Introduction to Big Data

Mahmoud Parsian

Ph.D. in Computer Science

What is Big Data?

Large

datasets, complex and changeable

Cannot be stored, managed, and processed adequately by traditional software management and software tools.

Solutions? Cluster Computing

Relying on the Technology that handle , process and analyze large quantities of data:

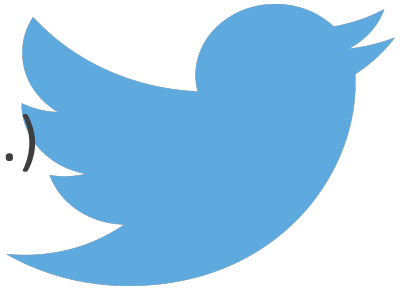
- MapReduce
- Hadoop
- Spark
- Snowflake
- Amazon Athena
- Google BigQuery

Solutions? Types of Data

- Technology that **handle** , **process** and **analyze** large quantities of data:
- Types of Data:
  - Structured
    - XML
    - CSV
    - Parquet
  - Semi-Structured
    - JSON
    - XML
  - Unstructured
    - Text files, log files

The need of Big Data…

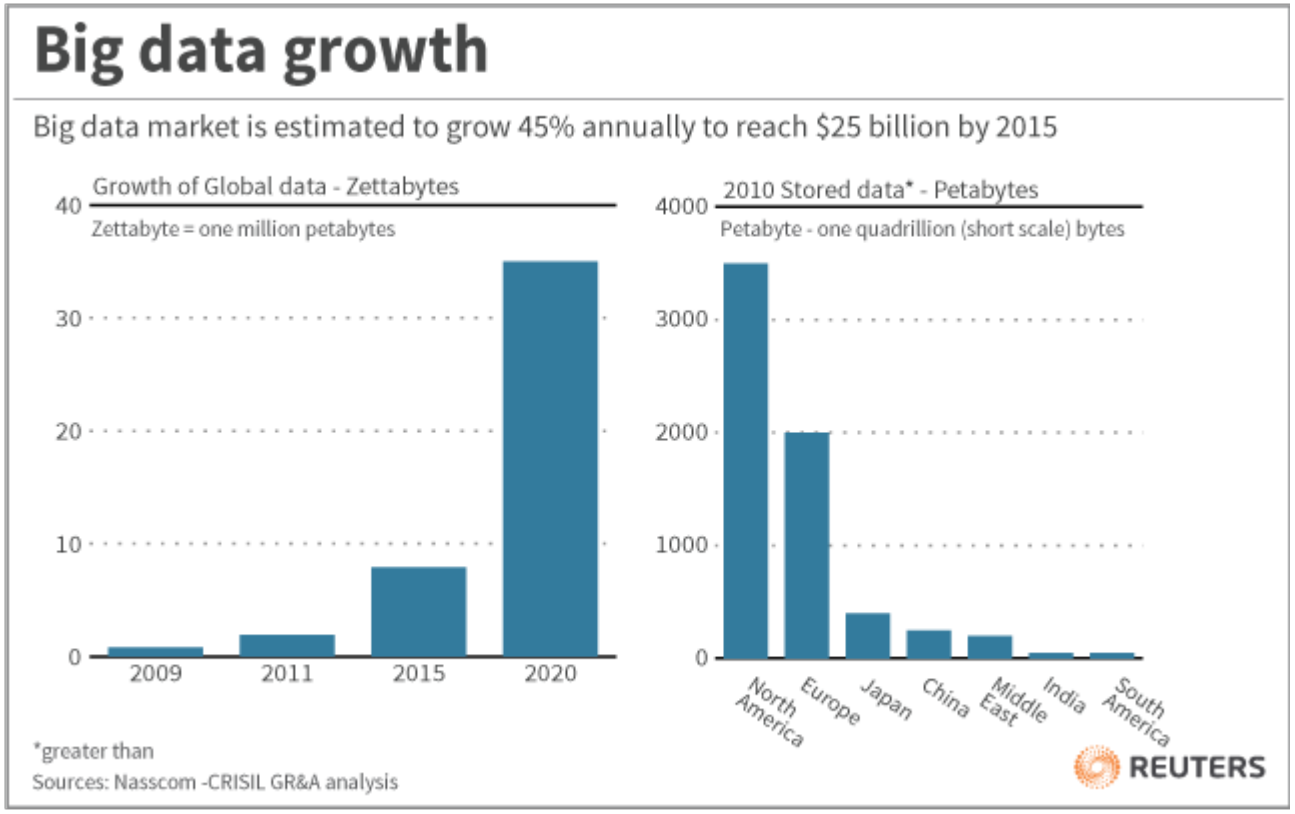
- Data growth:
  - Social networking (Twitter, Facebook, ..)
  - Media (Netflix, Cable TV, …)
  - Commerce sites (Amazon, Ebay, …)
  - **Health and genomics**, data growth increasing exponentially and so on for the future.
- Analyzing and managing these data properly is the key to business expansion and growth.



Example of Big Data?

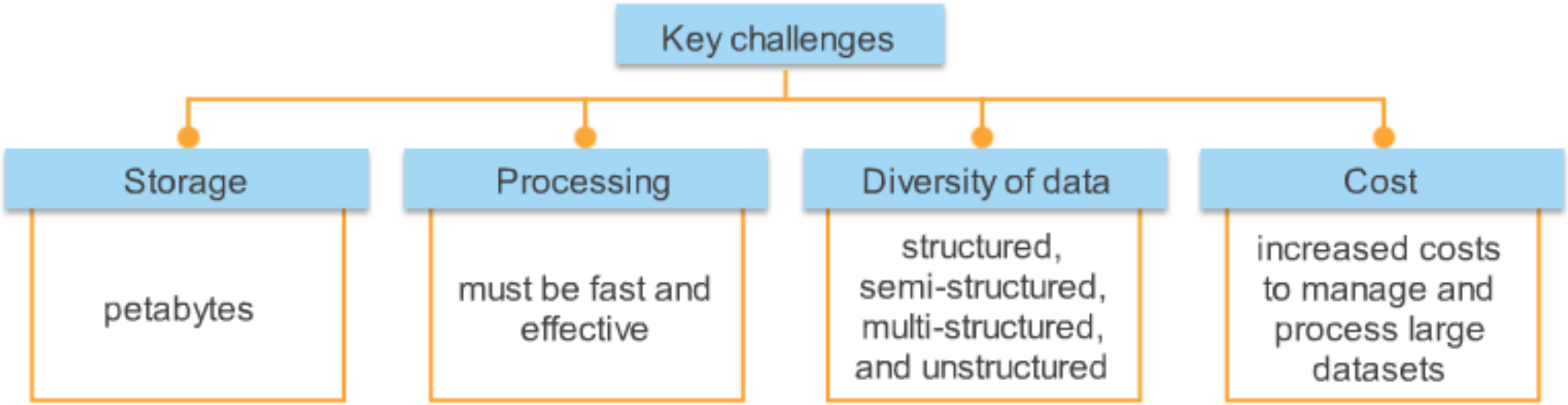
- Credit Card Transactions
- Billions of documents indexed for search engine
- Logs generated by web servers
- DNA & Genomics data
- Twitter feeds
- Facebook Messages
- Medical records
- COVID data

Data Creation



- Every Year, the world created
- more and more Zeta bytes of data
- Managing this data became crucial to extract more value in retail, finance, media and publishing.

Big Data Challenges



Google MapReduce Paper Revolutionized Big Data (2004)

MapReduce: Simplified Data Processing on Large Clusters

by Jeffrey Dean and Sanjay Ghemawat

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a *map* and a *reduce* function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct MapReduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.

Google MapReduce Paper → Hadoop

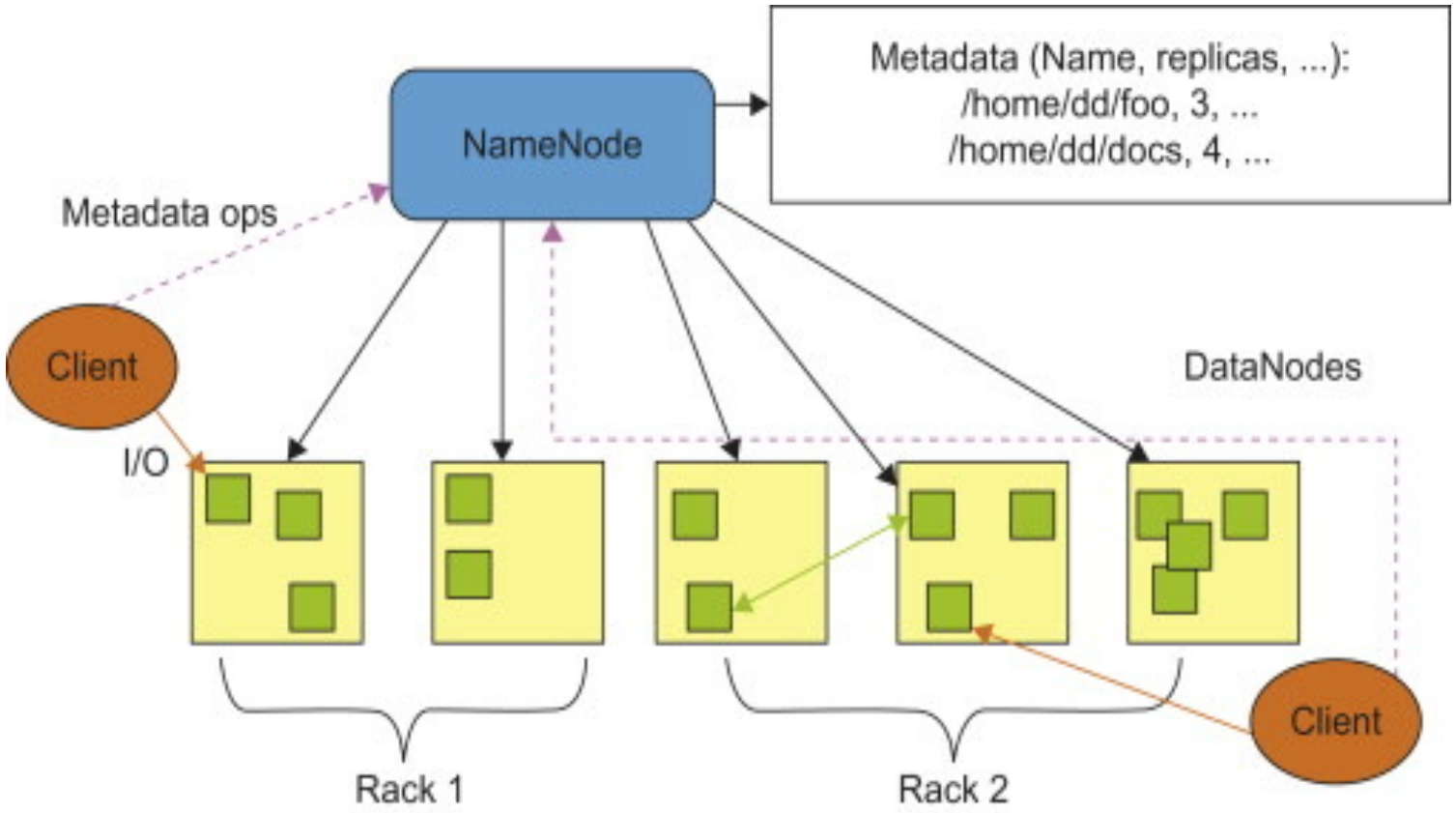
- Google implemented MapReduce (2004)
  - index billions of documents for its search engine
- Google did NOT release any MapReduce code to public
- Hadoop was implemented based on Google's MapReduce paper
- Hadoop became an open-source Apache project (2006)

- Hadoop came as the answer of software framework that would support technologies, tools and infrastructure and also more effective and efficient management and analysis of large datasets.
- Hadoop tools and services working together in completing its tasks, forming a Hadoop ecosystem.

Key features and capabilities

- Storage and processing in Big Clusters (1000 nodes)
- Parallel processing - MapReduce framework
- High availability - failover mechanism
- Data Distribution
- Scalability
- Cost effectiveness

Hadoop Architecture



Problems with Hadoop

- Complex: write lots of code
- Low level API
- Uses disk I/O extensively
- Did not use memory/RAM (RAM is 100 times faster than Disk I/O)
- Only supported Text files
- Not Multi-language support (Python, Java, Scala, SQL)
- Hadoop Problems → Apache Spark
  - Spark addresses Hadoop Problems

Apache Spark

- Spark is a unified engine for large-scale data analytics
- Spark is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.
- Use memory/RAM first, Then Disk
- Parallel Processing (superset of MapReduce)
- Data Partitioning
- Scalability
- Multi-language support (Python, Java, Scala, SQL)
- Data Abstractions: RDDs and DataFrames
- Enables SQL access to Big Data

Apache Spark Example

Solving Classic Word Count Problem:

```
1. # word_frequencies : {(word1, freq1), (word2, freq2), …}
2. input_path = “s3://my_bucket/project23/”
3. word_frequencies =
4.     spark.sparkContext.textFile(input_path)\
5.     .flatMap(lambda x: x.split(' ')) \
6.     .map(lambda x: (x, 1)) \
7.     .reduceByKey(lambda a,b : a+b)
```

Ecosystem Component

Other components in the ecosystem

- Primary component:
- HDFS – Distributed file system for storage
- MapReduce –Parallel processing framework
- Spark: superset of MapReduce

Typical Hadoop eco-system project might use

- ZooKeeper –Coordination framework
- Hive – SQL-like inferface for querying
- Pig – High level scripting language for batch processing and ETL
- Hbase – Tabular/column storage (non-relational, distribution database)
- Flume – Distribution, reliable, service for collecting,aggregating,
and moving large data

Hadoop vs. Spark

- Hadoop: very hard to write MapReduce programs
  - Long programs (complex, low-level API)
- Spark: very simple to write distributed programs and beyond
  - Short programs (simple, powerful, high-level API)
- Hadoop: Java support
- Spark: Multi-language support (Python, Java, Scala, SQL)
- Hadoop: slow, uses disk I/O
- Spark: fast, uses RAM as much as possible
- Hadoop: Parallel Processing
- Spark: Parallel Processing

Spark is (new technology, fast, de facto standard) superior to Hadoop (old technology, slow)

Big Data's Characteristic

Volume

Variety

Velocity

Complexity

Validity

Stored Data Processing

- Batch-based stored
- Real-Time Data-stream processing

Batch Based Stored Data Processing

- Process large volumes of data
- Can be periodic or one-time processing
- Batch results are produced after data is collected, entered and processed
- Separate techniques or programs for input, processing and output

Real Time Data Processing

- Data captured, processed, and acted 24/7
- Computing data real-time

- Advantages of Real-time:
  - System scales elastically based on need
  - Instant Results
  - Parallel processing is available

Tools and Techniques for analyzing big Data

The choice of tools mostly driven by:

Who is going to use the data

+

the business requirement for a particular scenario

Where to Store Big Data

While traditional RDBMSs has many limitations, Big Data provide alternatives address some of these limitations.

- HDFS (Hadoop Distributed File System)
- Amazon S3
- Database alternatives:
  - Document Stores – Apache CouchDB, MongoDB
  - Graph stores –Neo4j
  - Key-Value Stores –Apache Cassandra, Riak
  - Tabular stores –Apache Hbase

Accessing the Big Data

- SQL like connectivity initiates for big data:
  - Amazon Athena (JDBC), Snowflake (JDBC), Impala, Hive and Stinger
  - SQL access to Spark DataFrames (table of named columns)

- These big data later can be used in:
  - Sandboxes for data science projects
  - Analytics development using MapReduce
  - Analytics-query performance enablers
  - Search indexes on multi structured data in Hadoop
  - Analyzing multi structured Big data Using search
  - Data visualization and in memory data in big data environment

Considering Using big Data

- Does your problem require a Big Data Solution?
- Does the solution need to handle data variety?
- Does the solution require ability to deal with high data velocity?
- Does the solution handle high data volume?
- Can the solution handle complexity?
- How can you optimize the solution?

Case Study: Social Media Analytics

Using people's history on internet, what they buy, what they search giving a rough view of attitude on a product.

More, these output can be used to study:
customer satisfaction, churn prediction, financial performance, stock performance.