BITS Pilani
Pilani Campus

Lectuer-3
Big Data Systems(SEZG522)

Slides: Courtesy:.Prof. Anindya

BITS Pilani
Pilani Campus

First Semester

2022-23

Lecture -3 Contents

Analytics

- –Definitions
- –Maturity model
- –Types

Big Data Analytics

- –Characterization
- –Adoption challenges
- –Requirements
- –Technology challenges
- –Popular technologies - Hadoop, Spark, Cloud, …

Will help you to pitch a Big Data project proposal

- –Case study - Nasdaq

Characteristics of Big Data Systems

- –Failures - Reliability and Availability
- –Consistency

Will help you to build the system

Analytics - Definitions

Extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact based management to drive decisions and actions

Purpose

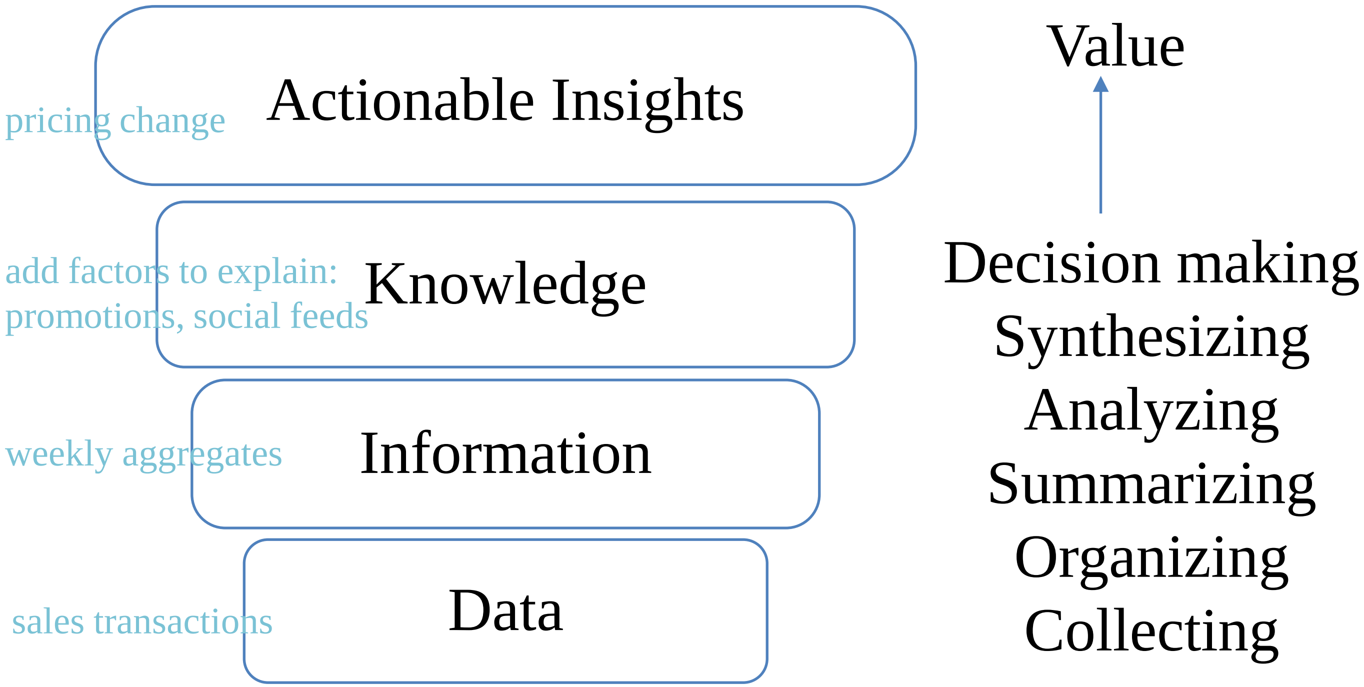
- ✓ To unearth hidden patterns
  - ✓ 2 / 100 stores had no sales for a promotion item because it was not in the right shelf
- ✓ To decipher unknown correlations
  - ✓ The famous “Beer and diapers” story
- ✓ Understand the rationale behind trends
  - ✓ What do users like about a popular product that has growing sales
- ✓ Mine useful business information
  - ✓ Popular items during specific holiday sales

Helps in

- ✓ Effective marketing
- ✓ Better customer service and satisfaction
- ✓ Improved operational efficiency
- ✓ Competitive advantage over rivals

Process of Analysis

Transformation of Data



pricing change — Actionable Insights

add factors to explain: promotions, social feeds — Knowledge

weekly aggregates — Information

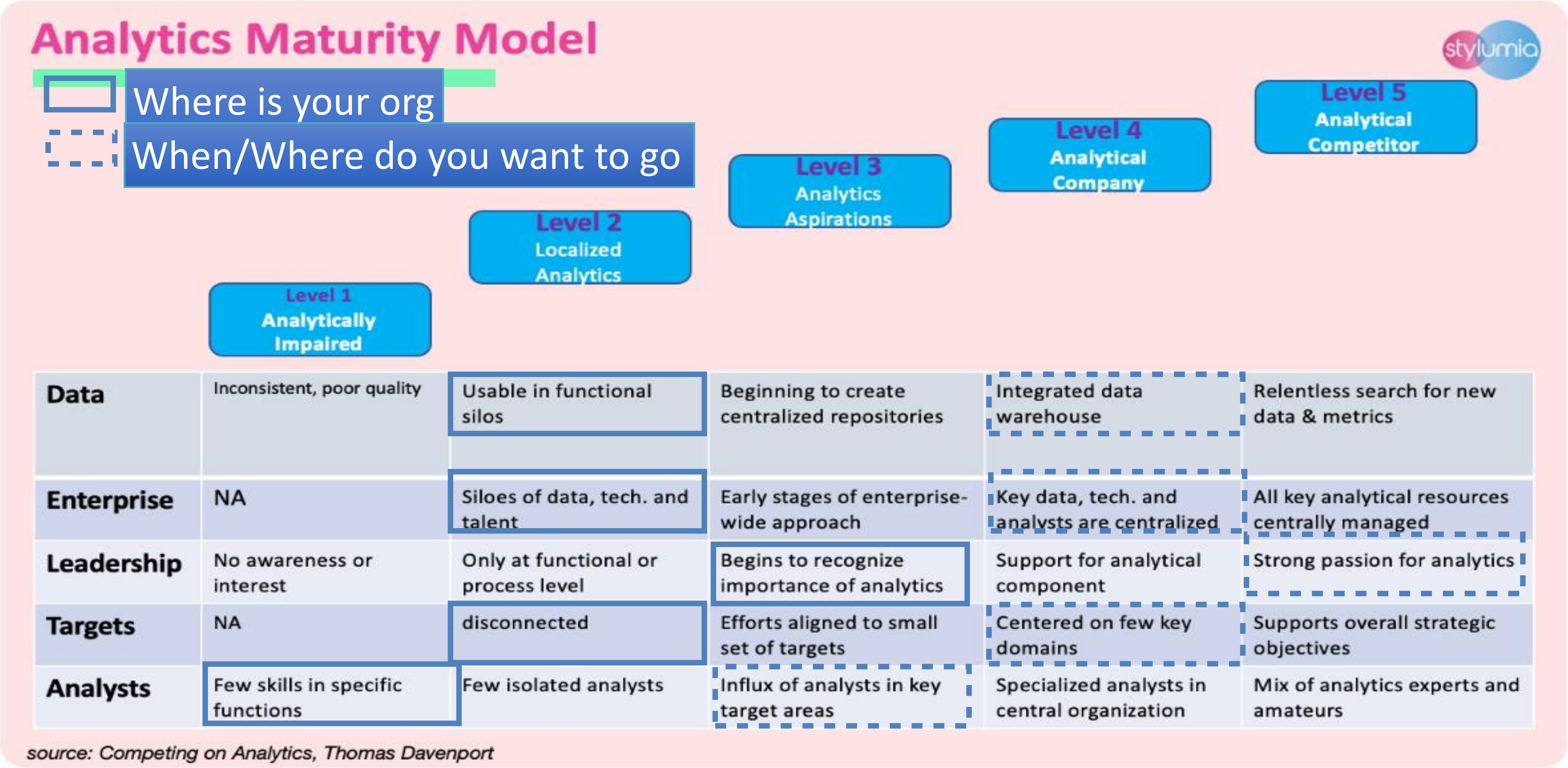
sales transactions — Data

Value

Decision making
Synthesizing
Analyzing
Summarizing
Organizing
Collecting

- Apply functions / transformations on data to get to the next level till it is actionable insight useful for the business
- Keep attaching more meta-data for context and make it meaningful

Analytics Maturity Model



Analytics Maturity Model

stylumia

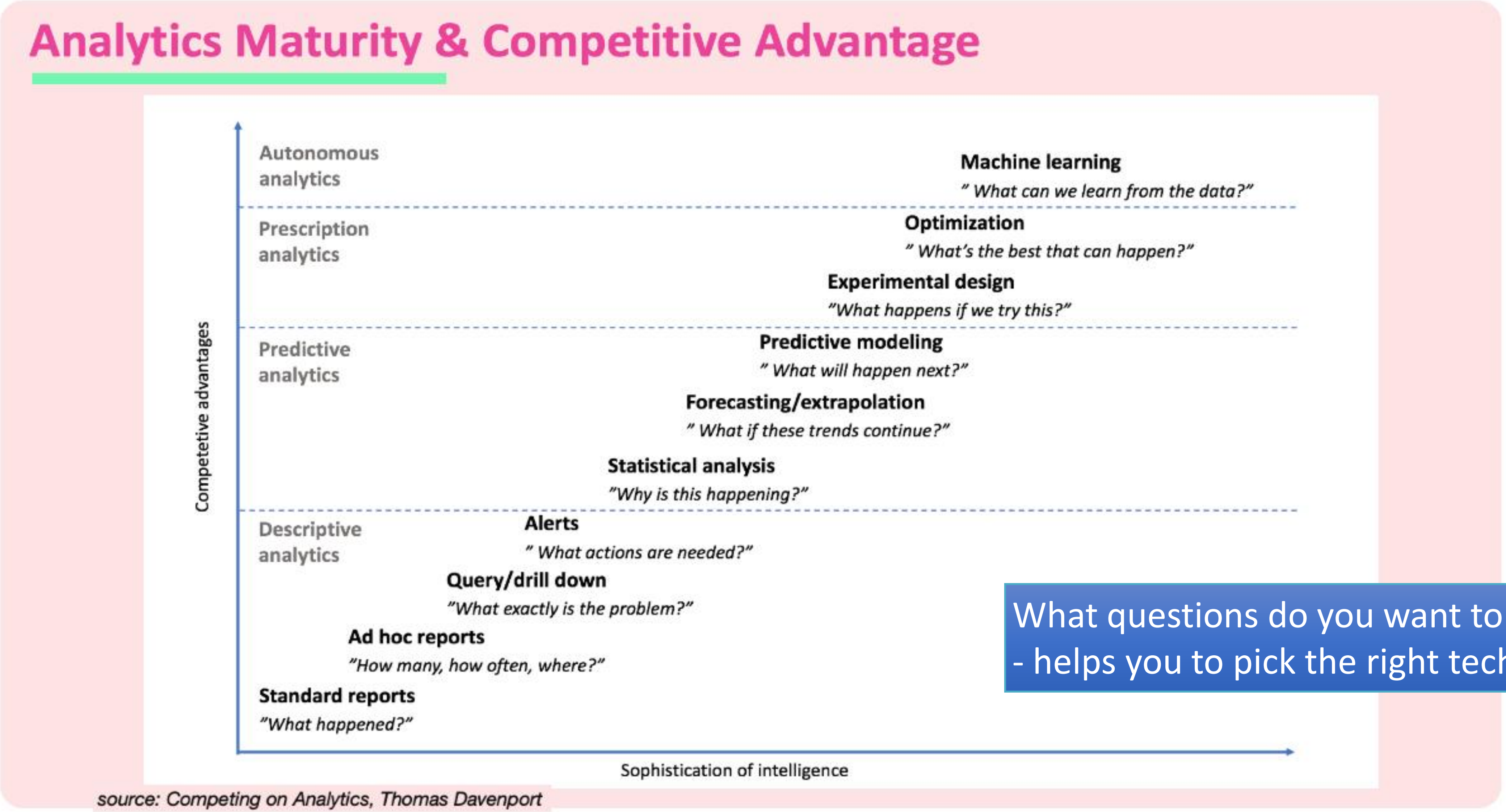
□ Where is your org

⬚ When/Where do you want to go

	Level 1 Analytically Impaired	Level 2 Localized Analytics	Level 3 Analytics Aspirations	Level 4 Analytical Company	Level 5 Analytical Competitor
Data	Inconsistent, poor quality	Usable in functional silos	Beginning to create centralized repositories	Integrated data warehouse	Relentless search for new data & metrics
Enterprise	NA	Siloes of data, tech. and talent	Early stages of enterprise-wide approach	Key data, tech. and analysts are centralized	All key analytical resources centrally managed
Leadership	No awareness or interest	Only at functional or process level	Begins to recognize importance of analytics	Support for analytical component	Strong passion for analytics
Targets	NA	disconnected	Efforts aligned to small set of targets	Centered on few key domains	Supports overall strategic objectives
Analysts	Few skills in specific functions	Few isolated analysts	Influx of analysts in key target areas	Specialized analysts in central organization	Mix of analytics experts and amateurs

source: Competing on Analytics, Thomas Davenport

Analytics Maturity and Competitive Advantage



Analytics Maturity & Competitive Advantage

Competetive advantages

Autonomous analytics

Machine learning
" What can we learn from the data?"

Prescription analytics

Optimization
" What's the best that can happen?"

Experimental design
"What happens if we try this?"

Predictive analytics

Predictive modeling
" What will happen next?"

Forecasting/extrapolation
" What if these trends continue?"

Statistical analysis
"Why is this happening?"

Descriptive analytics

Alerts
" What actions are needed?"

Query/drill down
"What exactly is the problem?"

Ad hoc reports
"How many, how often, where?"

Standard reports
"What happened?"

Sophistication of intelligence

source: Competing on Analytics, Thomas Davenport

What questions do you want to answer - helps you to pick the right technology

Types of Analytics - Descriptive

- Provides ability to alert, explore and report using mostly internal and external data
- Business Intelligence (BI) or Performance reporting
- Provides access to historical and current data
- Reports on events, occurrences of the past
- Usually data from legacy systems, ERP, CRM used for analysis
- Based on relational databases and warehouse
- Structured and not very large data sets
- Sometimes also referred as Analytics 1.0
- Era : mid 1950s to 2009

- Questions asked
  - ✓ What happened?
  - ✓ Why did it happen? (Diagnostic analysis)

E.g. number of infections is significantly higher this month than last month and highly correlated with factor X

Types of Analytics - Predictive

- Uses past data to predict the future
- Uses quantitative techniques like segmentation, forecasting etc. but also makes use of descriptive analytics for data exploration
- Uses technologies like models and rule based systems
- Based on large data set gathered over period of time
- Externally sourced data also used
- Unstructured data may be included
- Hadoop clusters, SQL on Hadoop data etc. technologies used
- aka Analytics 2.0
- Era : from 2005 to 2012

- Key questions
  - ✓What will happen?
  - ✓Why it will happen?
  - ✓When will it happen?

E.g. number of infections will reach X in month Y and likely cause will be Z

Types of Analytics - Prescriptive

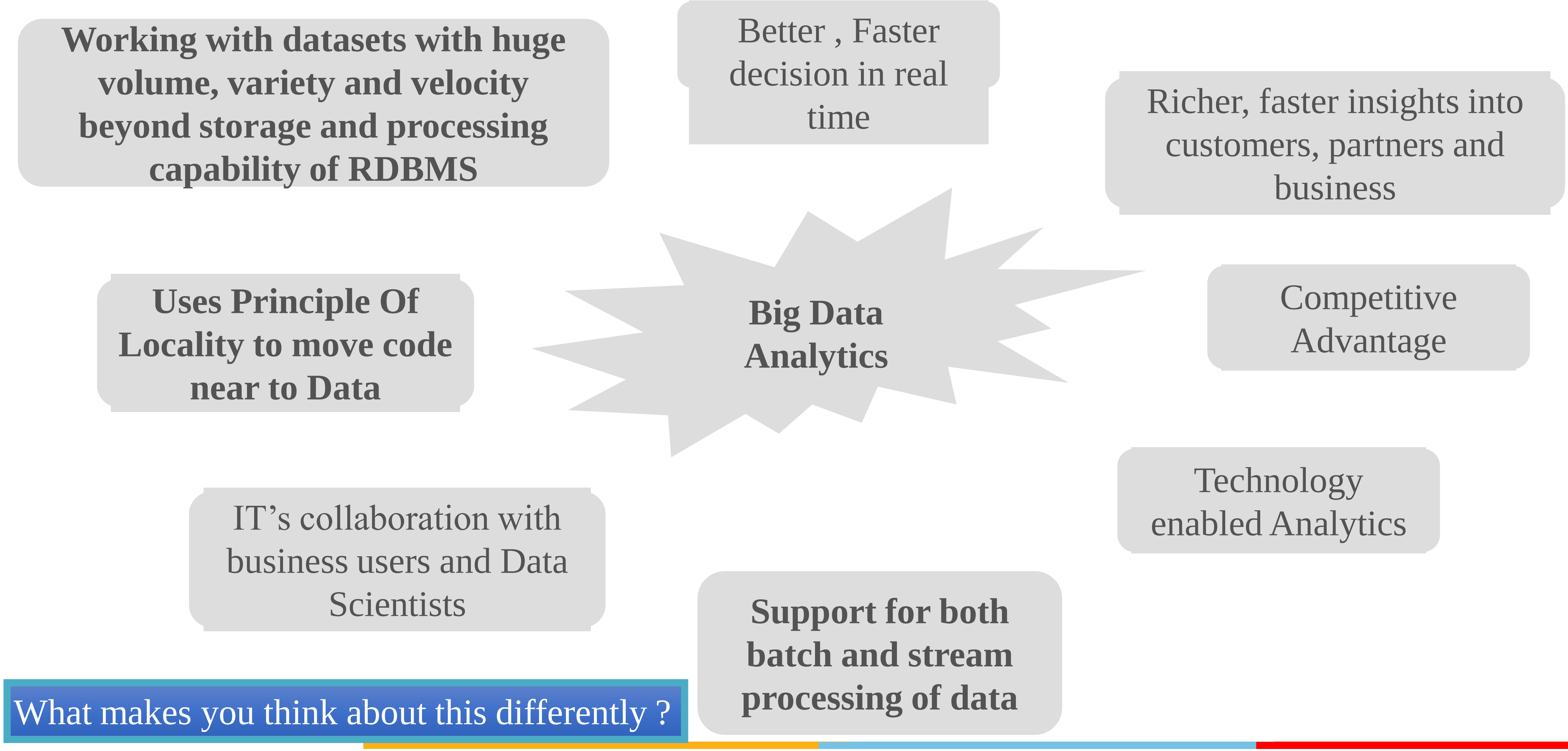
- Uses data from past to make prophecies of future and at the same time make recommendations to leverage the situation to one’s advantage
- Suggests optimal behaviors and actions
- Uses a variety of quantitative techniques like optimization and technologies like models, machine learning and recommendations engines
- Data is blend from Big data and legacy systems, ERP, CRM etc.
- In-memory analysis etc.
- Aka Analytics 3.0 = Descriptive + Predictive + Prescriptive
- post 2012
- Questions:
  - ✓ What will happen
  - ✓ When will it happen
  - ✓ Why will it happen
  - ✓ What actions should be taken

E.g. number of infections will reach X in month Y and likely cause will be Z. W is the best recommended action to keep the number in month Y below X/2.

Alternative Categorization

- Basic Analytics
  - ✓Slicing and dicing of data to help with basic insights
  - ✓Reporting on historical data, basic visualizations etc.
- Operationalized Analytics
  - ✓Analytics integrated in business processes
- Advanced Analytics
  - ✓Forecasting the future by predictive modelling
- Monetized Analytics
  - ✓Used for direct business revenue

Big Data Analytics



Big Data Analytics

Working with datasets with huge volume, variety and velocity beyond storage and processing capability of RDBMS

Better , Faster decision in real time

Richer, faster insights into customers, partners and business

Uses Principle Of Locality to move code near to Data

Competitive Advantage

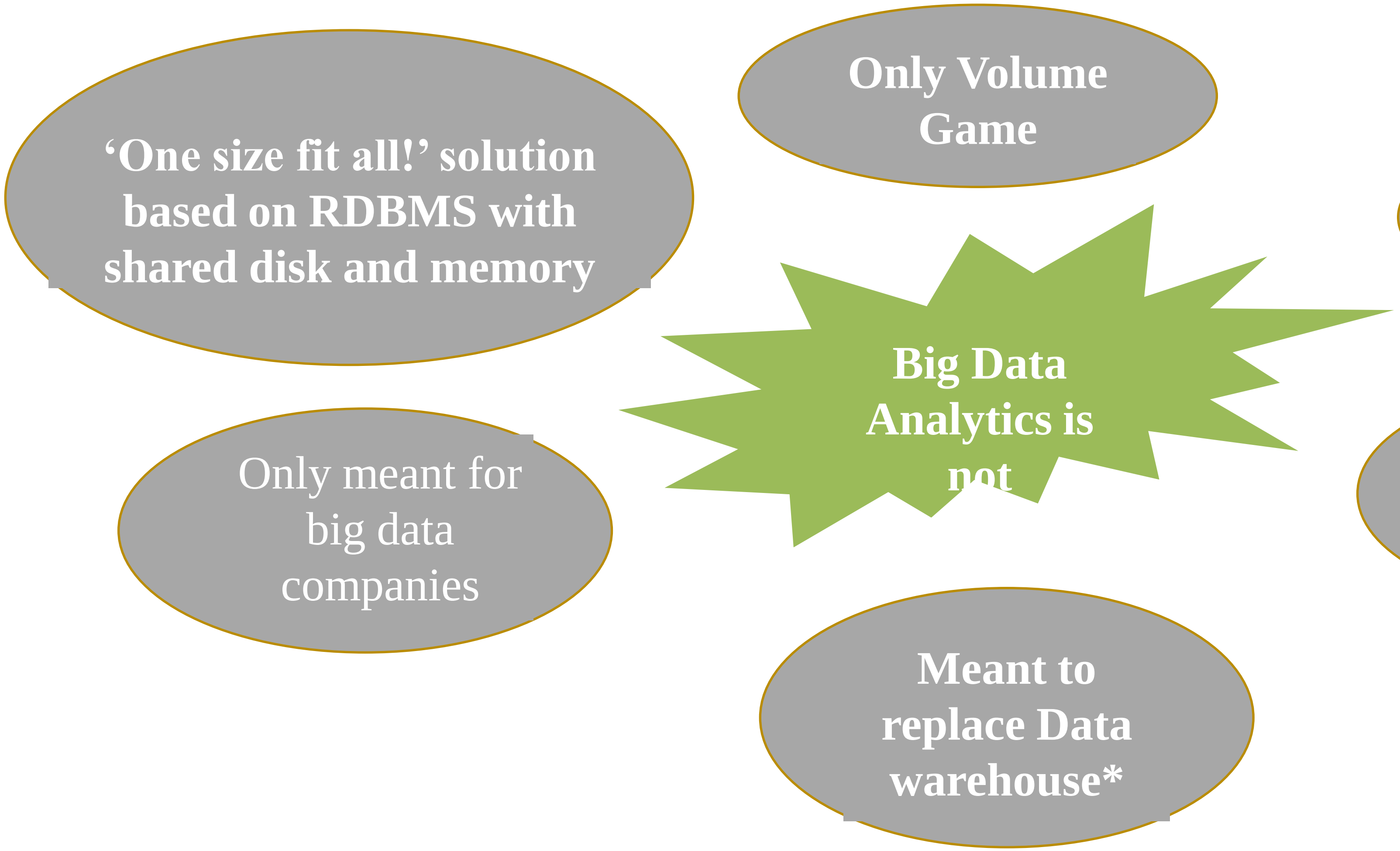
IT’s collaboration with business users and Data Scientists

Technology enabled Analytics

Support for both batch and stream processing of data

What makes you think about this differently ?

What Big Data Analytics is not



Big Data Analytics is not

‘One size fit all!’ solution based on RDBMS with shared disk and memory

Only Volume Game

Just bothered about Technology

Only meant for big data companies

Meant to replace RDBMS

Meant to replace Data warehouse*

Things to know to avoid friction in Big Data projects

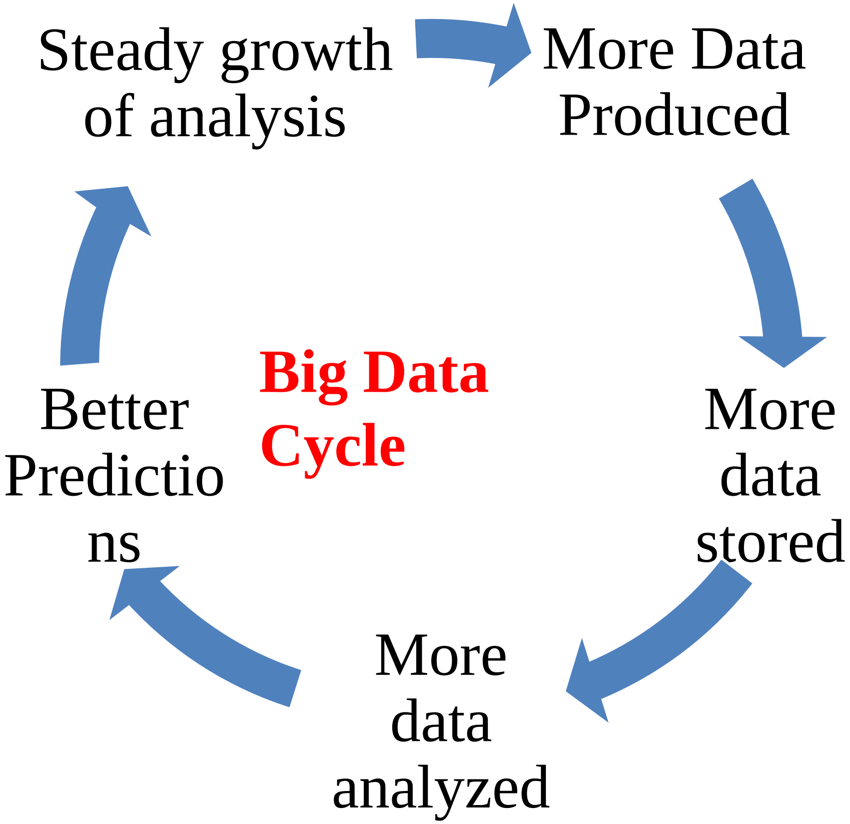
Why the sudden hype ?

Data is growing at 40% compound annual rate

- ✓ 45 ZB in 2020
- ✓ In 2010, 1.2 trillion Gigabytes data generated
- ✓ In 2012, reached to 2.4 trillion Gigabytes
- ✓ Volume of world wide data expected to double every 1.2 years
- ✓ Every day 2.5 quintillion bytes of data is created
- ✓ 90% of today’s data is generated in last few years only!
- ✓ Walmart processes one million customer transaction per hour
- ✓ 500 million “tweets” are posted by users every day
- ✓ 2.7 billion “likes” and comments by Facebooks users per day

Cost of storage has hugely dropped

Large number of user friendly analytics tools available for data processing



Steady growth of analysis

More Data Produced

More data stored

More data analyzed

Better Predictions

Big Data Cycle

Adoption Challenges in Organizations

- Obtaining executive sponsorship for investments in big data and its related activities
- Getting business units to share data / information across organizational silos
- Finding right skills (Business Analysts/Data Scientists and Data Engineers) that can manage large amount of variety of data and create insights from it
- Determining approach to scale rapidly, address storage and processing of large volume, velocity and variety of Big data
- Deciding whether to use structured or unstructured, internal or external data to make business decisions
- Choosing optimal way to report findings and analysis of big data
- Determining what to do with the insights created from big data

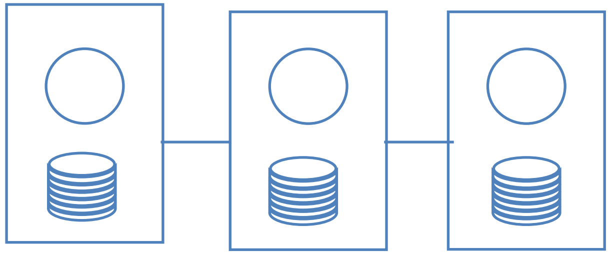
Requirements of Big Data analytics

- Cheap abundant storage
- Processing options
  - batch / streaming,
  - disk based / memory based
- Open source platforms, e.g. Hadoop, Spark
- Parallel and distributed systems with high throughput rather than low latency
- Cloud or other flexible resource allocation arrangements
  - Flexibility to setup and tear down infrastructure for quick projects across various teams

Technology Challenges

Scale

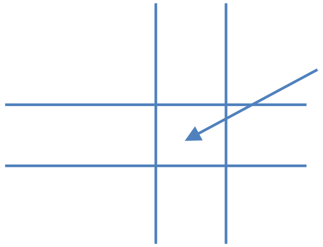
- ✓ Need is to have storage that can best withstand large volume, velocity and variety of data
- ✓ Scale vertically / horizontally ?
- ✓ RDBMS / NoSQL ?
- ✓ How does compute scale with storage - coupled or de-coupled, i.e. good idea to put common nodes for compute and storage (refer Nasdaq case)



compute + data on same node: locality helps, but does compute scale with storage ?

Security

- ✓ Most of recent NoSQL big data platforms have poor security mechanisms, e.g. challenges:
  - ✓ Fine grain control in semi-structured data, esp with columnar storage
  - ✓ Options for inconsistent data complicate matters
  - ✓ Larger attack surface across distributed nodes
  - ✓ Often encryption is turned off for performance
- ✓ Lack of authorization techniques while safeguarding big data
  - ✓ May contain PII data (personally identifiable info)



Easier in RDBMS to control at row / column / cell level with always consistent values in a tightly coupled system

+ Ref: NoSQL security

Contd..

Schema

- ✓ Need is to have dynamic schema, static / fixed schemas don’t fit

Continuous availability

- ✓ Needs 24 * 7 * 365 support as data is continuously getting generated and needs to be processed
- ✓ Almost all RDBMS, NoSQL big data platforms has some sort of downtime
  - ✓ Memory cleanup, replica rebalancing, indexing, …
  - ✓ Most of the large scale NoSQL systems also need weekly maintenance

Contd..

Consistency

- ✓ Should one go for strict consistency or eventual consistency? Is this like social media comments or application needs consistent reads ?

Partition Tolerant

- ✓ When a system get’s partitioned by hardware / software failures. How to build partition tolerant systems ? When faults happen is consistent data available ?
- ✓ We will discuss options in CAP Theorem

Data quality

- ✓ How to maintain data quality – data accuracy, completeness, timeliness etc.?
- ✓ Do we have appropriate metadata in place esp with semi/un-structured data ?

Popular Technologies

How to manage voluminous, varied, scattered and high velocity data ?

- ✓ Think beyond an RDBMS depending on use case but not necessarily to replace it

Some popular technologies

- ✓ Distributed and parallel processing (covered in session 2)
- ✓ Hadoop (more details in session 6-9)
  - ✓ File based large scale parallel data processing tasks
- ✓ In-memory computing (more details in session 13-16)
  - ✓ Usage of main memory (RAM) helps to manage data processing tasks faster
- ✓ Big Data Cloud (more details in session 11)
  - ✓ Helps to save cost and better management of resources using a services model

What problems does Hadoop solve

Storage of huge amount of data

- ✓ Problems
  - ✓ Multiple partitions of data for parallel access but more systems means more failures
  - ✓ Multiple nodes can make the system expensive
  - ✓ Arbitrary data - binary, structured …
- ✓ Solution
  - ✓ Replication Factor (RF) for failures : Number of data copies of a given data item / data block stored across the network
  - ✓ Uses commodity heterogenous hardware
  - ✓ Multiple file formats

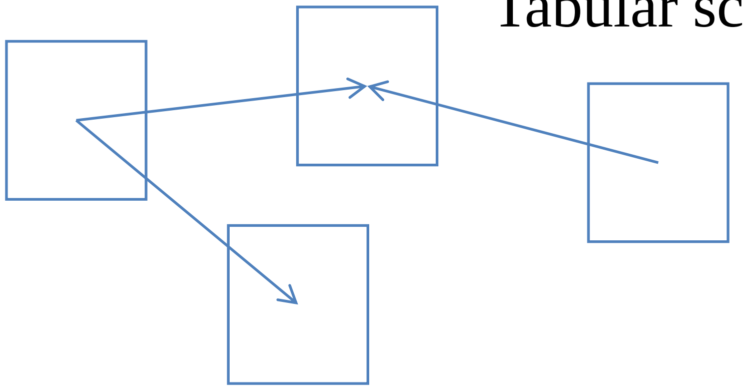
Processing the huge amount of data

- ✓ Problems
  - ✓ Data is spread across systems, how to process it in quick manner?
  - ✓ Challenge is to integrate data from different machines before processing
- ✓ Solution
  - ✓ MapReduce programming model to process huge amount of data with high throughput
  - ✓ Compute is close to storage for handling large data sets

What’s different from a Distributed Database

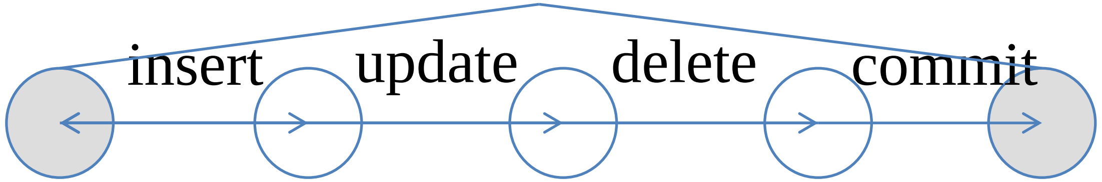
Distributed Databases

- **Data model**
  - Tables and relations
  - Schema is predefined (during write)
  - Supports partitioning
  - Fast indexed reads
  - Read and write many times
- **Compute model**
  - Generate notations of a transaction
  - ACID properties
  - Allow distributed transactions
  - OLTP workloads



Tabular schema

consistent states of a transaction

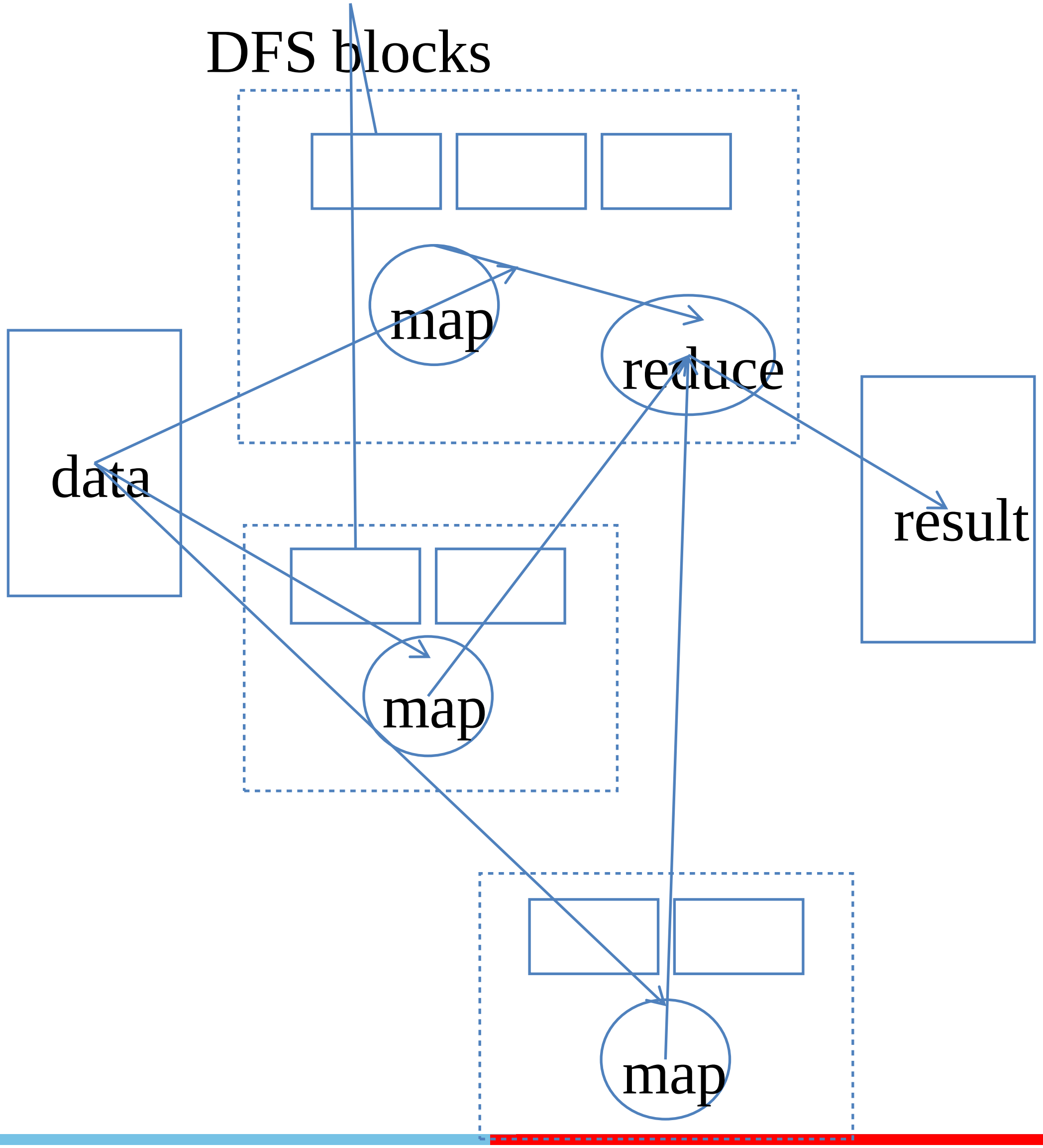


insert update delete commit

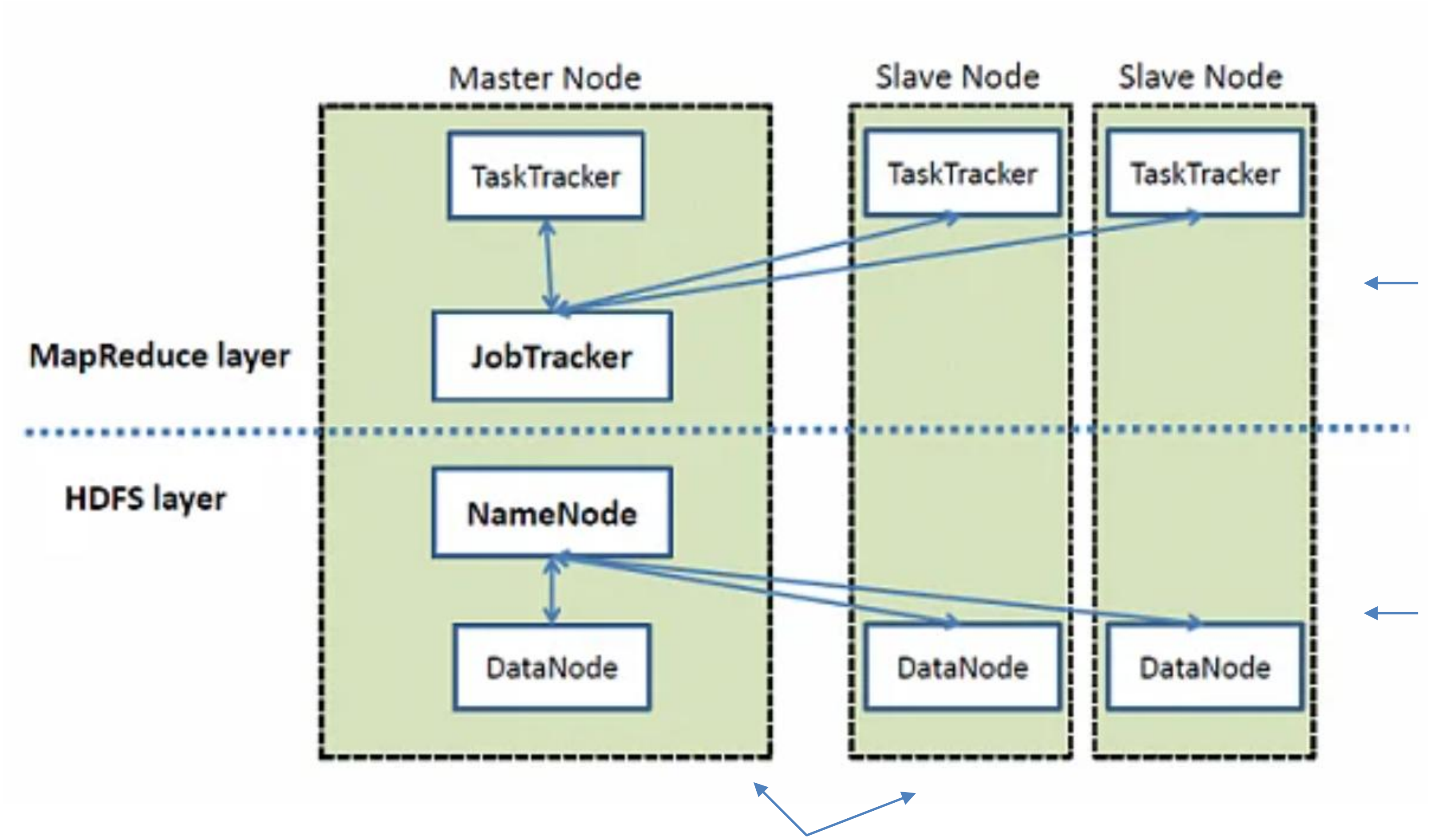
rollback on errors

Hadoop in Contrast …

- **Data model**
  - Flat files supporting multiple formats, including binary
  - No pre-defined schema (i.e. during write)
  - Divides files automatically into blocks
  - Handles large files
  - Optimized for write
  - Write once and read many times workload
  - Meant for scan workloads with high throughput
- **Compute model**
  - Generate notations of a job divided into tasks
  - MapReduce compute model
  - Every task is a map or a reduce
  - High latency analytics, data discovery workloads



Hadoop High Level Architecture



Tasks consisting of Map and Reduce jobs run on the DataNodes.
They are managed by JobTracker and TaskTrackers.

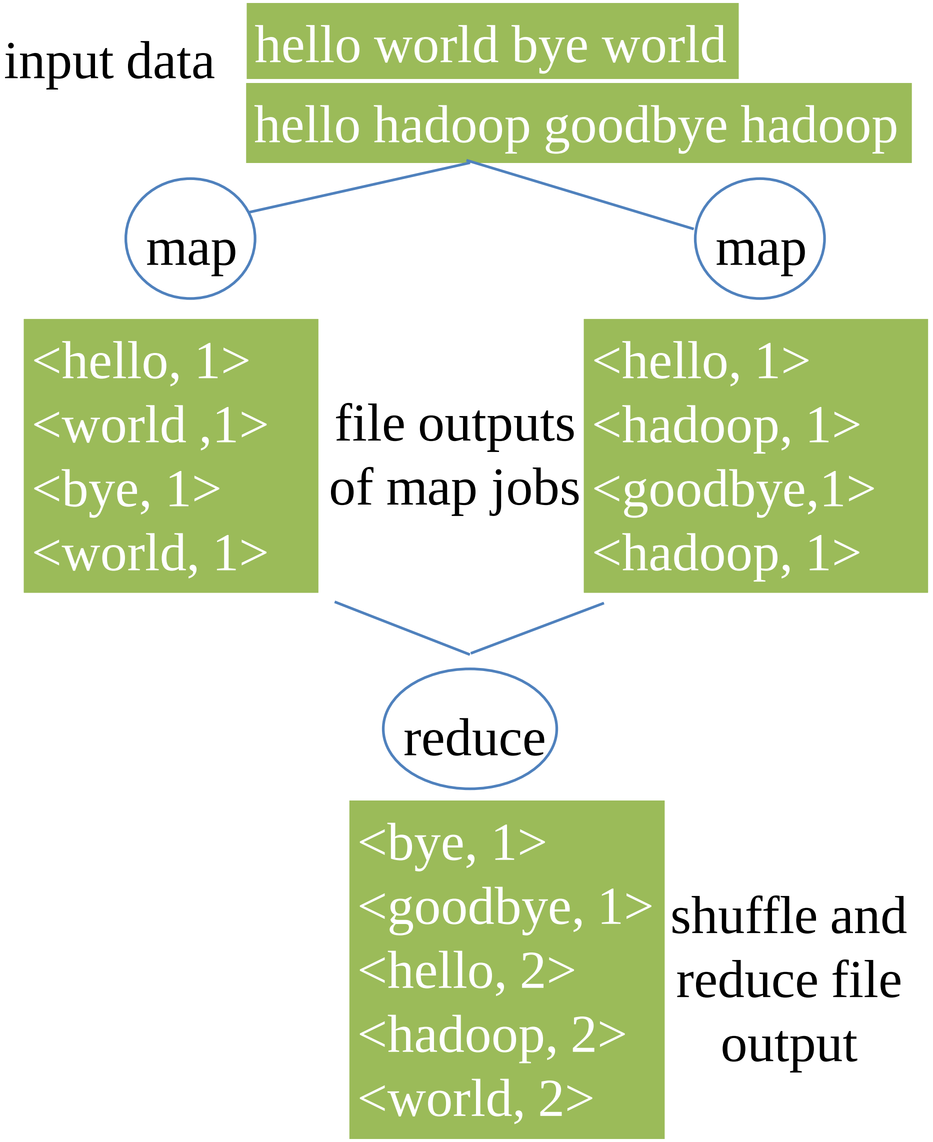
Data is partitioned into files on DataNodes.
NameNode is the file system management node.

Hadoop cluster of DataNodes that also run compute Tasks having one Master Node for management and control at compute task and file system layers.

Example - Word count

```
public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }


public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
```



input data

hello world bye world

hello hadoop goodbye hadoop

<hello, 1>
<world ,1>
<bye, 1>
<world, 1>

file outputs of map jobs

<hello, 1>
<hadoop, 1>
<goodbye,1>
<hadoop, 1>

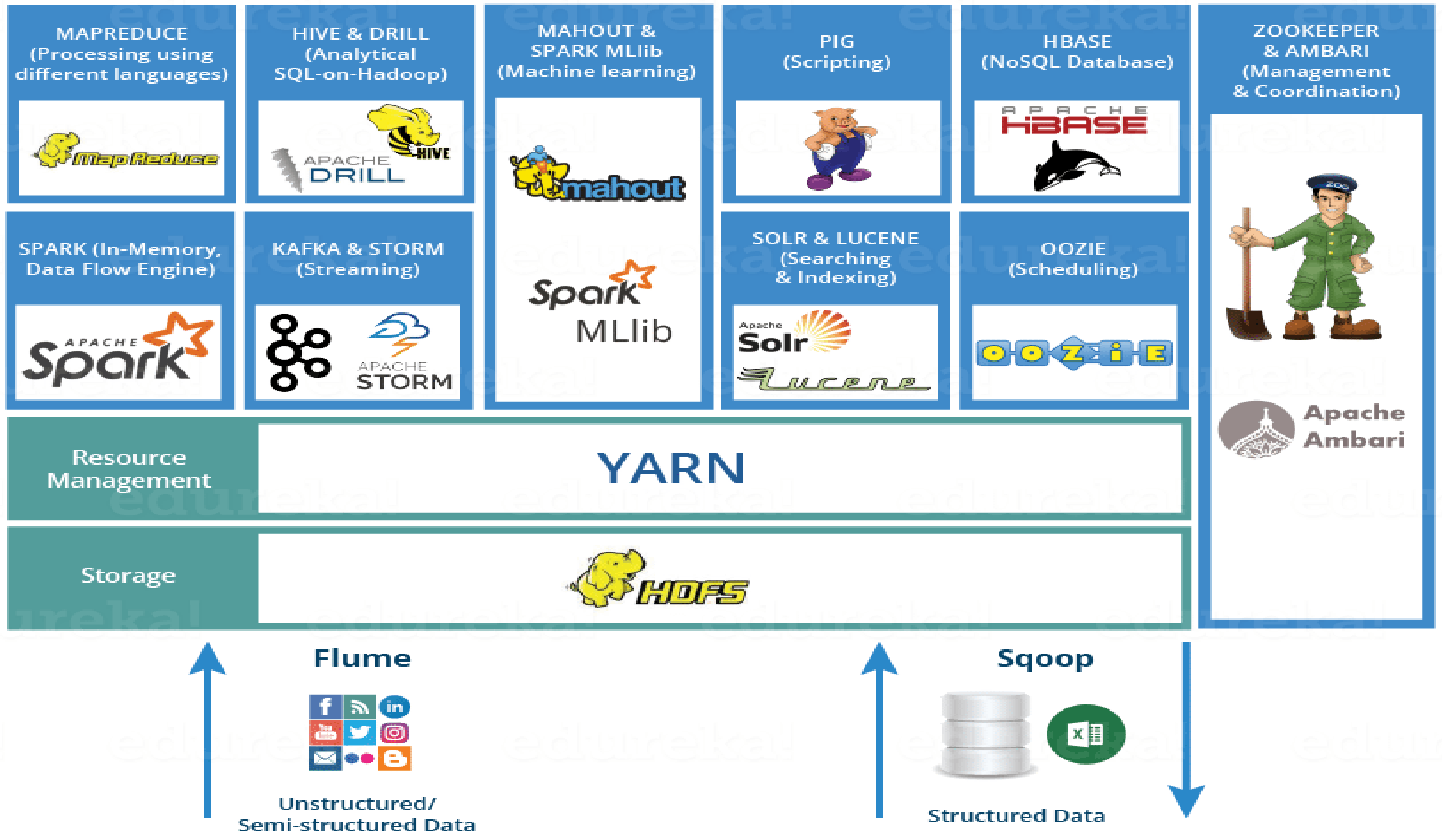
<bye, 1>
<goodbye, 1>
<hello, 2>
<hadoop, 2>
<world, 2>

shuffle and reduce file output

Advantages of using Hadoop

- Low cost – open source and low cost commodity storage
- Computing power – many nodes can be used for computation
- Scalability – simple to add nodes in system for parallel processing and storage
- Storage Flexibility – can store unstructured data easily
- Inherent data protection – protects against hardware failures

Hadoop Ecosystem



MAPREDUCE (Processing using different languages)

HIVE & DRILL (Analytical SQL-on-Hadoop)

MAHOUT & SPARK MLlib (Machine learning)

PIG (Scripting)

HBASE (NoSQL Database)

ZOOKEEPER & AMBARI (Management & Coordination)

SPARK (In-Memory, Data Flow Engine)

KAFKA & STORM (Streaming)

SOLR & LUCENE (Searching & Indexing)

OOZIE (Scheduling)

Resource Management — YARN

Storage — HDFS

Flume — Unstructured/ Semi-structured Data

Sqoop — Structured Data

Image Source : Edureka

Issues with MapReduce on Hadoop

- ✓ Revolutionized big data processing, enabling users to store and process huge amounts of data at very low costs.
- ✓ An ideal platform to implement complex batch applications as diverse as
  - sifting through system logs
  - running ETL
  - computing web indexes
  - recommendation systems etc.
- ✓ Its reliance on persistent storage to provide fault tolerance and its one-pass computation model make MapReduce a poor fit for
  - low-latency applications
  - iterative computations, such as machine learning and graph algorithms
    - There are extensions for iterative MapReduce that we study later

Adapted from : https://databricks.com/blog/2013/11/21/putting-spark-to-use.html

In-Memory Computing

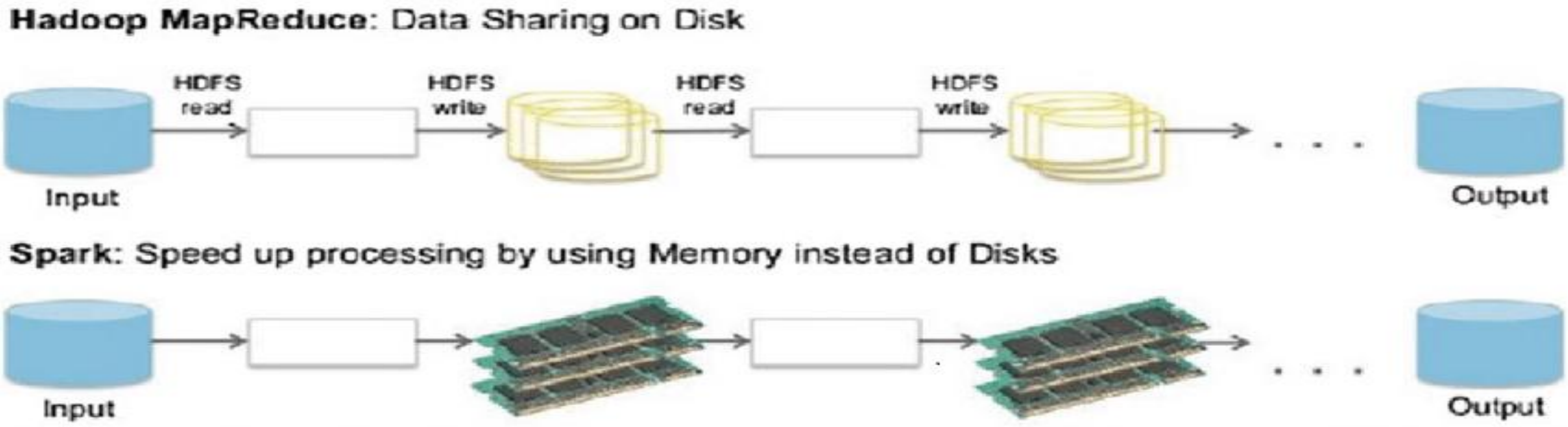
In-memory computing

- ✓ means using a type of middleware software that allows one to store data in RAM, across a cluster of computers, and process it in parallel

For example,

- ✓ Operational datasets typically stored in a centralized database which you can now store in “connected” RAM across multiple computers.
- ✓ RAM is roughly 5,000 times faster than traditional spinning disk.
- ✓ Native support for parallel processing makes it faster

Note:
Could be batch or streaming


Hadoop MapReduce: Data Sharing on Disk
HDFS read
HDFS write
HDFS read
HDFS write
Input
Output
Spark: Speed up processing by using Memory instead of Disks
Input
Output

Fast and easy big data processing with Spark

At its core, Spark provides a general programming model that enables developers to write application by composing arbitrary operators, such as

- ✓ mappers
- ✓ reducers
- ✓ joins
- ✓ group-bys
- ✓ filters

This composition makes it easy to express a wide array of computations, including iterative machine learning, streaming, complex queries, and batch.

Spark keeps track of the data that each of the operators produces, and enables applications to reliably store this data in memory using RDDs*.

- ✓ This is the key to Spark’s performance, as it allows applications to avoid costly disk accesses.

Example - Word count

```
sparkContext.textFile("hdfs://...")
  .flatMap(line => line.split(" "))
  .map(word => (word, 1))
  .reduceByKey(_ + _)

.saveAsTextFile("hdfs://...")
```

convert a file into lines

map: transform each word into <k,v> pair

reduce: sum up values for each key

- Can read data from many sources, including HDFS
- Map / reduce output is written to memory instead of files
- Memory content can be written out to files etc.
- A rich set of primitives on top of MapReduce model to make it easier to program

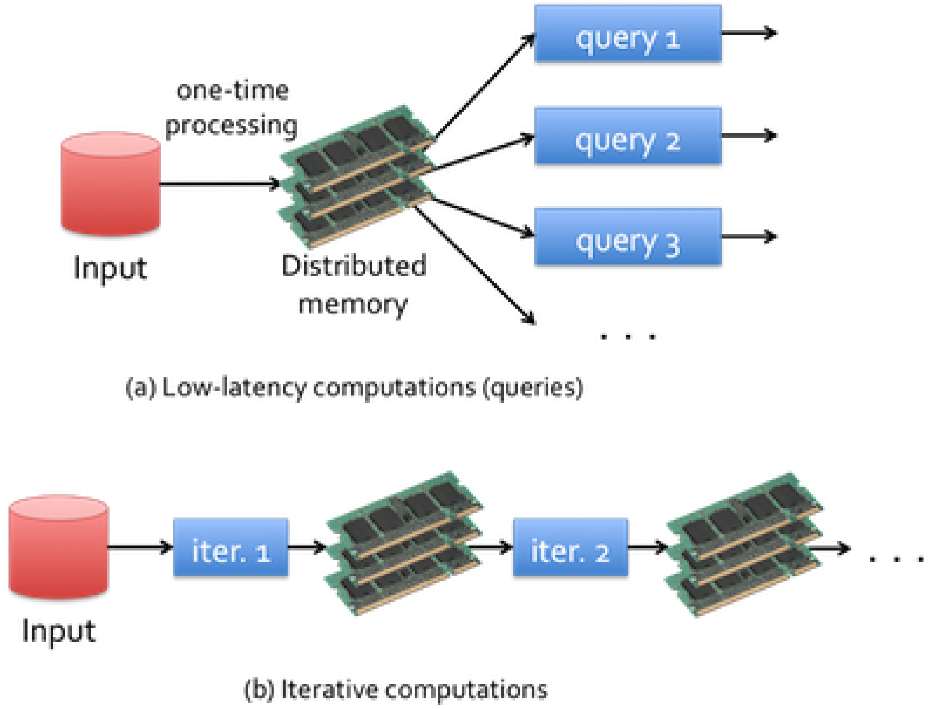
Ideal Apache Spark applications

Low-latency computations

✓ by caching the working dataset in memory and then performing computations at memory speeds

Efficient iterative algorithm

✓ by having subsequent iterations share data through memory, or repeatedly accessing the same dataset



(a) Low-latency computations (queries)

(b) Iterative computations

Cloud Computing-Definition

The US National Institute of Standards (NIST) defines cloud computing as follows:

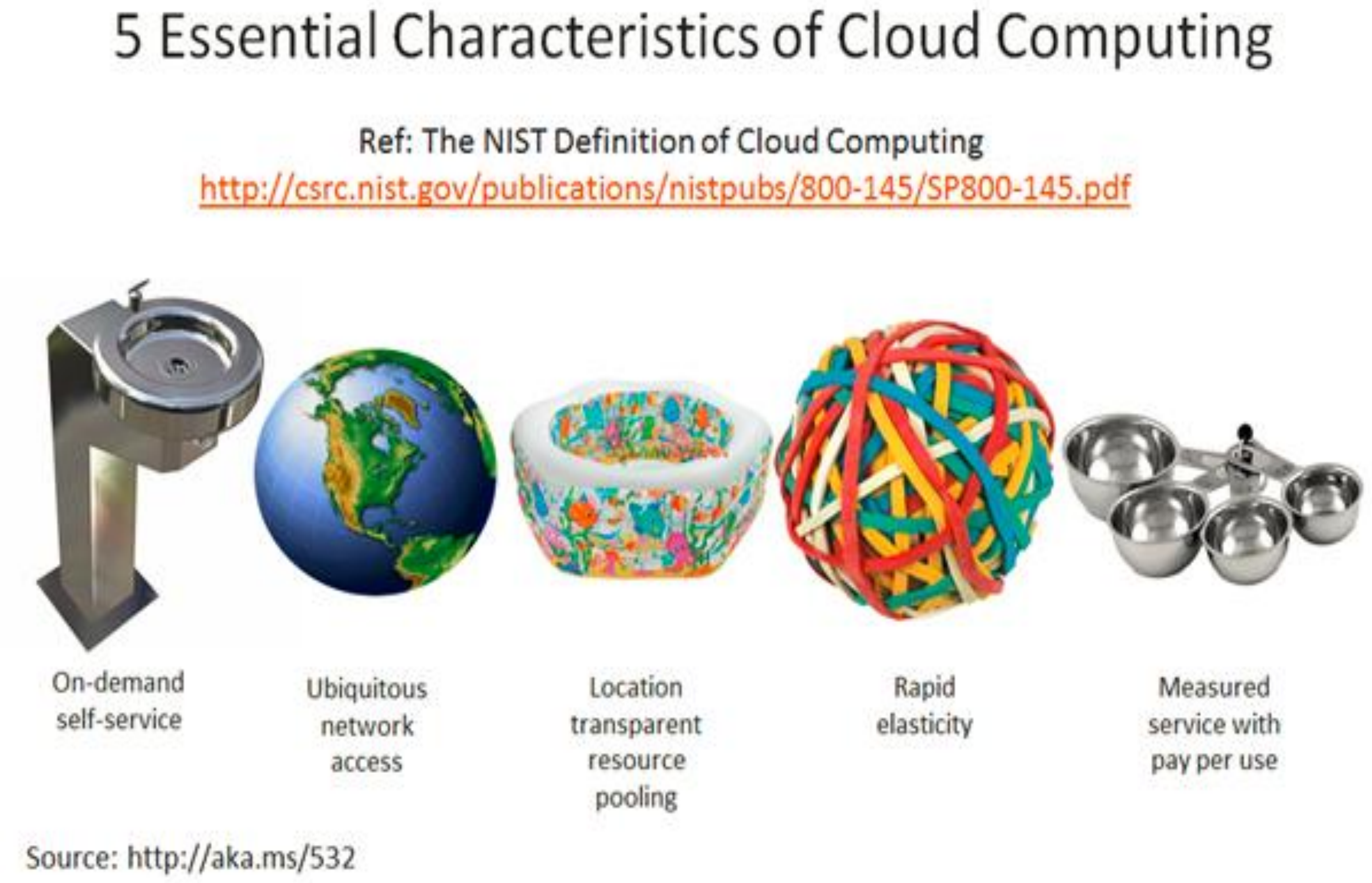
Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

NIST's 3-4-5 rule of Cloud Computing

- 3 cloud service models or service types for any cloud platform
- 4 deployment models
- 5 essential characteristics of cloud computing infrastructure

5 Characteristics of Cloud Computing

- On demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service



5 Essential Characteristics of Cloud Computing

Ref: The NIST Definition of Cloud Computing
http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

On-demand self-service

Ubiquitous network access

Location transparent resource pooling

Rapid elasticity

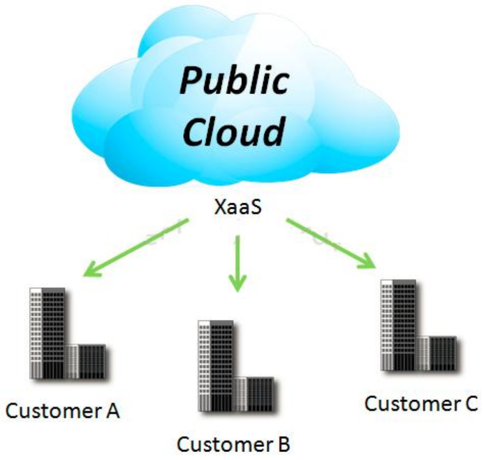
Measured service with pay per use

Source: http://aka.ms/532

4 Deployment models

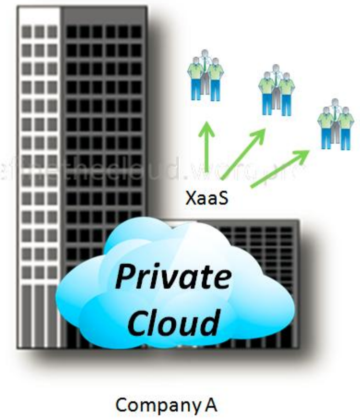
Public Cloud

Mega-scale cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services. E.g. AWS, Azure, Google, …



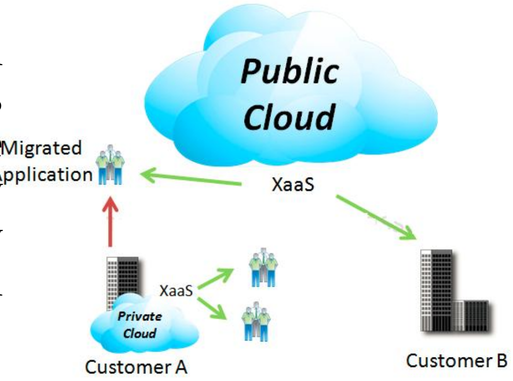
Private Cloud

The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.



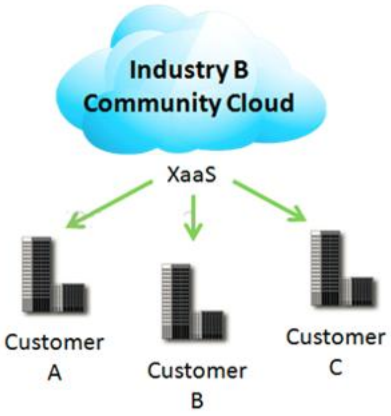
Hybrid Cloud

The cloud infrastructure is a composition of two or more clouds (private or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability, cross domain security etc.

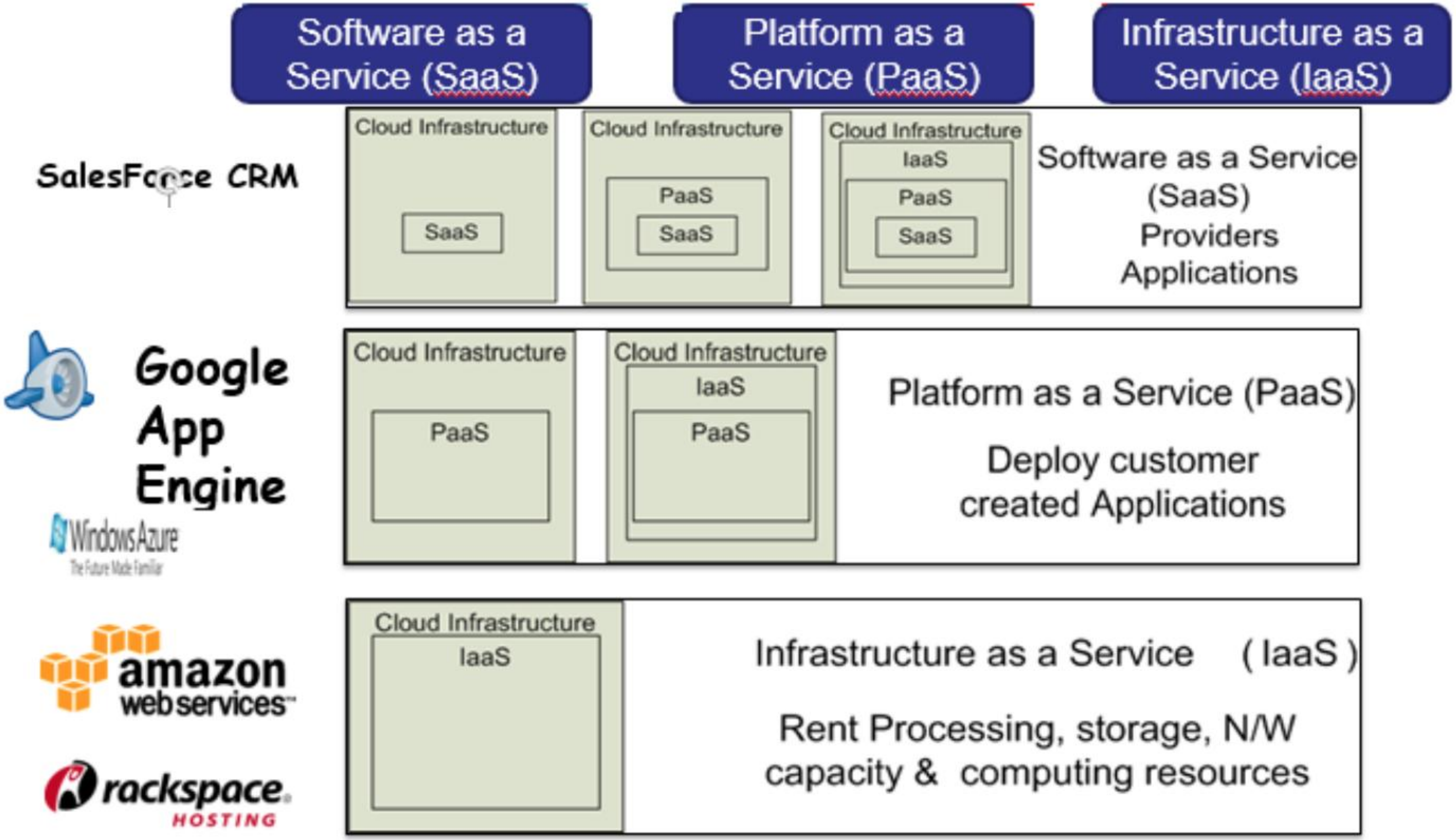


Community Cloud

An ‘infrastructure shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise’ according to NIST.



3 Cloud Service Models



Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

SalesForce CRM

Cloud Infrastructure

SaaS

Cloud Infrastructure

PaaS

SaaS

Cloud Infrastructure

IaaS

PaaS

SaaS

Software as a Service (SaaS) Providers Applications

Google App Engine

WindowsAzure

The Future Made Familiar

Cloud Infrastructure

PaaS

Cloud Infrastructure

IaaS

PaaS

Platform as a Service (PaaS)

Deploy customer created Applications

amazon web services

rackspace HOSTING

Cloud Infrastructure

IaaS

Infrastructure as a Service (IaaS)

Rent Processing, storage, N/W capacity & computing resources

Cloud services for Big Data

Cloud follows same model as Big Data, both requiring distributed clusters of computing devices.

- ✓ Cloud Computing considered as ideal platform for handling big data

IaaS:

- ✓ Can provide huge storage and computational power requirements for Big Data through limitless storage and computing ability of cloud computing, e.g. AWS S3, EC2

PaaS:

- ✓ Vendors offers platforms ready with Hadoop and MapReduce (AWS EMR).
- ✓ Saves hassles of installations and managements of these environments

SaaS:

- ✓ Great help to organizations which requires specialized software's for big data like for social media analytics, feedback monitoring etc.
- ✓ SaaS vendors provides out of the box solution for such common use cases

Cloud providers in Big Data market

Amazon (Amazon Web Services AWS)

- ✓ EC2
- ✓ Elastic MapReduce
- ✓ DynamoDB
- ✓ Amazon S3
- ✓ High Performance Computing
- ✓ Redshift

Google (Google Cloud Platform GCP)

- ✓ Google Compute Engine
- ✓ Google BigQuery
- ✓ Google Prediction API

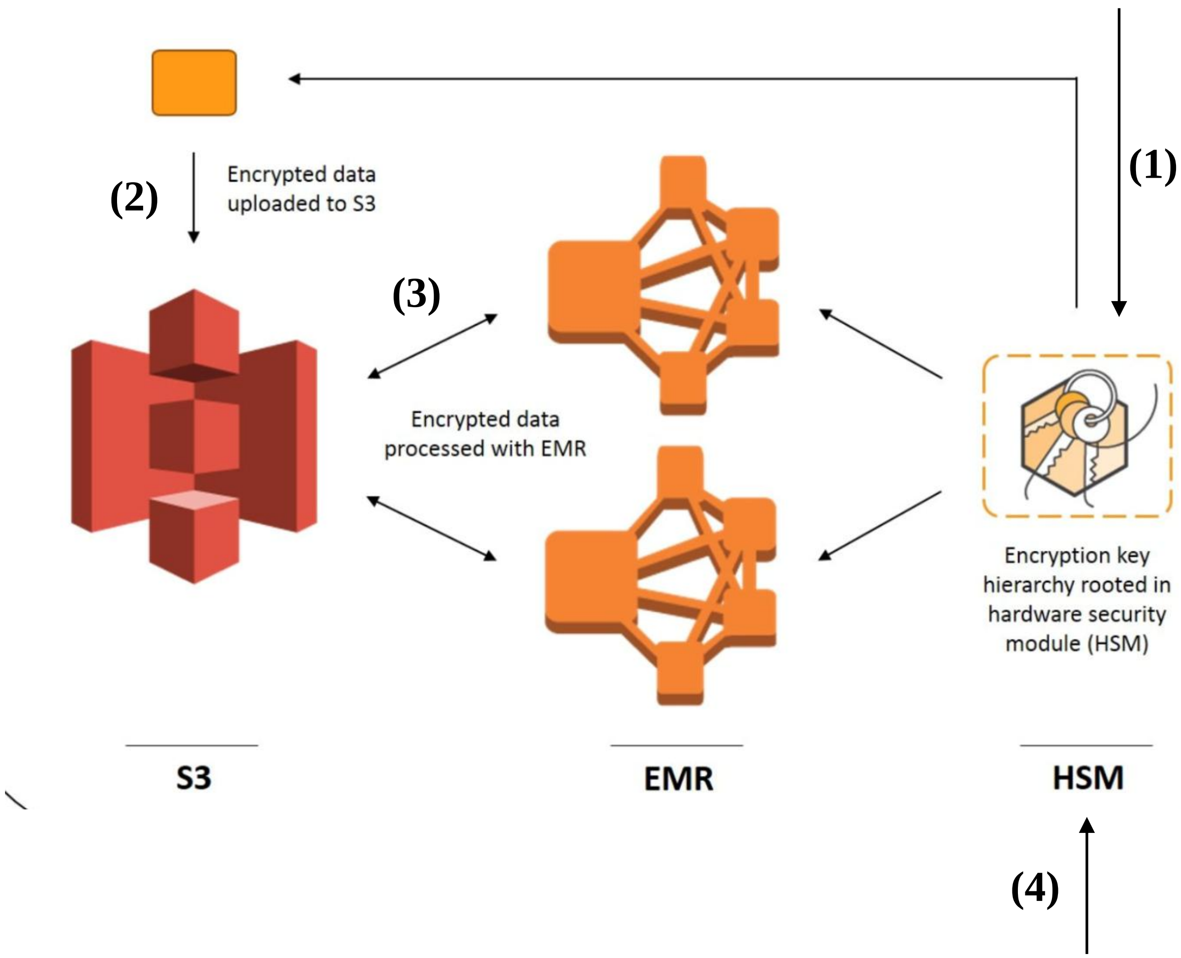
Windows Azure

- ✓ Azure PaaS cloud based on Windows and SQL
- ✓ Windows Azure HD Insight

Case study: Nasdaq

- Operates financial exchanges
- AWS Redshift is the data warehouse with 5.5B rows/day with peak 14B/day in Oct 2014.
- A new DWH environment is setup using EMR and S3
  - give more teams access to gigantic data sets
  - cost efficiency
- Hadoop enables large and more varied historical data access at lower cost
- Why EMR and S3 ?
  - On demand set up of Hadoop clusters on Cloud to run short term projects.
  - Existing tech to move around TBs of data.
  - Storage on Cloud in S3 to scale to very large data sets.
- Why not HDFS and use S3 ?
  - Decouple data and compute because data size is disproportionately higher than compute frequency - so can't add Hadoop nodes just for data esp with replication factor
  - Similar to Netflix. EMRFS layer can enable compute nodes access data in S3

Data from variety of sources


(1)
(2)
Encrypted data uploaded to S3
(3)
Encrypted data processed with EMR
Encryption key hierarchy rooted in hardware security module (HSM)
S3
EMR
HSM
(4)

Nasdaq case study

SQL and other queries

THANK YOU