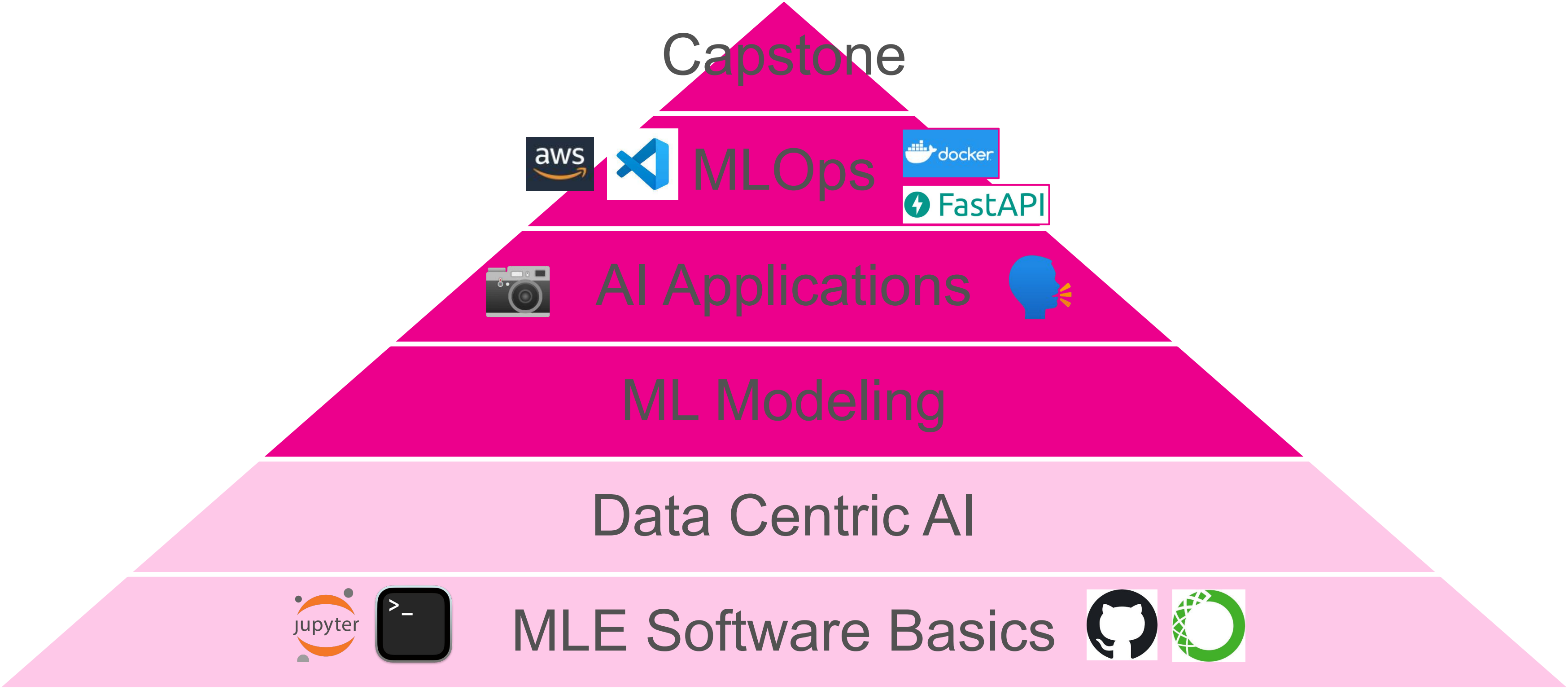
FourthBrain

MLE Program, Cohort 10 (MLE10)

Week 5: Big Data Processing and Big Data Tools

Becoming a Machine Learning Engineer



Capstone

MLOps

AI Applications

ML Modeling

Data Centric AI

MLE Software Basics

Our Updated Curriculum!

1. ML Project Scoping
2. Real, Live Data Streams
3. Data Wrangling & Exploratory Analysis
4. Big Data

DATA CENTRIC AI

5. Supervised ML
6. Deep Learning & AutoML
7. Unsupervised, Semi- & Self-supervised Learning

ML MODELING

8. Computer Vision
9. Natural Language Processing
10. Transformers & Fine Tuning Pre-Trained Networks

AI APPLICATIONS

11. Building ML Web Apps
12. Containerization
13. Model Serving
14. Machine Learning in Production

MLOps

Last Week

Concepts

- Exploratory Data Analysis
- Feature Engineering
- One-Hot Encoding
- Feature Selection
- Data Leakage

Hands-on

- Exploring and wrangling structured data for sales prediction

🤖 This Week!

Concepts

- Big Data Processing
- Big Data Tools
- Good Data over Big Data

Hands on

- Exploring and wrangling a big data set to predict customer/client behavior using an ML pipeline

What questions do you have?

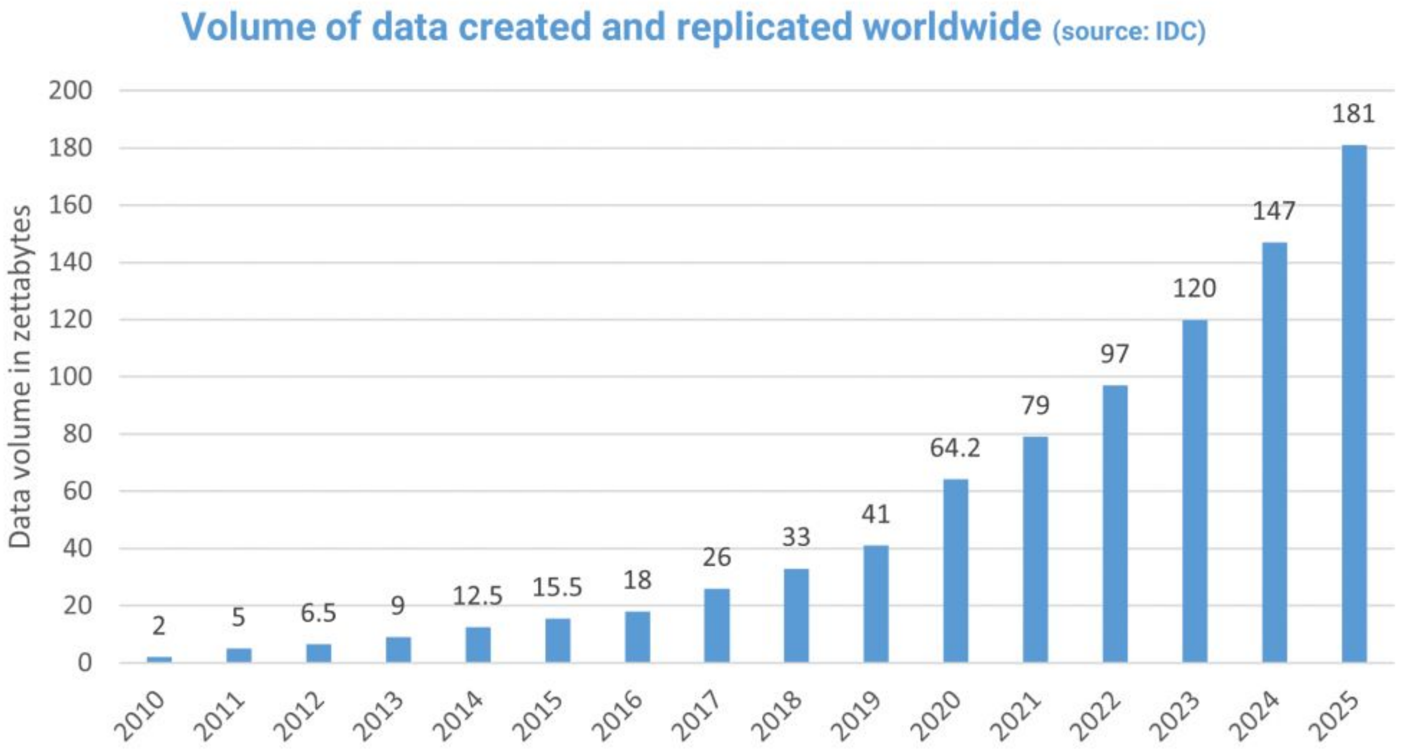
LET'S SOLVE THIS PROBLEM BY USING THE BIG DATA NONE OF US HAVE THE SLIGHTEST IDEA WHAT TO DO WITH
TOM FISHBURNE
© marketoonist.com

In Short : What is Big Data?

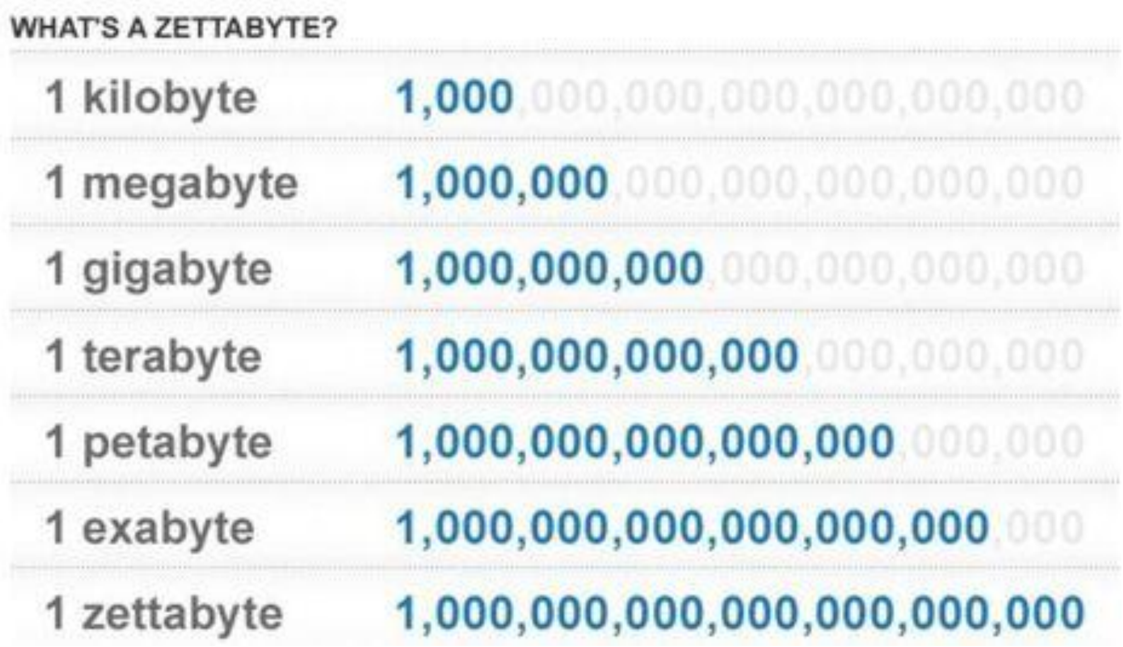
Big data refers to extremely large sets of structured and unstructured data that cannot be handled with traditional methods. Big data analytics can make sense of the data by uncovering trends and patterns.

Boring! isn’t it?

Why should we talk about Big data?



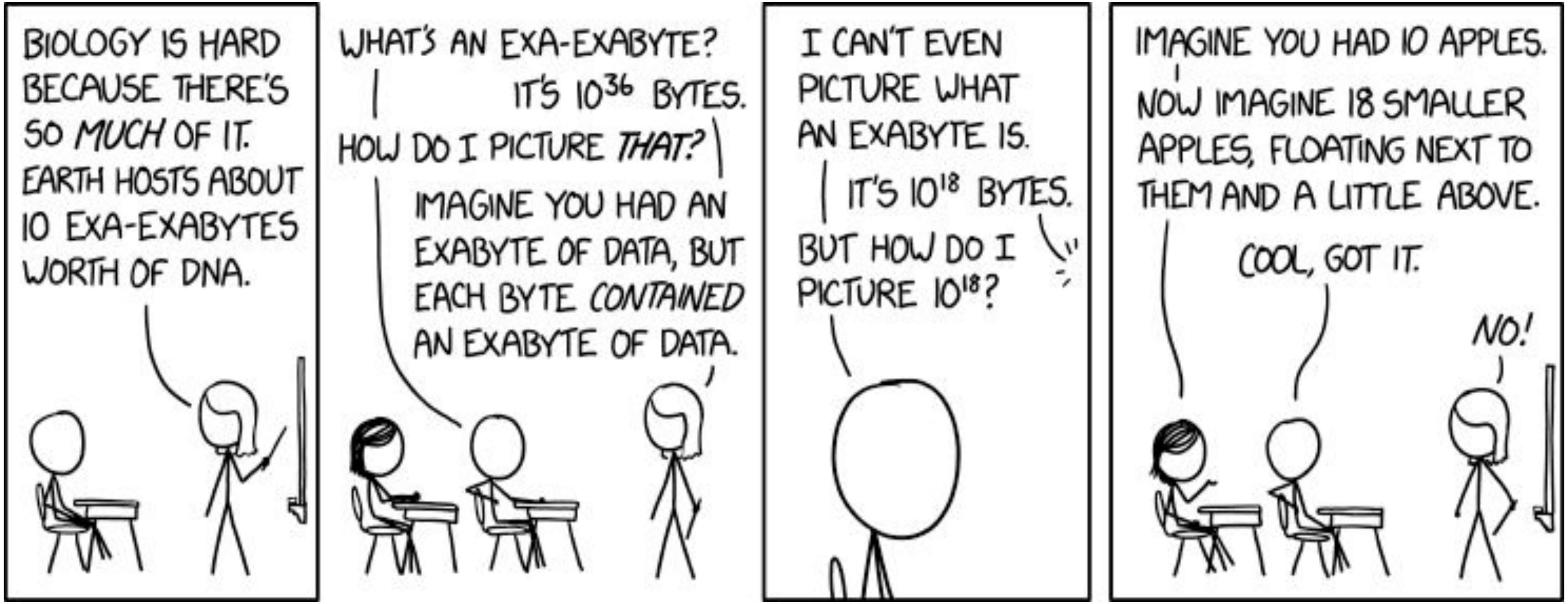
What is a Zetabyte?



WHAT'S A ZETTABYTE?

1 kilobyte	1,000,000,000,000,000,000,000
1 megabyte	1,000,000,000,000,000,000,000
1 gigabyte	1,000,000,000,000,000,000,000
1 terabyte	1,000,000,000,000,000,000,000
1 petabyte	1,000,000,000,000,000,000,000
1 exabyte	1,000,000,000,000,000,000,000
1 zettabyte	1,000,000,000,000,000,000,000

Putting things into perspective!


BIOLOGY IS HARD BECAUSE THERE'S SO *MUCH* OF IT. EARTH HOSTS ABOUT 10 EXA-EXABYTES WORTH OF DNA.
WHAT'S AN EXA-EXABYTE?
IT'S 10^{36} BYTES.
HOW DO I PICTURE *THAT?*
IMAGINE YOU HAD AN EXABYTE OF DATA, BUT EACH BYTE *CONTAINED* AN EXABYTE OF DATA.
I CAN'T EVEN PICTURE WHAT AN EXABYTE IS.
IT'S 10^{18} BYTES.
BUT HOW DO I PICTURE 10^{18}?
IMAGINE YOU HAD 10 APPLES. NOW IMAGINE 18 SMALLER APPLES, FLOATING NEXT TO THEM AND A LITTLE ABOVE.
COOL, GOT IT.
NO!

But wait! isn’t this a Machine Learning course?

Why should a Machine Learning Engineer care about big data?

Movie Lens Dataset

Movie review dataset with 27,000,000 ratings and 1,100,000 tag applications applied to 58,000 movies by 280,000 users

movie.csv - contains details about movies
rating.csv - contains user rating for each movie

MovieLens Task

Task 1

Join movie and rating data to get user rating and movie details for each movie in a single table

Task 2

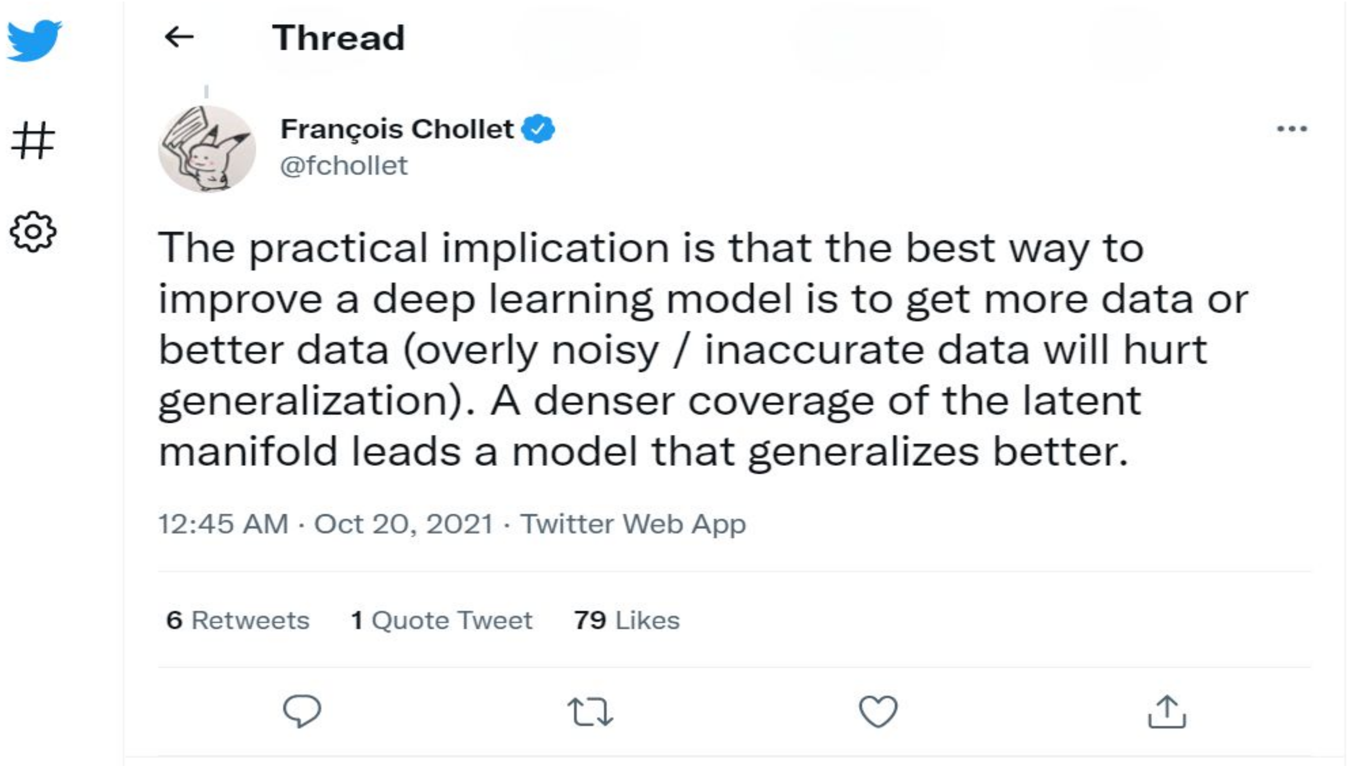
Join the resultant table from task 1 with itself - to get the movie rating for each pair of users (usually this operation is required for recommendation systems aka “you may like” in netflix)

Let’s dive into a colab notebook to play with this data

Lets try a small demo

https://colab.research.google.com/drive/17GH7jX7ONf-KQ8jdG7_3WjglTcxl2SE4?authuser=2#scrollTo=iMFewAiiPd2M

Important!!! Good Data vs Big data



Thread

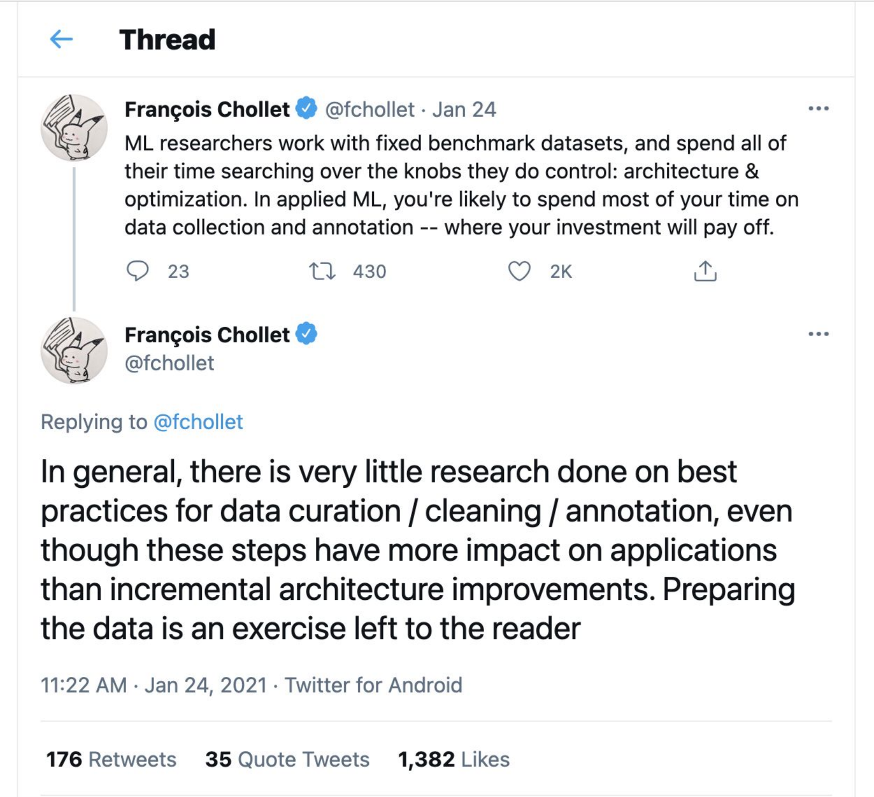
François Chollet
@fchollet

The practical implication is that the best way to improve a deep learning model is to get more data or better data (overly noisy / inaccurate data will hurt generalization). A denser coverage of the latent manifold leads a model that generalizes better.

12:45 AM · Oct 20, 2021 · Twitter Web App

6 Retweets 1 Quote Tweet 79 Likes

Big Data vs Good Data



Thread

François Chollet @fchollet · Jan 24

ML researchers work with fixed benchmark datasets, and spend all of their time searching over the knobs they do control: architecture & optimization. In applied ML, you're likely to spend most of your time on data collection and annotation -- where your investment will pay off.

23 430 2K

François Chollet
@fchollet

Replying to @fchollet

In general, there is very little research done on best practices for data curation / cleaning / annotation, even though these steps have more impact on applications than incremental architecture improvements. Preparing the data is an exercise left to the reader

11:22 AM · Jan 24, 2021 · Twitter for Android

176 Retweets **35** Quote Tweets **1,382** Likes

Good data vs Bad data

“The question isn’t necessarily “Is big data good or bad?”, but rather “How effectively and efficiently can large data sets be organized, stored, and analyzed to produce actionable intelligence?”

ref :https://planergy.com/blog/is-big-data-good-or-bad/

Now let’s define what big data is

Five V’s of Big Data

- **Volume:** Big data is based on volume. Quantum volume determines how big the data is. Usually contains a large amount of data in terabytes, petabytes, etc. Based on volume size, data scientists plan various tools and integrations for data set analysis.

- **Velocity:** The speed of data collection is critical because some companies require real-time data information, and others prefer to process data in packets. The faster the data flow, the more data scientists can evaluate and provide relevant information to the company.

- **Variety:** Data comes from different sources and, importantly, not in a fixed format. Data is available in structured (database format), semi-structured (XML/RDF) and unstructured (binary data) formats. Based on data structures, big data tools are used to create, organize, filter, and process data.

The Five V’s of Big Data

- **Veracity:** The Data accuracy and credible sources define the big data context. The data set comes from various sources such as computers, network devices, mobile devices, social media, etc. Accordingly, the data must be analyzed to be sent to its destination.

- **Value:** Finally, how much is a company's big data worth? The role of the data scientist is to make the best use of data to demonstrate how data insights can add value to a business.

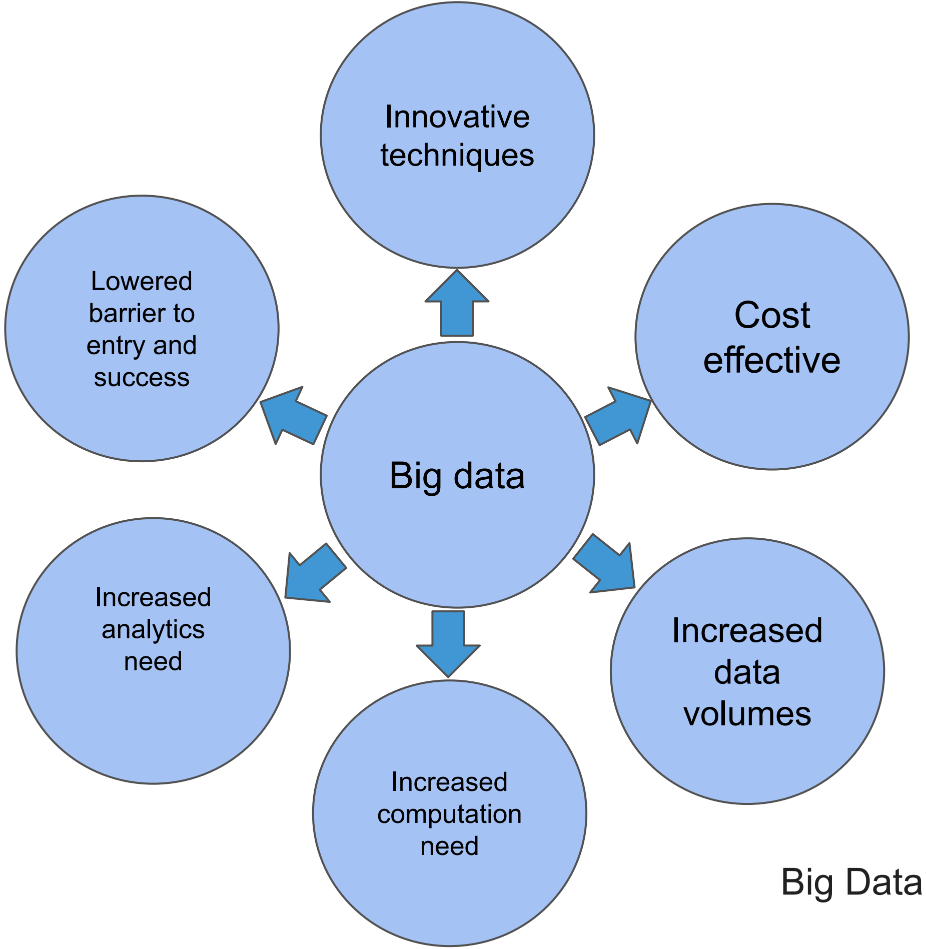
Let’s talk more about the Vs

- **Volume – Size and Scale**
  - Up to 40,000 sensors in the Airbus A380
  - 7 TB per day
- **Velocity – Data Rate and Streaming Data**
  - Sensor data collected in msec
  - Type and No of Sensors
- **Variety – Cross Media Data**
  - Sensors
  - Images and videos
  - Text data
  - Relational business data
- **Validity - Reliability**
  - Poor data quality
  - Missing data
  - Data collected doesn't suit targeted use cases
- **Value – Usefulness and Importance**
  - Data per se is not valuable
  - How to extract real value from data?

Data is the new oil!

	Traditional Analytics	Big data Analytics
Focus on	• More Descriptive analytics focus • Diagnosis analytics	• Both Descriptive and Predictive analytics • Data science
Data sets	• Limited data sets • Cleansed data • Simple models	• Large scale data sets • More types of data • Raw data • Complex data models
Result	• Insights are noisy and not generalizable	• **Insights are more generalizable. Augmenting multiple sources of data can lead to more powerful business insights.**

What made big data needed?



Big Data Analytics”, David Loshin

Three phases of big data

Phase 1 Period: 1970-2000	Phase 2 Period: 2000-2010	Phase 3 Period: 2010-present
DBMS-based, structured content: ● RDBMS & data warehousing ● Extract Transfer Load ● Online Analytical Processing ● Dashboard & Scorecards ● Data mining & Statistical analysis	web-based, unstructured content ● Information retrieval and extraction ● Opinion mining ● Question answering ● Web analytics and web intelligence ● Social media analytics ● Social Network analytics ● Spatial-temporal analytics	Mobile and sensor-based content ● Location-aware analysis ● Person-centered analysis ● Context-relevant analysis ● Mobile visualization ● Human-Computer-Interaction

Introduction to Big data (link)

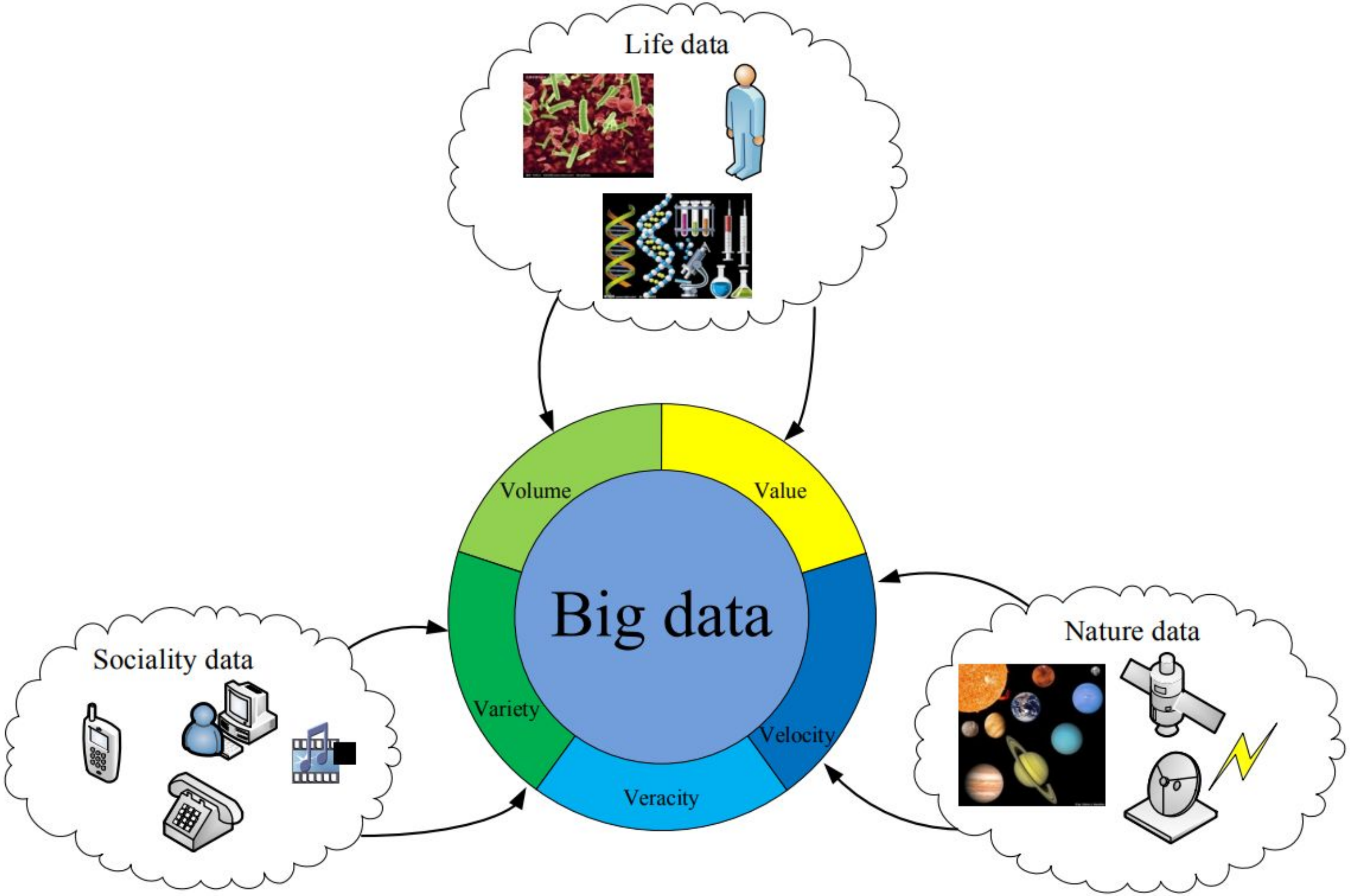
Breakout

5 min
(3-4 per room)

When have you come across the word “Big data”? How has it impacted you so far? Tell us some big data story if you have

Designate *one person to share* from your breakout room

BIG DATA
BIG DATA EVERYWHERE
makeameme.org


Life data
Big data
Volume
Value
Variety
Velocity
Veracity
Sociality data
Nature data

Applications of Big Data

Applications: Carrying out market research and segmentation

- The target audience is the cornerstone of any business. Every enterprise needs to understand the audience and market that it wants to target in order to be successful. That is the reason enterprises need to carry out market research that can delve deep into the minds of potential customers and provide insightful data.

- Machine learning can help in this regard by using supervised and unsupervised algorithms to interpret consumer patterns and behaviors accurately. Media and the entertainment industry use machine learning to understand the likes and dislikes of their audiences and target the right content to them.

Applications: Exploring Customer Behavior

- Machine learning does not stop after drawing a picture of your target audience. It also helps businesses explore audience behavior and create a solid framework of their customers. This system of machine learning, known as user modeling, is a direct outcome of human-computer interaction.

- It mines data to capture the mind of the user and enable business enterprises to make intelligent decisions. Facebook, Twitter, Google and others rely on user modeling systems to know their users inside out and make relevant suggestions.

Applications: Personalized Recommendations

- Businesses need to offer personalization to their customers. Be it a smartphone or a web series, companies need to establish a strong connection with their users to deliver what's relevant to them. Big data machine learning is best used in a recommendation engine. It combines context with user behavior predictions to influence user experience based on their activities online.

- This way, it can empower businesses to make correct suggestions that customers find interesting. Netflix uses machine learning-based recommender systems to suggest the right content to its viewers.

Applications: Predicting trends

- Machine learning algorithms use big data to learn future trends and forecast them to businesses. With the help of interconnected computers, a machine learning network can constantly learn new things on its own and improve its analytical skills every day.

- In this way, it not just calculates data but behaves like an intelligent system that uses past experiences to shape the future.

- Example: A car company can depend on machine learning to predict the demand for a specific brand of car in the next year (hybrid, electric or gasoline) and plan its production accordingly.

Applications: Aiding decision-making

- Machine learning uses a technique called time series analysis that is capable of analyzing an array of data together. It is a great tool for aggregating and analyzing data and makes it easier for managers to make decisions for the future.

- Businesses, especially retailers, can use this ML-boosted method to predict the future with commendable accuracy.

Applications: Decoding patterns

- Machine learning can be highly efficient to decipher data in industries where understanding consumer patterns can lead to major breakthroughs.

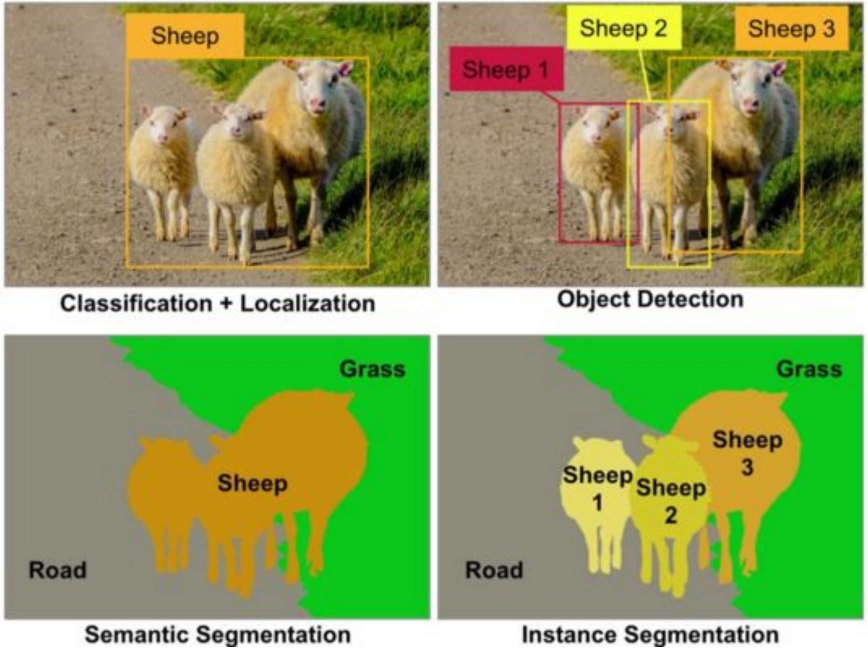
- For example, sectors like healthcare and pharmaceuticals have to deal with a lot of data. Machine learning can help them analyze the data to identify diseases in the initial stage among patients.

- Machine learning can also allow hospitals to manage patient services better by analyzing past health reports, pathological reports and disease histories. All of these can lead to better diagnoses at healthcare centers and boost medical research in the long run.

Applications: Computer Vision

- Object recognition, 3D-modeling, medical imaging, and smart cars are all examples of what current computer vision systems can do.

- A fundamental challenge of large-scale object recognition is how to attain proficiency in both feature extraction and classifier training without conceding performance



Applications: Information Retrieval

- Information retrieval is the process of providing users with the multimedia objects that will satisfy their information need. Performing fast retrieval of information becomes a problem area in Big Data as there are massive amounts of data, such as text, audio, image, and video.

- These multimedia files are collected and readily available across several domains. Deep networks are mainly utilized in information retrieval for extracting semantically meaningful features for subsequent object ranking stages

BREAK

Problems with Big Data and Machine Learning

- **High dimensional data – Find the relevant features?** Needs the involvement of domain experts and/or automatic feature selection techniques

- **Evaluating Data Quality:** Big data doesn’t always mean good quality data. Ensuring quality is not straightforward.

  No labels: Use unsupervised learning algorithms (e.g. anomaly detection)

- **Efficient Storage and Retrieval:** Big data comes with additional storage cost. Storing more means also retrieving more.

- **Compute Cost:** Big data comes with additional compute requirements. How to optimize this cost depends on compute system used.

- **Multimodal Data:** Big data can come in different formats - like text, images etc.
- **Handling Data Velocity:** Streaming vs Batch processing.

The 3 key Questions to ask!

1. **How to store the data?**

1. **How to process the data? -** From data processing to running machine learning algorithm

1. **How to visualize/build dashboards?**

Breakout

5 min
(3-4 per room)

What are some applications/capstone project idea you can think of which requires Big Data? Have you used any specific tools?

Think in terms of the 3 key challenges - processing, storage and visualization

Designate *one person to share* from your breakout room

Design Decisions as an MLE

Key challenges:

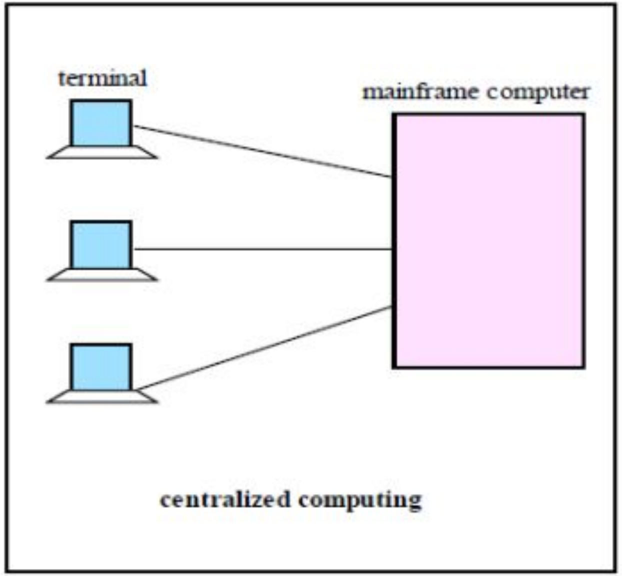
1. Data Storage - How to store efficiently?
2. Data processing - How to run analytics and build algorithm
3. Data Mining Platform/ Data Visualization - Collaborative and integrated enterprise level platform for efficient big data analytics and visualization.

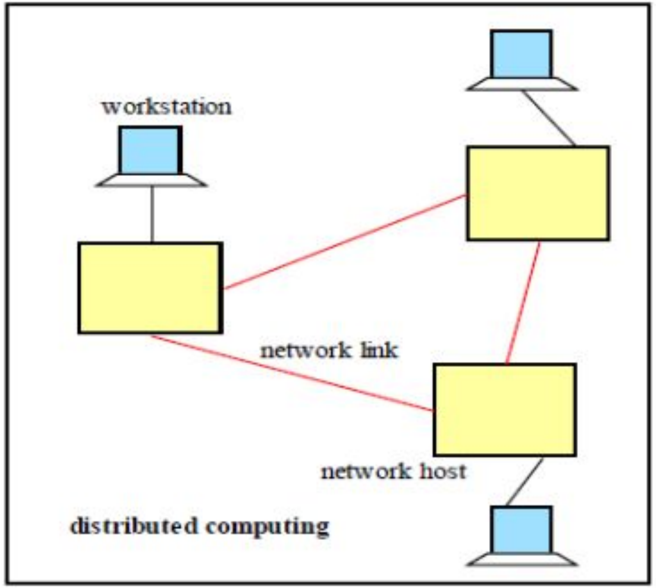
There is no one way to address these challenges. It is a design problem in itself.

1. Scalability requirements
2. Cost and Objective
3. Legacy design principles

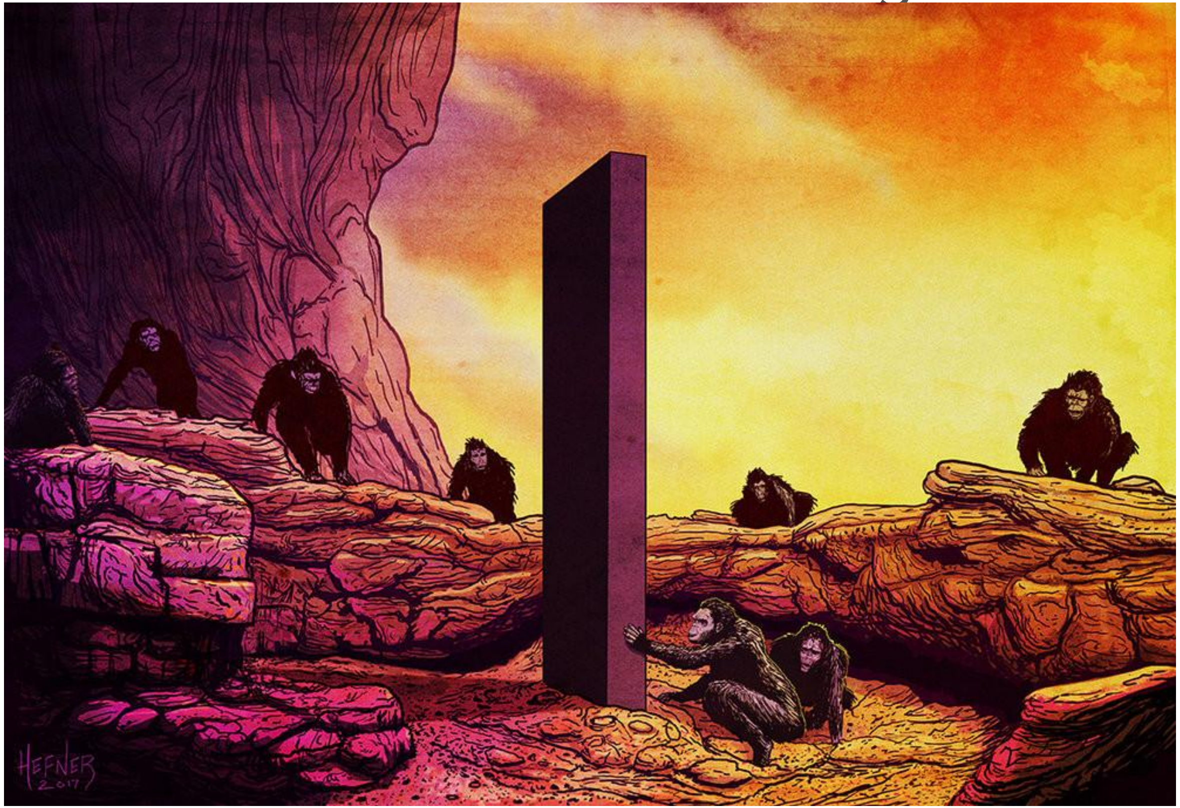
One way to scale is to distribute

What’s monolithic vs distributed computing?





Well so far we didn't find monoliths to be a good choice for AI!!!



Why not monolithic?

We need to process a lot of data

- Today in terms of data we are processing exabytes. In 2010 It was in petabytes.

- A single machine cannot serve all the data

- You need a distributed system to store and process in parallel

A single machine is liable to single point of failure

A single machine cannot scale

Why distribute?

Commodity hardware are cheap

Easy to scale as requirement

Distribution of computing and data storage can protect against single point of failure

But wait!! No free lunch right?

What about parallel programming?

Parallel programming?

- Threading is hard!
- How do you facilitate communication between nodes?
- How do you scale to more machines?
- How do you handle machine failures?

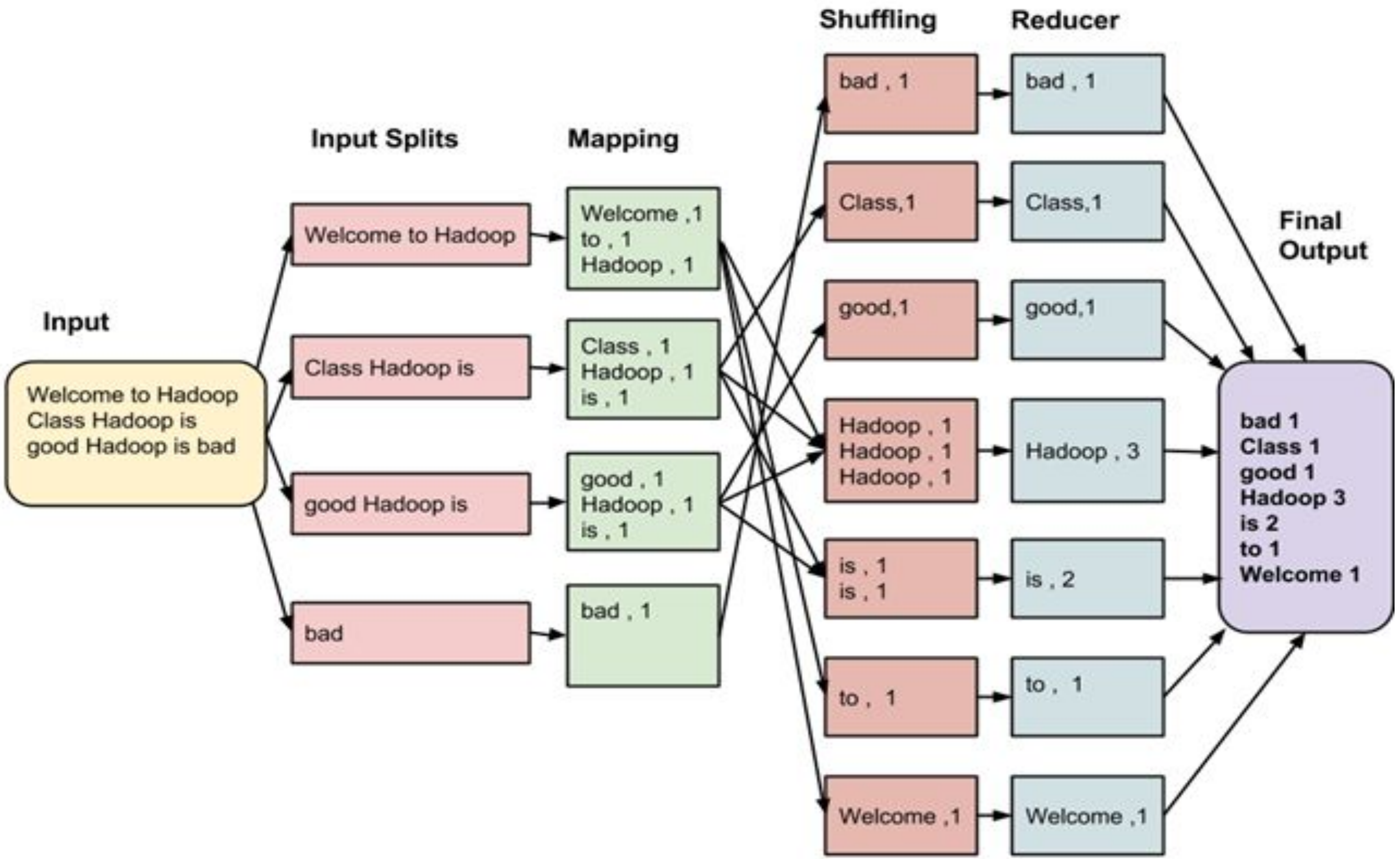
Distributed Word Count - MapReduce

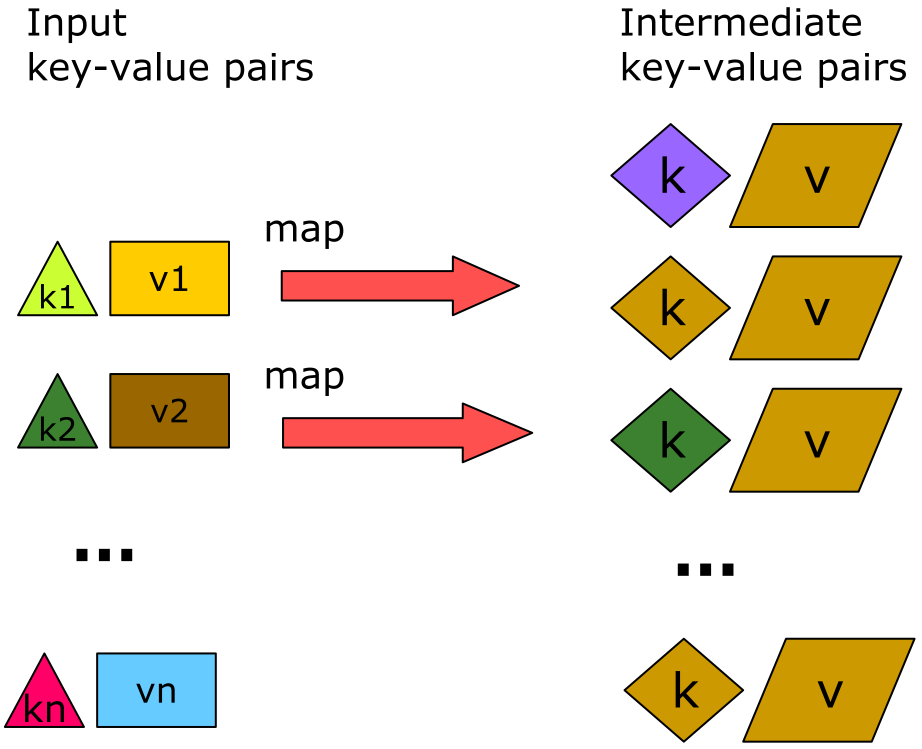
I HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO CLOUD COMPUTING.
BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD.
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLAH BLAH PLATFORM.
Dilbert.com DilbertCartoonist@gmail.com
© 2011 Scott Adams, Inc./Dist. by UFS, Inc.
1-7-11

Learn more about MapReduce here: https://www.youtube.com/watch?v=b-IvmXoO0bU

What is MapReuduce?

- **MapReduce [OSDI’04] provides**
  - Automatic parallelization, distribution
  - I/O scheduling
    - Load balancing
    - Network and data transfer optimization
  - Fault tolerance
    - Handling of machine failures
- **Need more power: Scale out, not up!**
    - Large number of commodity servers as opposed to some high end specialized servers


Input
Welcome to Hadoop Class Hadoop is good Hadoop is bad
Input Splits
Welcome to Hadoop
Class Hadoop is
good Hadoop is
bad
Mapping
Welcome ,1
to , 1
Hadoop , 1
Class , 1
Hadoop , 1
is , 1
good , 1
Hadoop , 1
is , 1
bad , 1
Shuffling
bad , 1
Class,1
good,1
Hadoop , 1
Hadoop , 1
Hadoop , 1
is , 1
is , 1
to , 1
Welcome ,1
Reducer
bad , 1
Class,1
good,1
Hadoop , 3
is , 2
to , 1
Welcome ,1
Final Output
bad 1
Class 1
good 1
Hadoop 3
is 2
to 1
Welcome 1

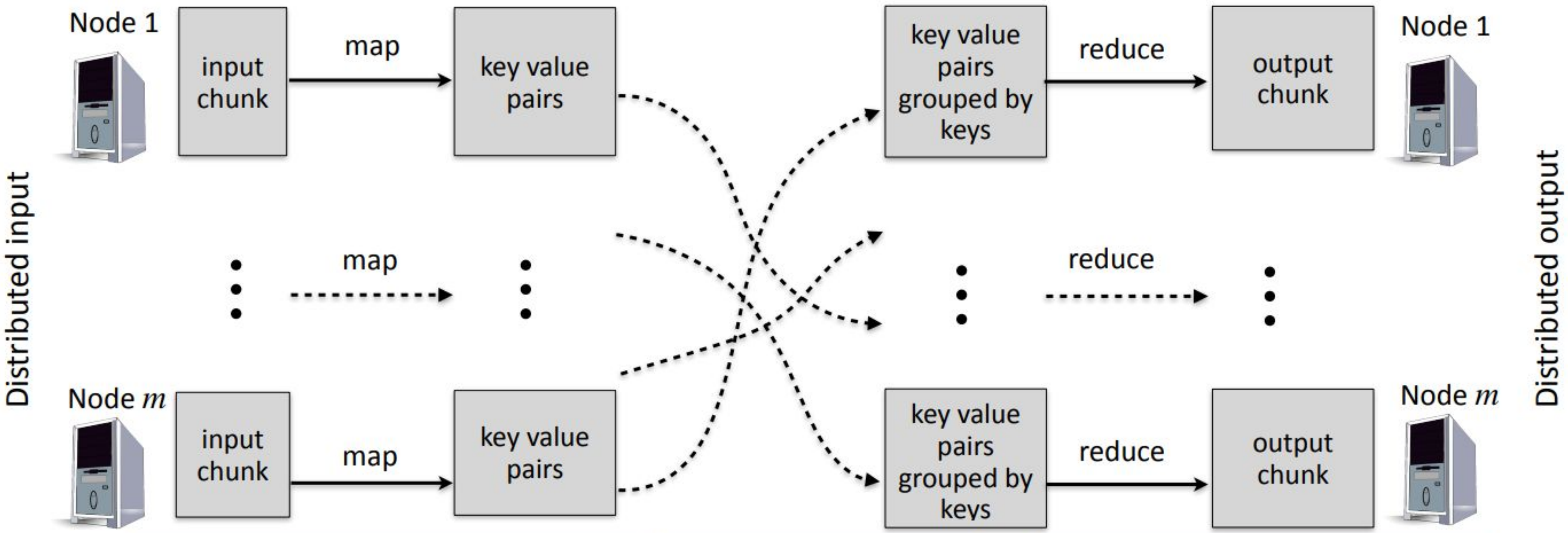


E.g. (doc—id,
doc-content)

E.g. (word,
wordcount-in-a-doc)

Adapted from Jeff Ullman's course slides

The MapReduce Paradigm



Map: proess every input segment, output appropriate key-value pair

Shuffle: group the key-value pairs by key

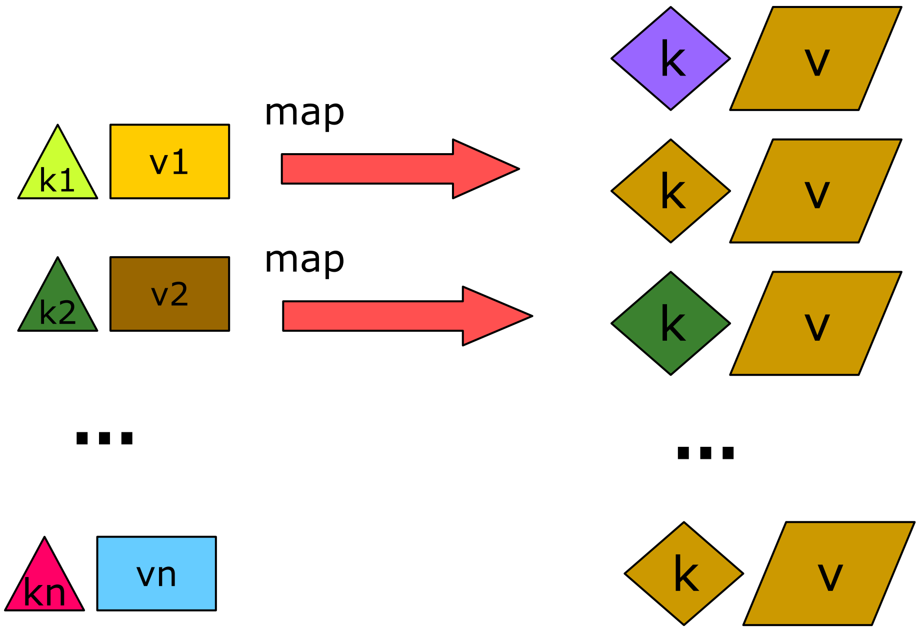
Reduce: for every key, perform some operation on the values

The user (programmer) needs to write the **map** and the **reduce** functions

MapReduce: The Map Step

Input key-value pairs

Intermediate key-value pairs



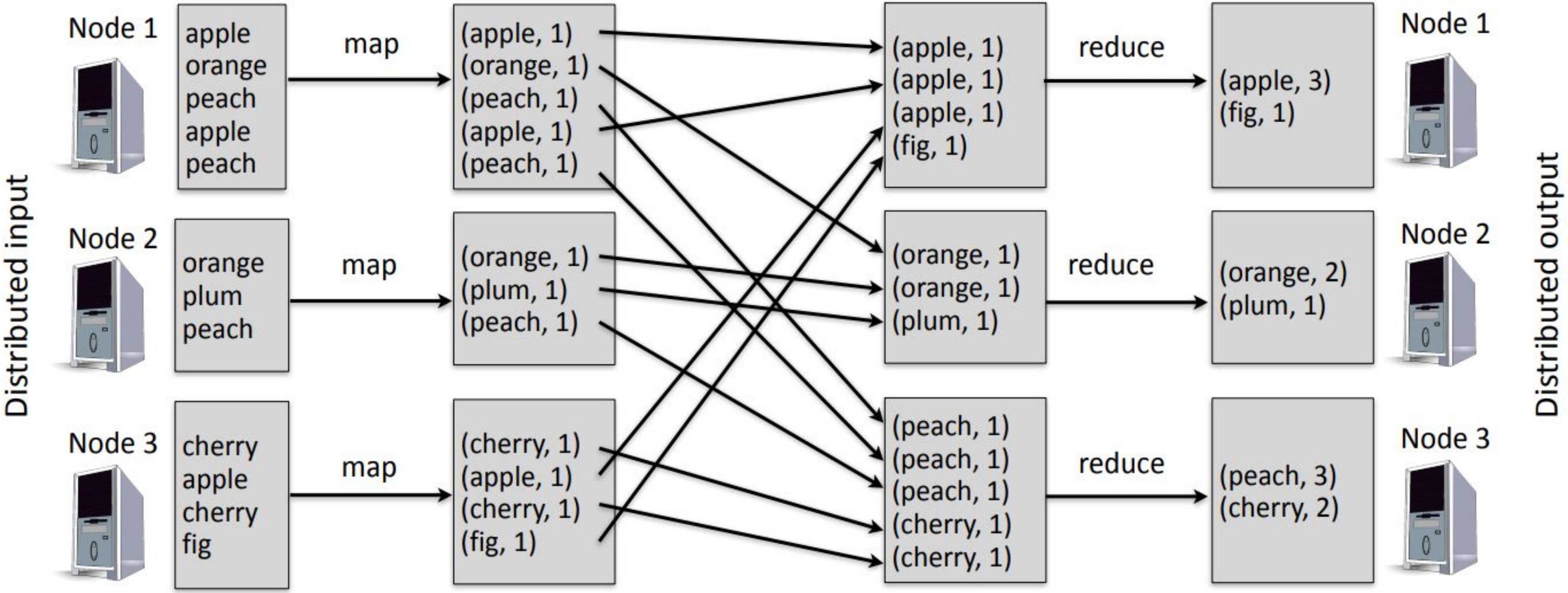
E.g. (doc—id, doc-content)

E.g. (word, wordcount-in-a-doc)

Adapted from Jeff Ullman’s course slides

Word frequency using MapReduce

Given a collection of documents, count the number of times each word occurs in the collection



Distributed input
Node 1
apple orange peach apple peach
map
(apple, 1) (orange, 1) (peach, 1) (apple, 1) (peach, 1)
Node 2
orange plum peach
map
(orange, 1) (plum, 1) (peach, 1)
Node 3
cherry apple cherry fig
map
(cherry, 1) (apple, 1) (cherry, 1) (fig, 1)
(apple, 1) (apple, 1) (apple, 1) (fig, 1)
reduce
(apple, 3) (fig, 1)
Node 1
(orange, 1) (orange, 1) (plum, 1)
reduce
(orange, 2) (plum, 1)
Node 2
(peach, 1) (peach, 1) (peach, 1) (cherry, 1) (cherry, 1)
reduce
(peach, 3) (cherry, 2)
Node 3
Distributed output

Map: for every word w, output the key-value pair $(w,1)$

Shuffle: group the key-value pairs by key

Reduce: for every key, sum the values

What questions do you have?

Apache Hadoop

- Apache Hadoop is an open-source Java platform that stores and processes large amounts of data.

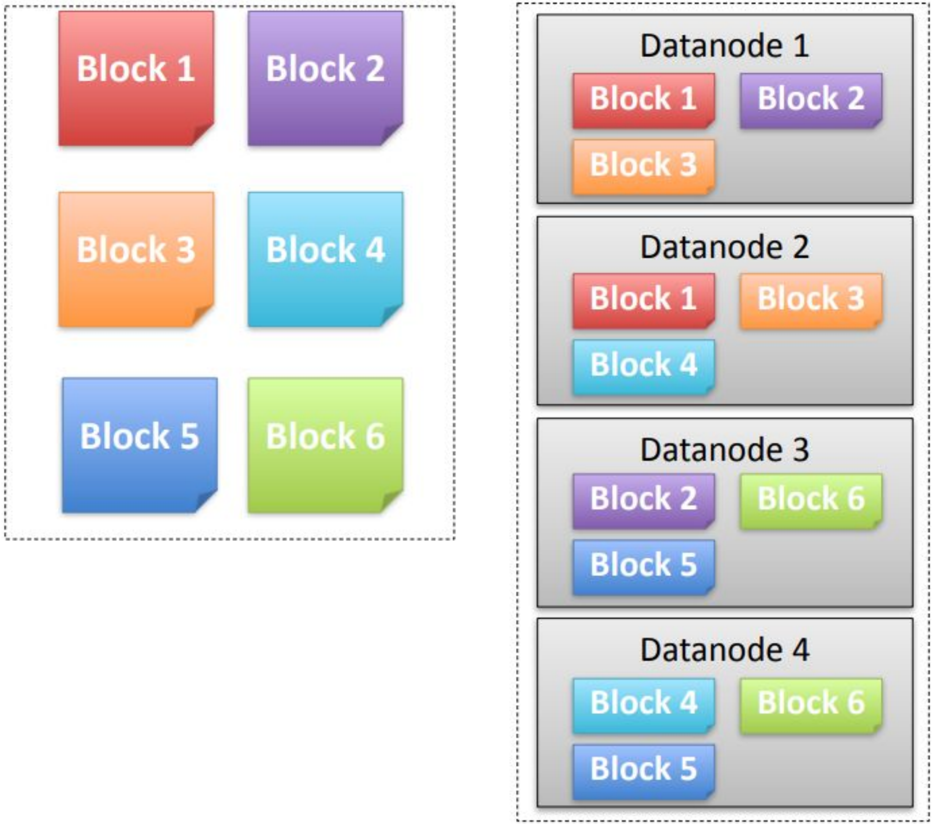
- Hadoop works by mapping large data sets (from terabytes to petabytes), analyzing tasks between clusters, and breaking them into smaller chunks (64MB to 128MB), resulting in faster data processing.

- To store and process data, data is sent to the Hadoop cluster, HDFS (Hadoop distributed file system) stores data, MapReduce processes data, and YARN (Yet another resource negotiator) divides tasks and assigns resources.

Summary

HDFS: Blocks

Big File



Block 1
Block 2
Block 3
Block 4
Block 5
Block 6

Datanode 1
Block 1
Block 2
Block 3

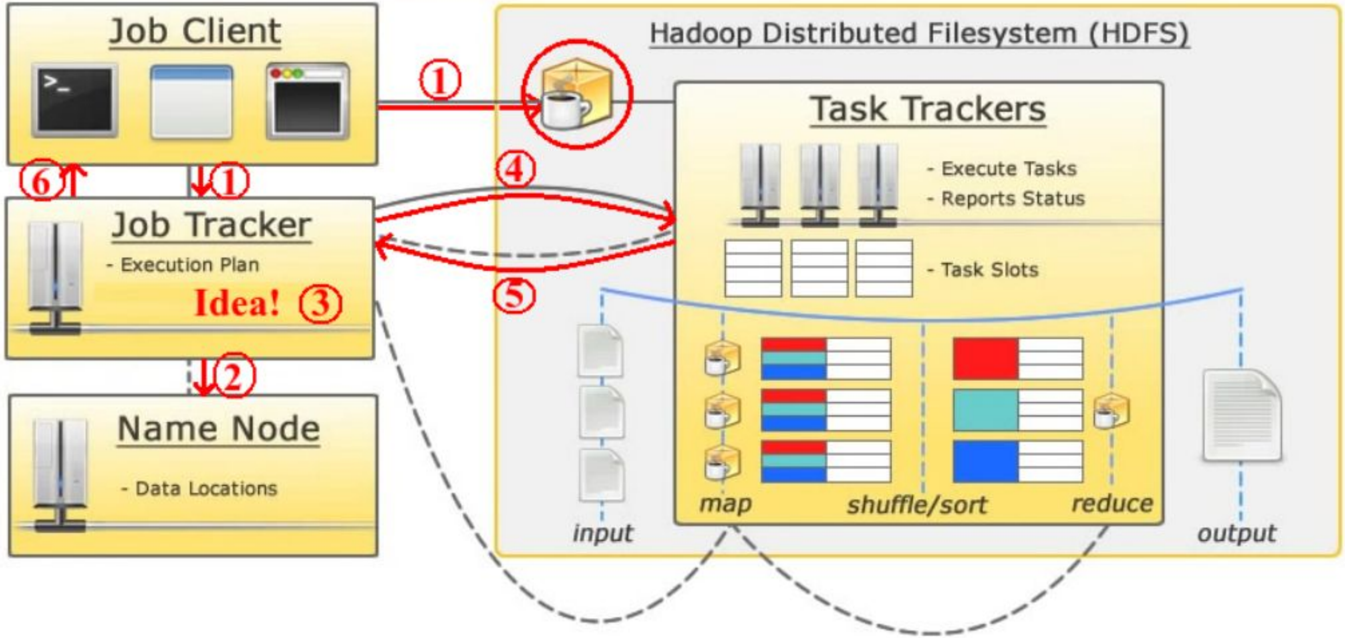
Datanode 2
Block 1
Block 3
Block 4

Datanode 3
Block 2
Block 6
Block 5

Datanode 4
Block 4
Block 6
Block 5

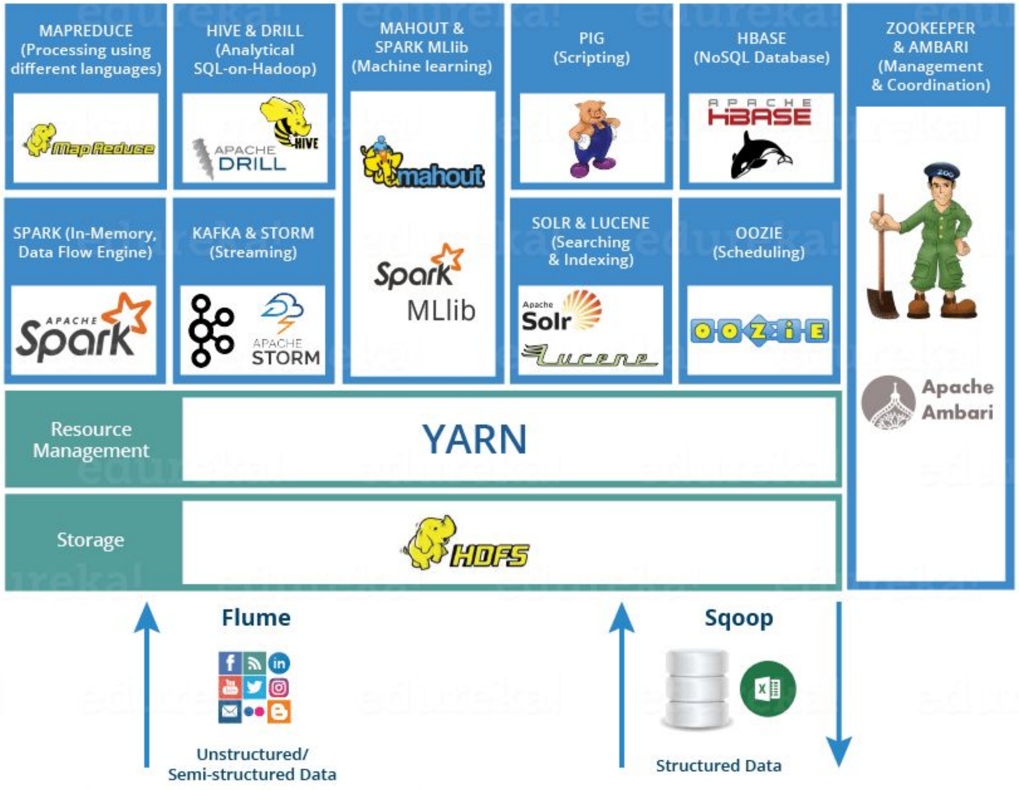
- Runs on top of existing filesystem
- Blocks are 64MB (128MB recommended)
- Single file can be > any single disk
- POSIX based permissions
- Fault tolerant

MapReduce: JobTracker and TaskTracker



1. JobClient submits job to JobTracker; Binary copied into HDFS
2. JobTracker talks to Namenode
3. JobTracker creates execution plan
4. JobTracker submits work to TaskTrackers
5. TaskTrackers report progress via heartbeat
6. JobTracker updates status

Hadoop Ecosystem


MAPREDUCE (Processing using different languages)
MapReduce
HIVE & DRILL (Analytical SQL-on-Hadoop)
HIVE
APACHE DRILL
MAHOUT & SPARK MLlib (Machine learning)
mahout
Spark MLlib
PIG (Scripting)
HBASE (NoSQL Database)
APACHE HBASE
ZOOKEEPER & AMBARI (Management & Coordination)
ZOO
Apache Ambari
SPARK (In-Memory, Data Flow Engine)
APACHE Spark
KAFKA & STORM (Streaming)
APACHE STORM
SOLR & LUCENE (Searching & Indexing)
Apache Solr
Lucene
OOZIE (Scheduling)
OOZIE
Resource Management
YARN
Storage
HDFS
Flume
Unstructured/ Semi-structured Data
Sqoop
Structured Data

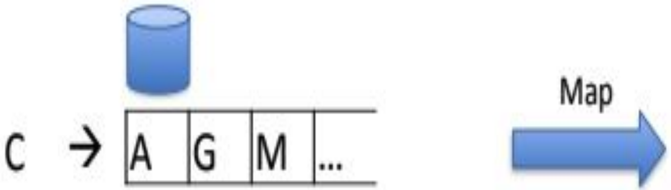
Apache Hive and Apache HBASE

- Hadoop/HDFS integration
- Apache Hive:
  - data warehousing on top of Hadoop
  - SQL features for Big Data
  - MapReduce jobs
- Apache HBASE:
  - NoSQL key/value store
  - real-time querying

Find #Mutual Friends in Social Media using Hadoop MapReduce

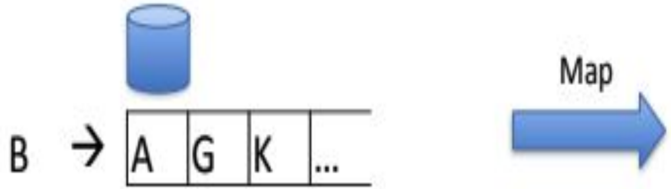
Map:

- For each member X, fetch his/her friendlist Y_1, Y_2, … , Y_n
  - For each pair (Y_1, Y_2), emit the key value pair {(Y_1, Y_2), X}



C → A | G | M | …

Key	Value
(A,G)	C
(A,M)	C
(G,M)	C
…	…

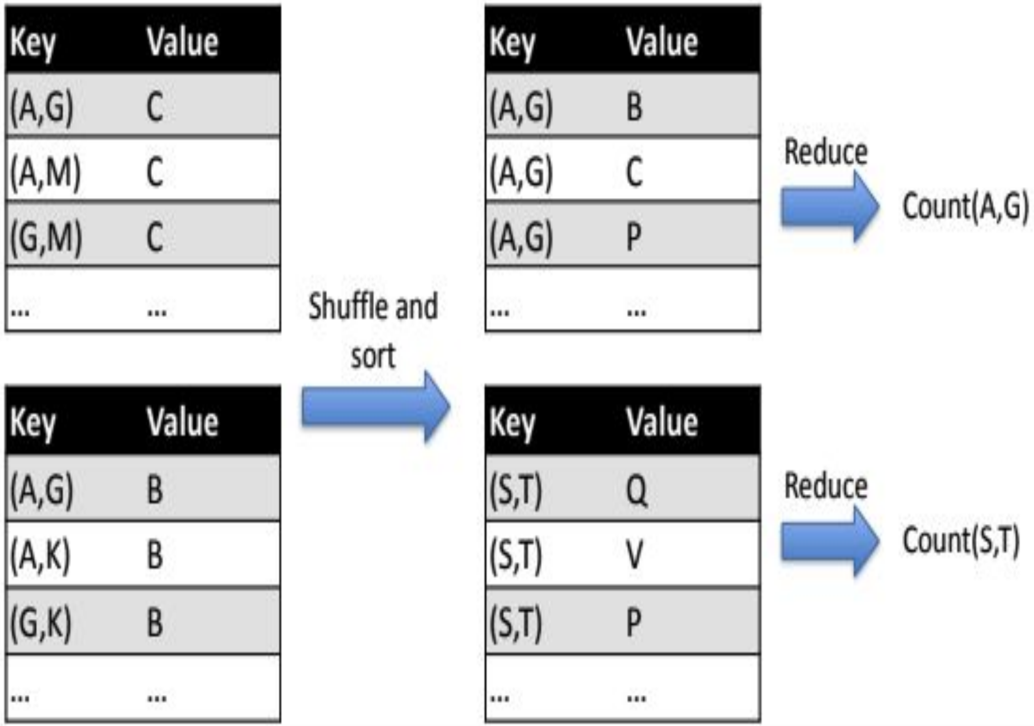


B → A | G | K | …

Key	Value
(A,G)	B
(A,K)	B
(G,K)	B
…	…

Reduce:

- For each Key = (X,Y), determine Count(X,Y) = # of mutual friends



Key	Value
(A,G)	C
(A,M)	C
(G,M)	C
…	…

Key	Value
(A,G)	B
(A,K)	B
(G,K)	B
…	…

Shuffle and sort

Key	Value
(A,G)	B
(A,G)	C
(A,G)	P
…	…

Key	Value
(S,T)	Q
(S,T)	V
(S,T)	P
…	…

Reduce

Count(A,G)

Reduce

Count(S,T)

What questions do you have?

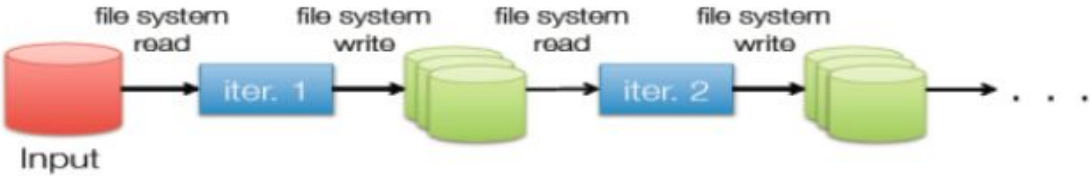
Big Data Tools: Apache Spark

- It is one of the largest open-source engines widely used by large companies. Spark is used by 80% of Fortune 500 companies, according to the website. It is compatible with single nodes and clusters for big data and ML.

- It is based on advanced SQL (Structured Query Language) to support large amounts of data and work with structured tables and unstructured data.

- The Spark platform is known for its ease of use, large community, and lightning speed. The Developers use Spark to build applications and run queries in Java, Scala, Python, R, and SQL.

Limitations of Vanilla MapReduce



- Map-reduce is fine for one-pass computation, but inefficient for multi-pass algorithms
  - Examples: k-means, PageRank
- No efficient mechanism for data sharing
  - State between steps goes to distributed file system
  - Slow due to replication & disk storage
- Not interactive or flexible
  Have to write *map* and *reduce* for any task
- Commonly spend 90% of time doing I/O

Apache Spark

- Goals
  - Extend the MapReduce model to better support two common classes of analytics apps:
    - Iterative algorithms (machine learning, graph)
    - Interactive data mining
  - Enhance programmability

- Approach: **Resilient Distributed Dataset (RDD)**
  - Allow apps to keep working sets in memory as long as possible
  - Retain the advantages of MapReduce (fault tolerance, data locality, scalability)
  - Support a wide range of applications

RDDs

- An RDD is a read-only , partitioned collection of records
- Can only be created by :
  - (1) Data in stable storage
  - (2) Other RDDs (transformation , lineage)
- Each RDD include:
  - 1) A set of partitions (atomic pieces of datasets)
  - 2) A set of dependencies on parent RDDs
  - 3) A function for computing the dataset based on its parents
  - 4) Metadata about its partitioning scheme
  - 5) Data placement
- An RDD has enough information about how it was derived from other datasets(its lineage)
  - Fault tolerance
- Users can control two aspects of RDDs
  - (1) Persistence (in RAM, reuse)
  - (2) Partitioning (hash, range, [<k, v>])
- Transformations are lazy, they don’t compute right away. Just remember the transformations applied to datasets(lineage). Only compute when an action require.

Case Study: Credit Card Fraud Detection

Case Study: Credit Card Fraud Detection

Suppose you are a Machine Learning engineer of a leading Bank.

1. Bank has millions of customer
2. Millions of credit card transactions happening every day - TB of data generated
3. Our objective is to build a simple fraud detection model

Case Study: Credit Card Fraud Detection

Suppose we already have a recipe for a good yet simple model

1. Fraud if user transaction location is different from last 30 user location
2. Amount is significantly higher than last 30 transaction amount

Case Study: Credit Card Fraud Detection

How is transaction data stored?

Usually today data are stored in form of logs.

What are logs?

Case Study: Credit Card Fraud Detection

As a Machine Learning engineer you need think about how to store and process the logs so that:

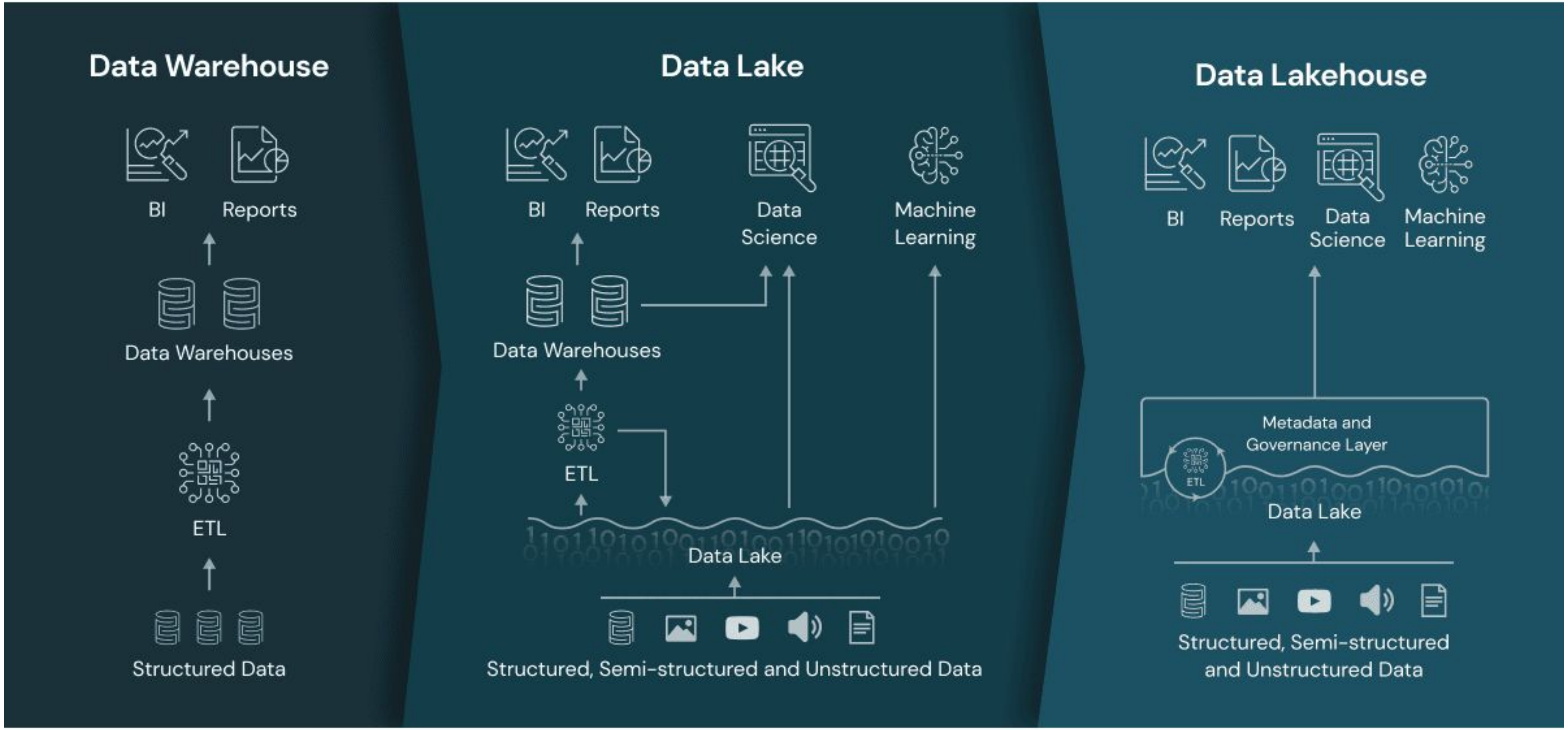
1. The storage doesn’t have single point failure
2. With increasing customer base the storage can scale
3. Faster processing of the data to compute statistics

Case Study: Credit Card Fraud Detection

Design choices to consider:

1. Storage should be able to scale and have fault tolerance
2. Data processing techniques/Algorithm that can scale to huge volume of data
3. Platform for workflow design, analytics and dashboarding

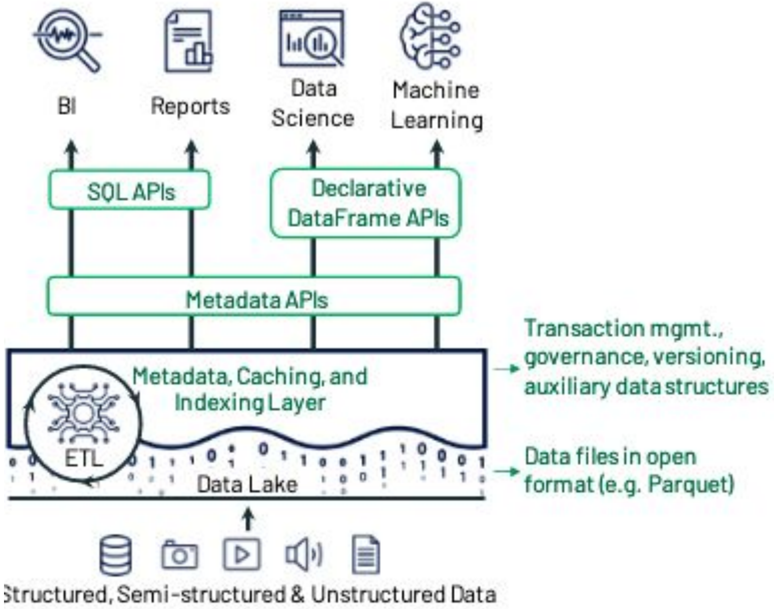
Data Lake vs Data Lakehouses vs Data Warehouses



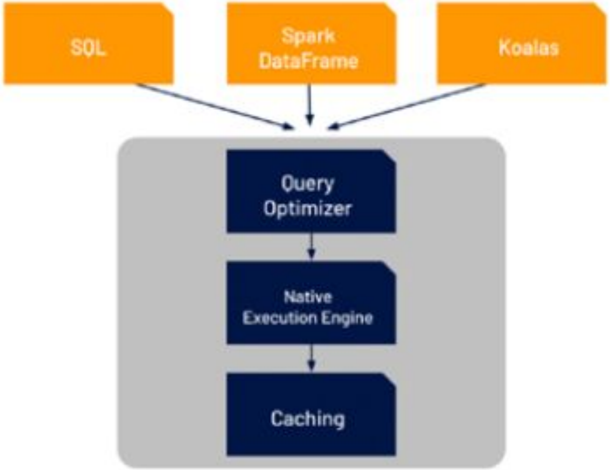
Img Source

Delta Lake

- unify ETL, data warehousing and ML


BI
Reports
Data Science
Machine Learning
SQL APIs
Declarative DataFrame APIs
Metadata APIs
Metadata, Caching, and Indexing Layer
ETL
Data Lake
Transaction mgmt., governance, versioning, auxiliary data structures
Data files in open format (e.g. Parquet)
Structured, Semi-structured & Unstructured Data

- Delta Engine



Img Source

Img Source

Popular Big Data Tools

High level tools summary

Data Storage and Management

mongoDB

cassandra

neo4j

APACHE HBASE

talend

APACHE hadoop

HDInsight Microsoft

Apache ZooKeeper™

Data cleaning

OPEN Refine

High level tools summary

Data mining

TERADATA

rapidminer

Data reporting

Power BI

Data Acquisition and Ingestion

sqoop

FLUME

APACHE STORM™

Data visualization

tableau

IBM Watson Analytics

IBM.

plotly

High level tools summary

Data analysis

HIVE
hadoop
MapReduce
APACHE
Spark™

Feedback on Lecture and Concepts?

Capstone Projects

Get ready for Thursday

- Make sure to install all requirements
- Run Imports and make sure you can import all packages
  - Having any issues? Reach out to us to resolve it before Thursday.

No classes next week (Happy Thanksgiving!

See you in two days!