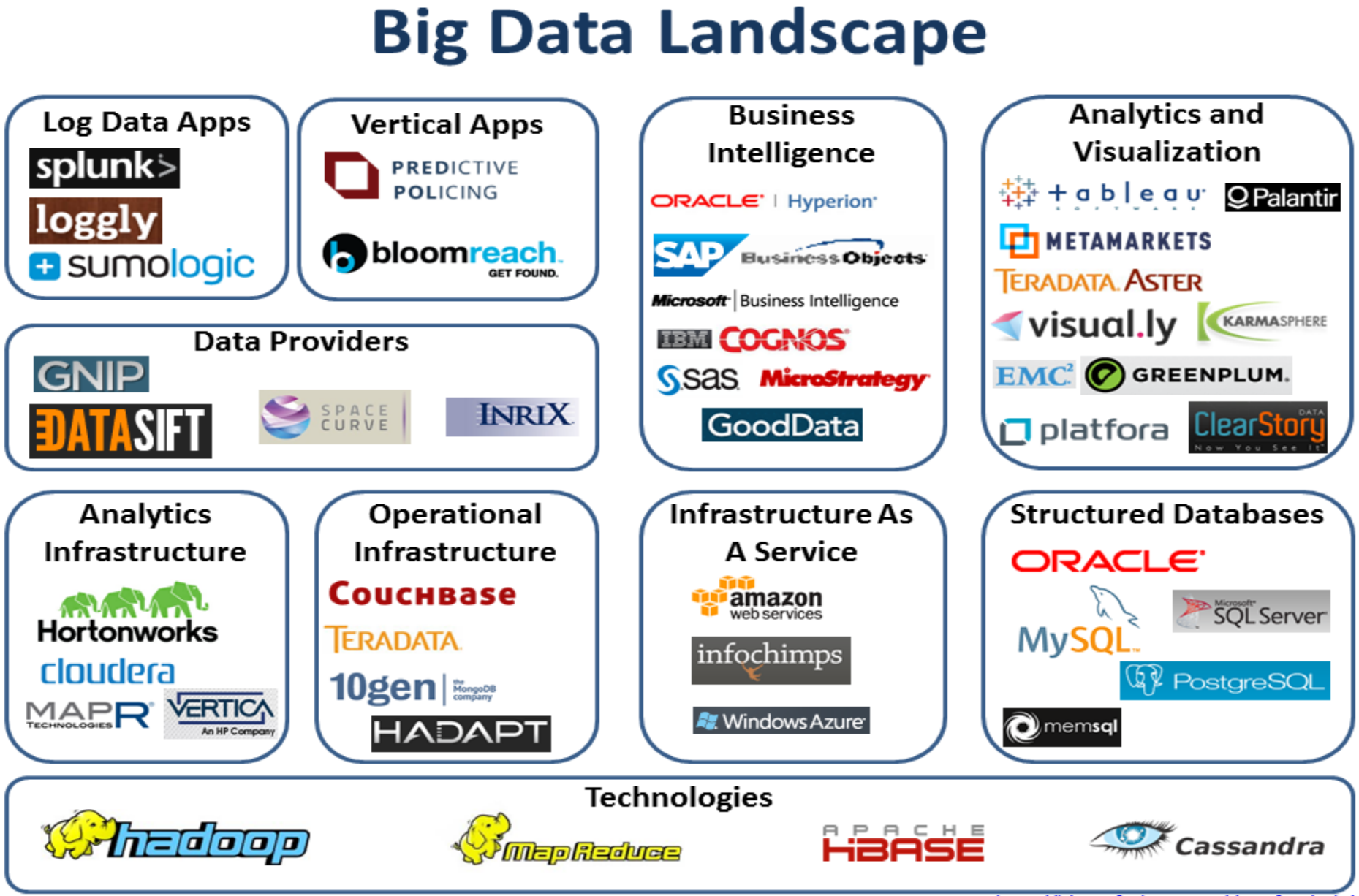
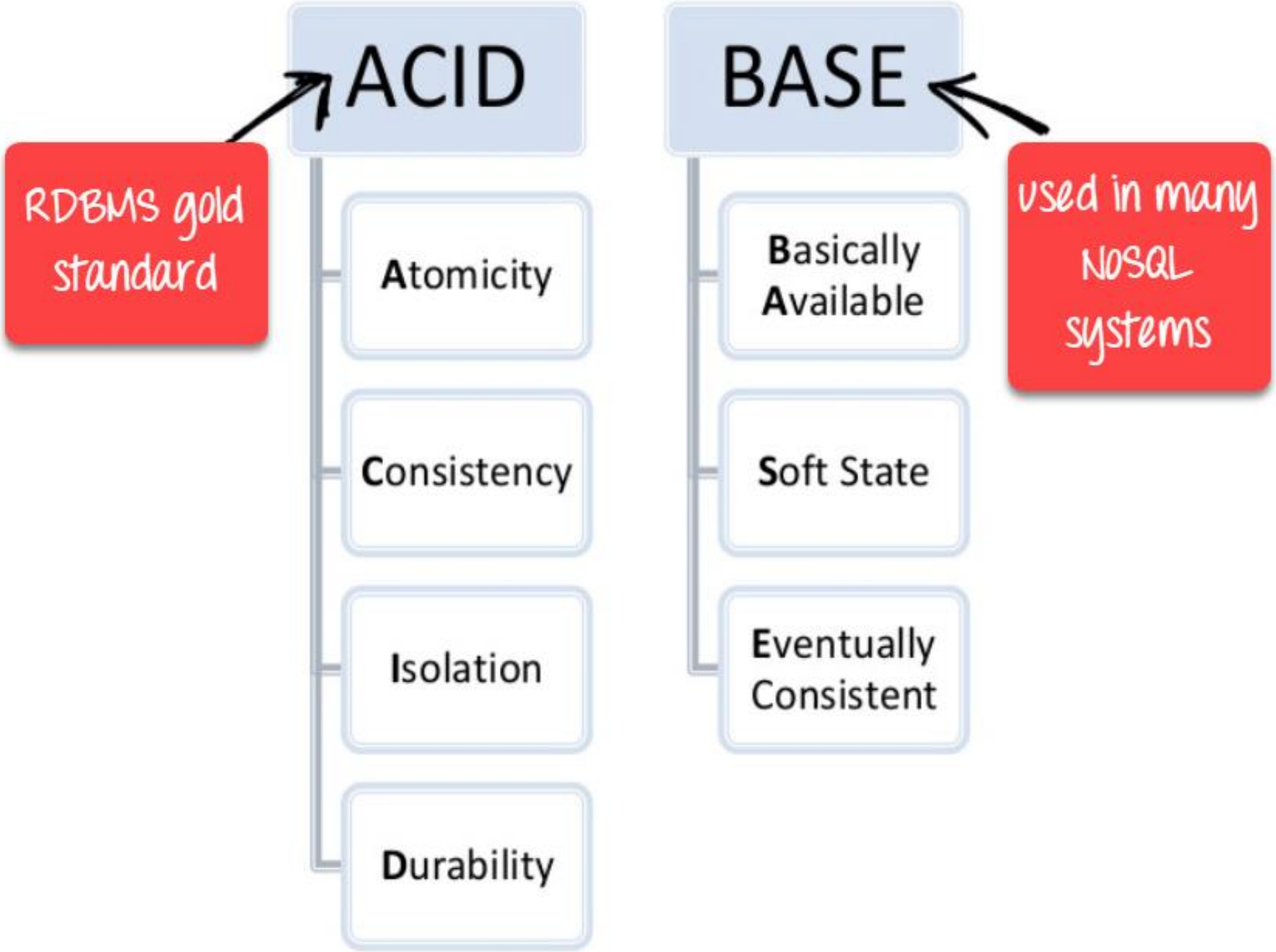


CHAPTER 03: Big Data technology landscape

Agenda

- NoSQL
  - What is it?
  - Types of NoSQL Databases
  - Why NoSQL?
  - Advantages of NoSQL
  - NoSQL Vendors
  - SQL versus NoSQL
  - NewSQL
  - Comparison of SQL, NoSQL and NewSQL
- Hadoop
  - Features of Hadoop
  - Key Advantages of Hadoop
  - Versions of Hadoop

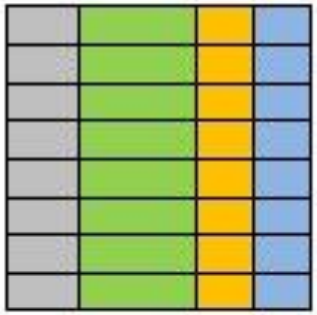

ACID
RDBMS gold standard
Atomicity
Consistency
Isolation
Durability
BASE
used in many NOSQL systems
Basically Available
Soft State
Eventually Consistent

NoSQL (Not Only SQL)

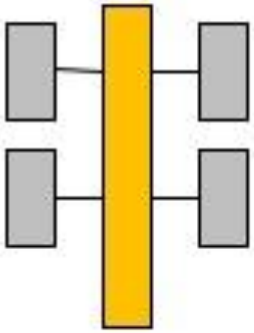
- The term **NoSQL** was **coined** by Carlo Strozzi in the year 1998. He used this term to name his Open Source, Light Weight, Database which did not have an SQL interface.
- It is triggered by the needs of Web 2.0 companies such as Facebook, Google, and Amazon.com.
- Most NoSQL databases offer a concept of "eventual consistency" in which database changes are propagated to all nodes "eventually" (typically within milliseconds).

Types of databases

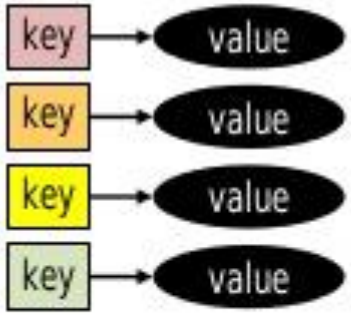
Relational



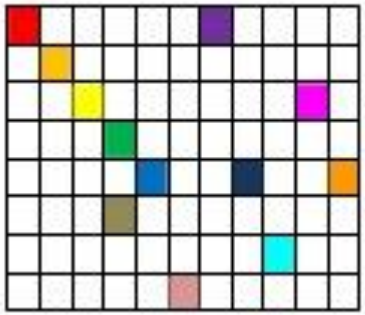
Analytical (OLAP)



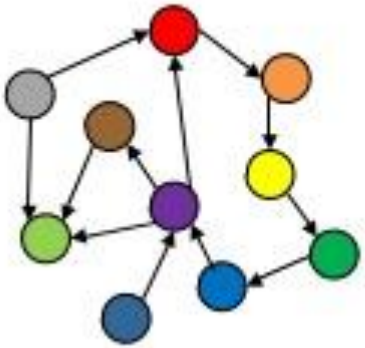
Key-Value



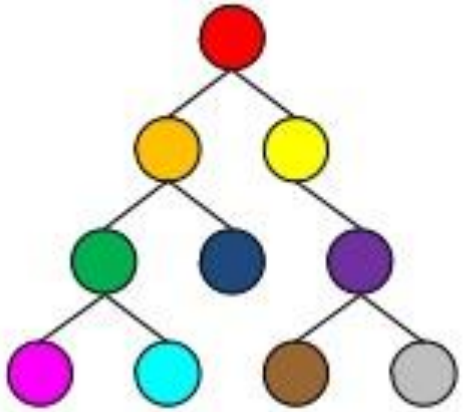
Column-Family

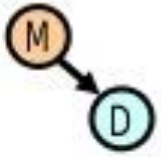


Graph

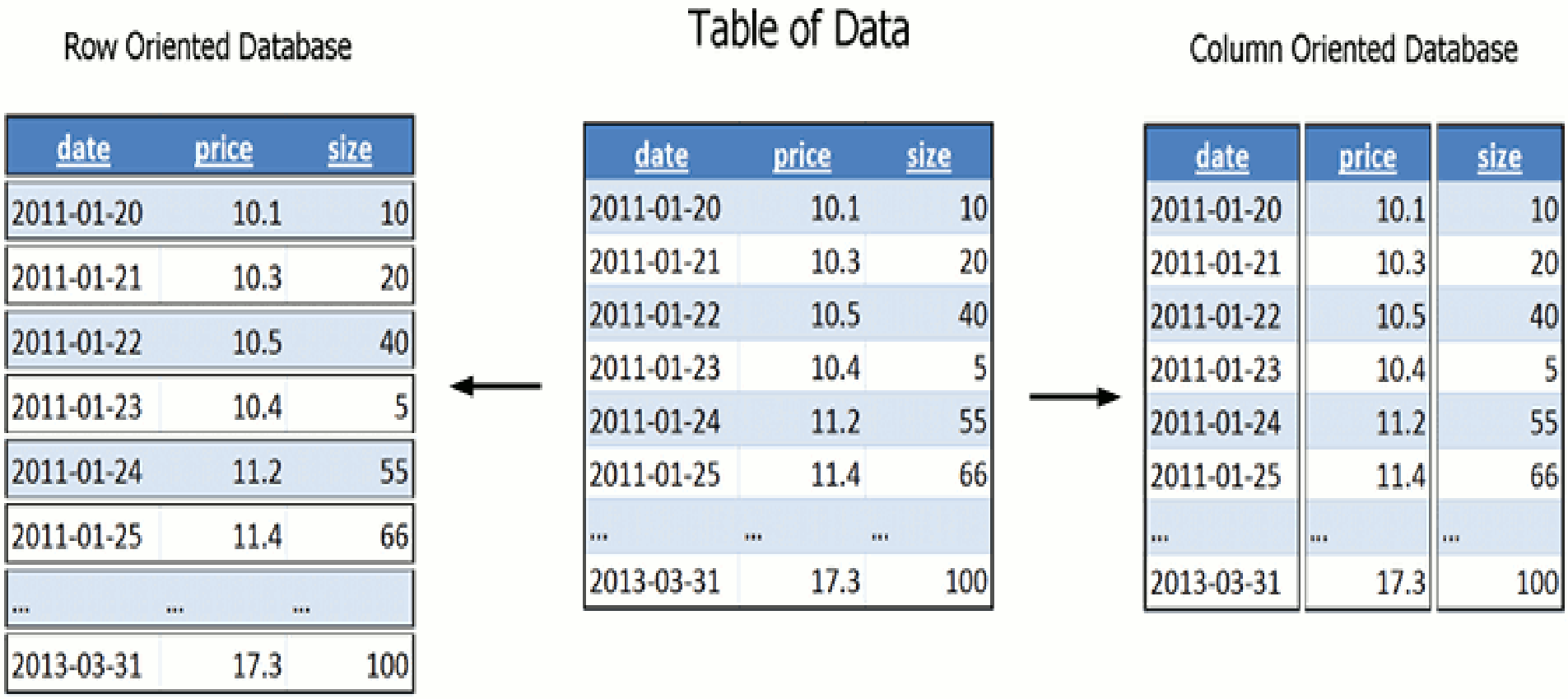


Document





Column oriented database



A column store database can also be referred to as a:

- Column database
- Column family database
- Column oriented database
- Wide column store database
- Wide column store
- Columnar database
- Columnar store

Where NoSQL is used?

- Big data (SD+USD+SSD [3Vs Data]) [IOT data]
- Real time (Time based data) web applications.
- Log data storage and analysis
- Social media data storage and analysis.

Example:

- Amazon's Shopping Cart (Amazon Dynamo is used)
- Netflix uses : SimpleDB, HBase and Cassandra.
- Facebook uses Cassandra and Hive

What is NoSQL?

- NoSQL refers to a general class of storage engines that store data in a non-relational format.
- A **NoSQL** is a non-relational (No tables), open source distributed database used for dealing with big data (SSD, USD and SD).
- NoSQL encompasses a wide range of technologies and architectures, to solve the scalability and big data .

What is NoSQL?

- NoSQL databases can solve problems regular databases can't handle:
  - indexing the entire Internet,
  - predicting subscriber behavior, or
  - targeting ads on a platform such as Facebook, YouTube, etc.

What is NoSQL?

- NoSQL databases especially target large sets of distributed data.
- NoSQL databases: (Other names) cloud databases, non-relational databases, or Big Data databases.
- NoSQL databases have become the first alternative to relational databases, with high performance, scalability (Horizontal), availability, and fault tolerance (which are key deciding factors).


redis
CouchDB
relax
RAVENDB
Neo4j
Cassandra
amazon
DynamoDB
mongoDB
APACHE
HBASE
membase
riak

NoSQL features

1. NoSQL databases are non-relational: No adhere to relational data model.
2. Distributed: Data is distributed across several nodes in a cluster of low-cost commodity hardware.
3. No support for ACID properties (Atomicity, Consistency, Isolation, and Durability): They adhere to CAP theorem.
4. No fixed table schema: Support for flexible schema i.e. no mandate for the data to strictly adhere to any schema structure at the time of storage.

Types of NoSQL databases

1. Key-Value store: Riak, Redis, Membase
2. Document oriented store: MongoDB, CouchDB, RavenDB
3. Column oriented store: Cassandra, HBase, HyperTable
4. Graph oriented database: InfiniteGraph, Neo4j, AllegroGraph

Key-Value store/databases

- These databases are designed for storing data in a schema-less way.
- In a key-value store, all of the data consists of an indexed key and a value, hence the name.
- It maintains a big hash table of keys and values.
- Examples:
  - DyanmoDB (Amazon), Azure Table Storage (ATS), Riak, BerkeleyDB.
  - Shopping carts, web user data analysis (Amazon and LinkedIn)

Key-Value pair Example

Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

Document oriented store

- A document database is a type of non-relational database that is designed to store semi-structured data as documents, typically in JSON or XML format.
- Documents are grouped into "collections," which is similar to a table in a relational database.
- A document database is used for storing, retrieving, and managing semi-structured data.
- The data model in a document database is not structured in a table format of rows and columns. The schema can vary, providing more flexibility for data modeling.
- Examples : MongoDB, MarkLogic, Apache CouchDB, etc

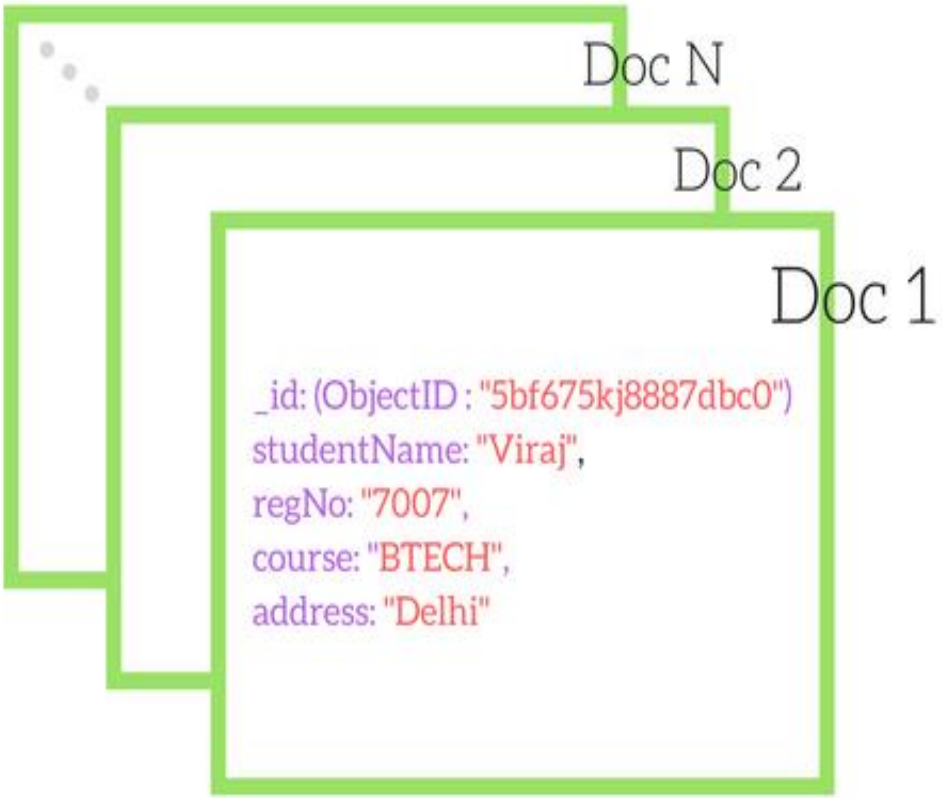

Key
<Document>
{
"customerid": "fc986e48ca6"
"customer":
{
"firstname": "Pramod",
"lastname": "Sadalage",
"company": "ThoughtWorks"
"likes": ["Biking","Photography"]
}
"billingaddress":
{ "state": "AK",
"city": "DILLINGHAM",
"type": "R"
}
}

Example

```
{
        "FirstName": "Bob",
        "Address": "5 Oak St.",
        "Hobby": "sailing"
}
```

Collection

A collection can have multiple documents



Applications of Document Databases

- Web analytics
- User preferences data
- Tweets
- Comments
- Sensor data from mobile devices
- Log files
- Real-time analytics
- Various other data from Internet of Things
- Product catalogs and so on.

Document Databases are used in:

- LinkedIn
- Dropbox Mailbox
- Large eCommerce platforms (Like Amazon)
- Blogging sites (such as Twitter)
- Content management systems (WordPress, windows registry)
- Analytical platforms
- Web analytics
- User preferences data

Column oriented store

- Column store: designed for storing data tables as sections of columns of data, rather than as rows of data.
- wide-column stores offer very high performance and a highly scalable architecture.
  - Examples include: Cassandra (Face book), HBase, BigTable (Google) and HyperTable.

Example

Table

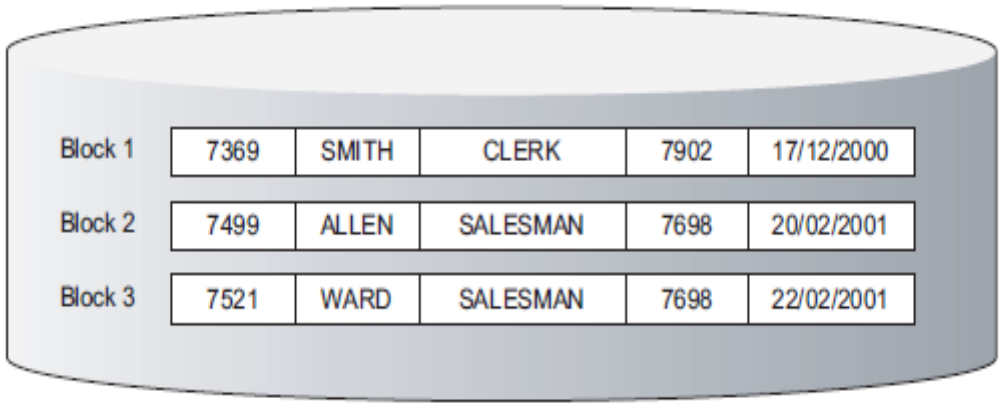
	Country	Product	Sales
Row 1	India	Chocolate	1000
Row 2	India	Ice-cream	2000
Row 3	Germany	Chocolate	4000
Row 4	US	Noodle	500

Row Store

Row 1	India
	Chocolate
	1000
Row 2	India
	Ice-cream
	2000
Row 3	Germany
	Chocolate
	4000
Row 4	US
	Noodle
	500

Column Store

Country	India
	India
	Germany
	US
Product	Chocolate
	Ice-cream
	Chocolate
	Noodle
Sales	1000
	2000
	4000
	500

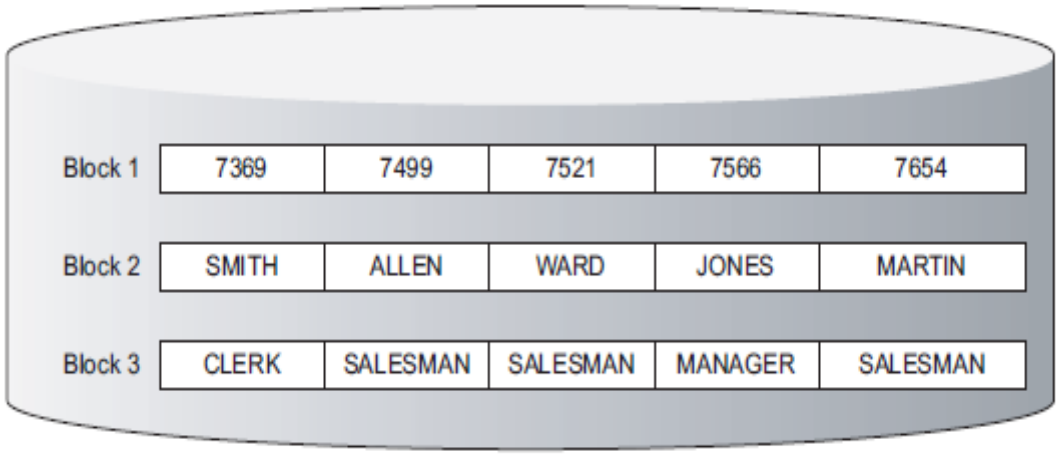


Row Database stores row values together

Row-Store Physical Layout

EmpNo	EName	Job	Mgr	HireDate
7369	SMITH	CLERK	7902	17/12/1980
7499	ALLEN	SALESMAN	7698	20/02/1981
7521	WARD	SALESMAN	7698	22/02/1981
7566	JONES	MANAGER	7839	2/04/1981
7654	MARTIN	SALESMAN	7698	28/09/1981
7698	BLAKE	MANAGER	7839	1/05/1981
7782	CLARK	MANAGER	7839	9/06/1981

Logical Schema



Column Database stores column values together

Column Store physical layout

Column databases are used in:

- Google Earth, Maps
- The New York Times
- eBay
- Twitter
- Facebook
- Netflix (Streaming service: TV, News, Movies, etc.)
- Sensor feeds
- Web user actions analysis.

Google Earth

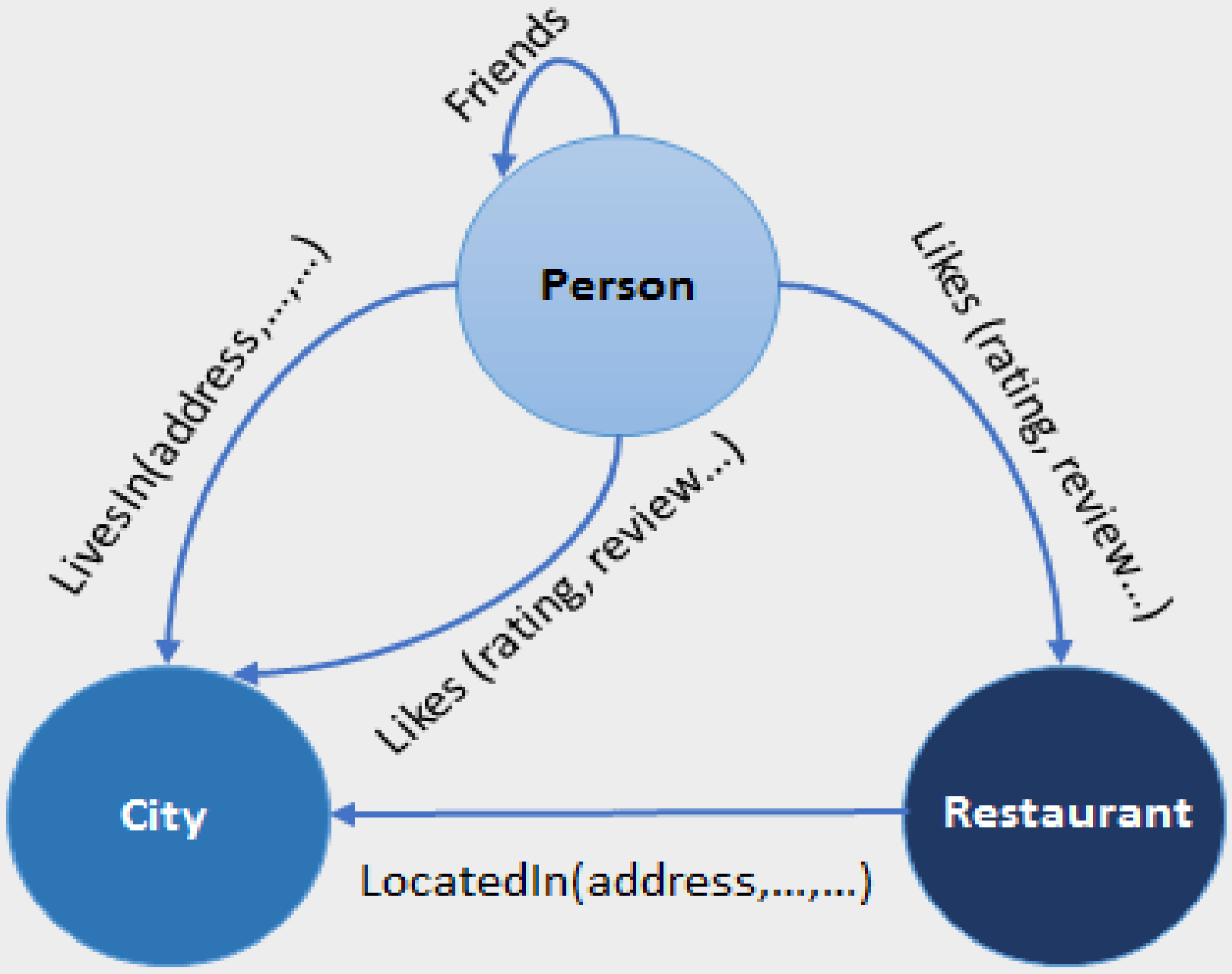
- **Google Earth is a computer program that renders a 3D representation of Earth based on satellite imagery.**
- **The program maps the Earth by superimposing satellite images, aerial photography, and GIS data onto a 3D globe, allowing users to see cities and landscapes from various angles.**

Graph oriented database

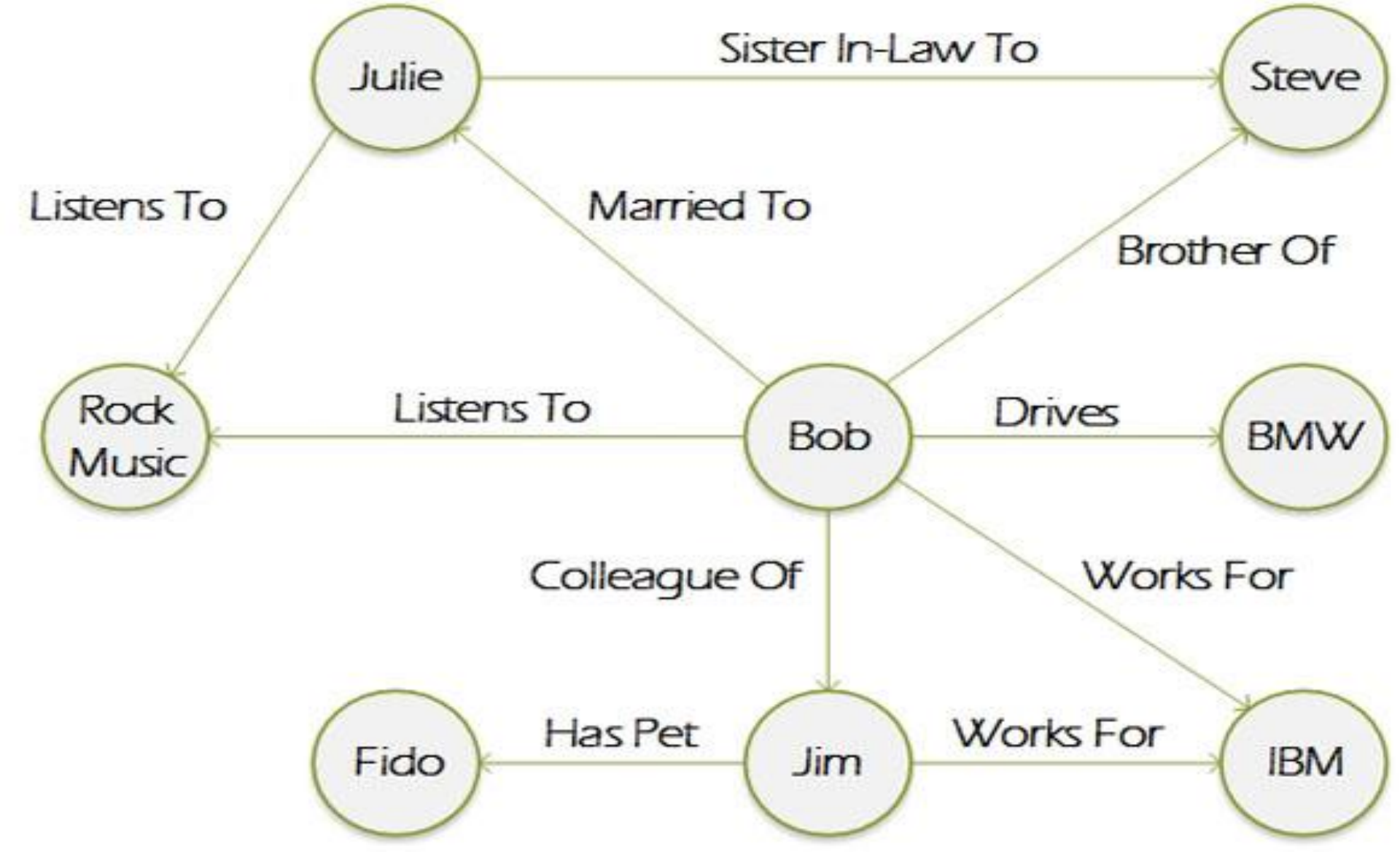
- A **graph database**, also called a **graph-oriented database**, is a type of NoSQL **database** that uses **graph** theory to store, map and query relationships.
- A graph database is an online database management system with Create, Read, Update and Delete (CRUD) operations working on a graph data model.
- Examples : Neo4j, Titan, Polyglot, HyperGraphDB, InfiniteGraph
- Graph databases are used on Social Network, Walmart-Upsell, Cross-sell, Recommendation.

Graph oriented database

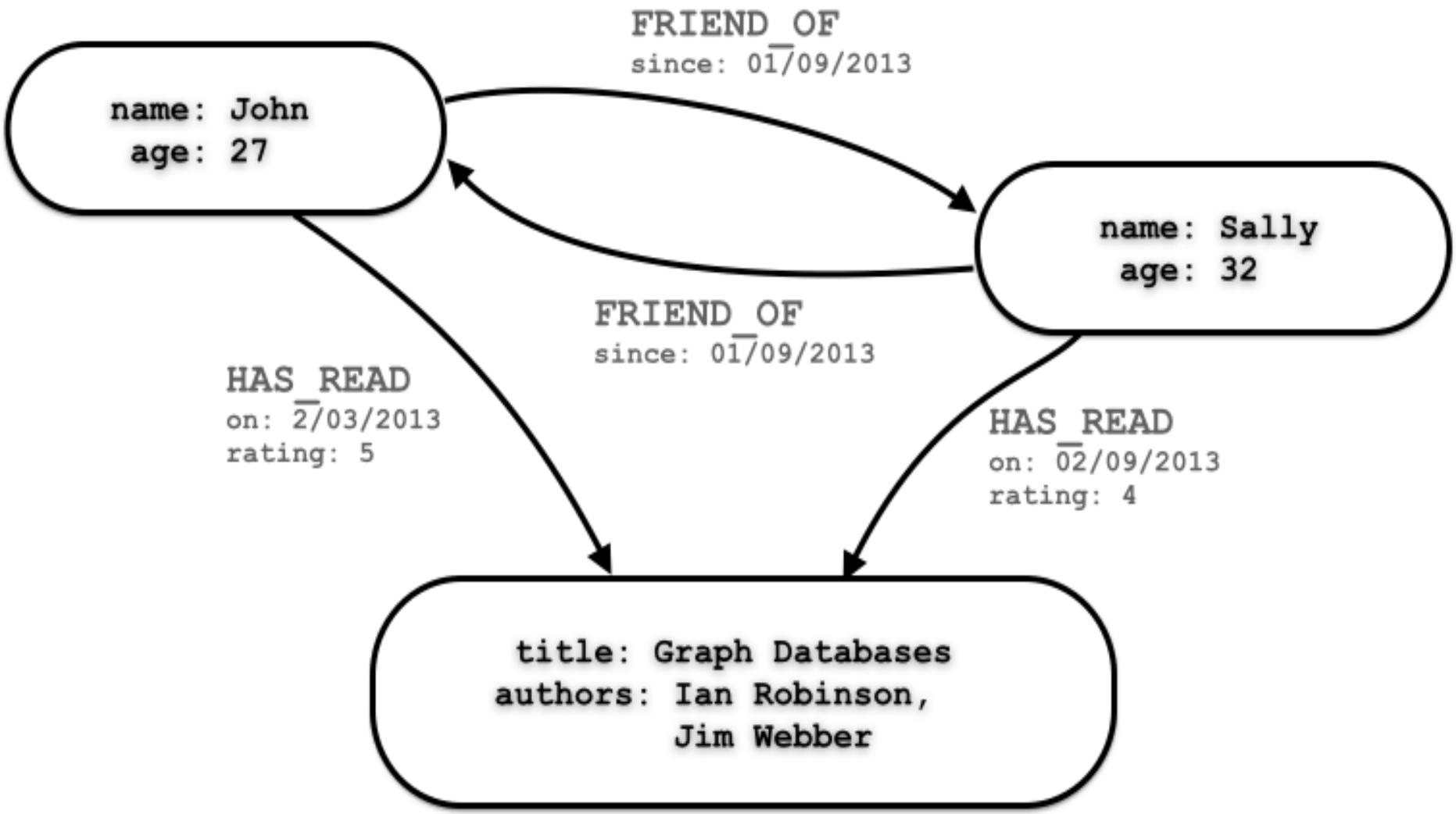
- A graph database is essentially a collection of nodes and edges.
- Each node represents an entity (such as a person or business) and each edge represents a connection or relationship between two nodes.
- Every node in a graph database is defined by a unique identifier, a set of outgoing edges and/or incoming edges and a set of properties expressed as key/value pairs.
- Each edge is defined by a unique identifier, a starting-place and/or ending-place node and a set of properties.


Friends
Person
LivesIn(address,...,...)
Likes (rating, review...)
Likes (rating, review...)
City
Restaurant
LocatedIn(address,...,...)

Graph oriented database



Graph oriented database



Why NoSQL

1. It has a scale out architecture, i.e. consisting of multiple low-cost machines -- that are configured to create a storage pool and to increase computing power by adding more nodes. (Horizontal scalability)
2. It can house large volumes of SD, USD and SSD.
3. Supports dynamic schema: Database allows insertion of data without a pre-defined schema.

Why NoSQL

4. Auto sharding : It automatically spreads data across an arbitrary number of servers. It balances the load of data and query on the available servers. It also supports self healing capability.
5. Replication: It offers good support for replication which in turn guarantees high availability, fault tolerance, and disaster recovery.
6. Support large numbers of concurrent users (tens of thousands, perhaps millions)

Why NoSQL?

7. In recent times we can easily capture and access data from various sources, like Facebook, Google, Twitter, Amazon, etc.
8. User's personal information, geographic location data, user generated content, social graphs and machine logging data are some of the examples where data is increasing rapidly.
9. Relational databases are not suitable for processing large volume of data.

Advantages of NoSQL

1. Support elastic scaling:
   a) Cluster scale: It allows distribution of database across 100+ nodes often in multiple data centers.
   b) Performance scale: It sustains over 100,000+ database reads and writes per second.
   c) Data scale: It supports housing of 1 billion+ documents in the database.
2. Doesn’t require a pre-defined schema: Does not require any adherence to pre-defined schema and supports flexible schema

Example (MongoDB)

1. . {_id:101, “Book Name”: ”Fundamentals of business analytics”, “Author Name”: “Seema Acharya”, “Publisher” :”Wiley India”}
2. {_id:102, “Book name”: “Big data and analytics”}

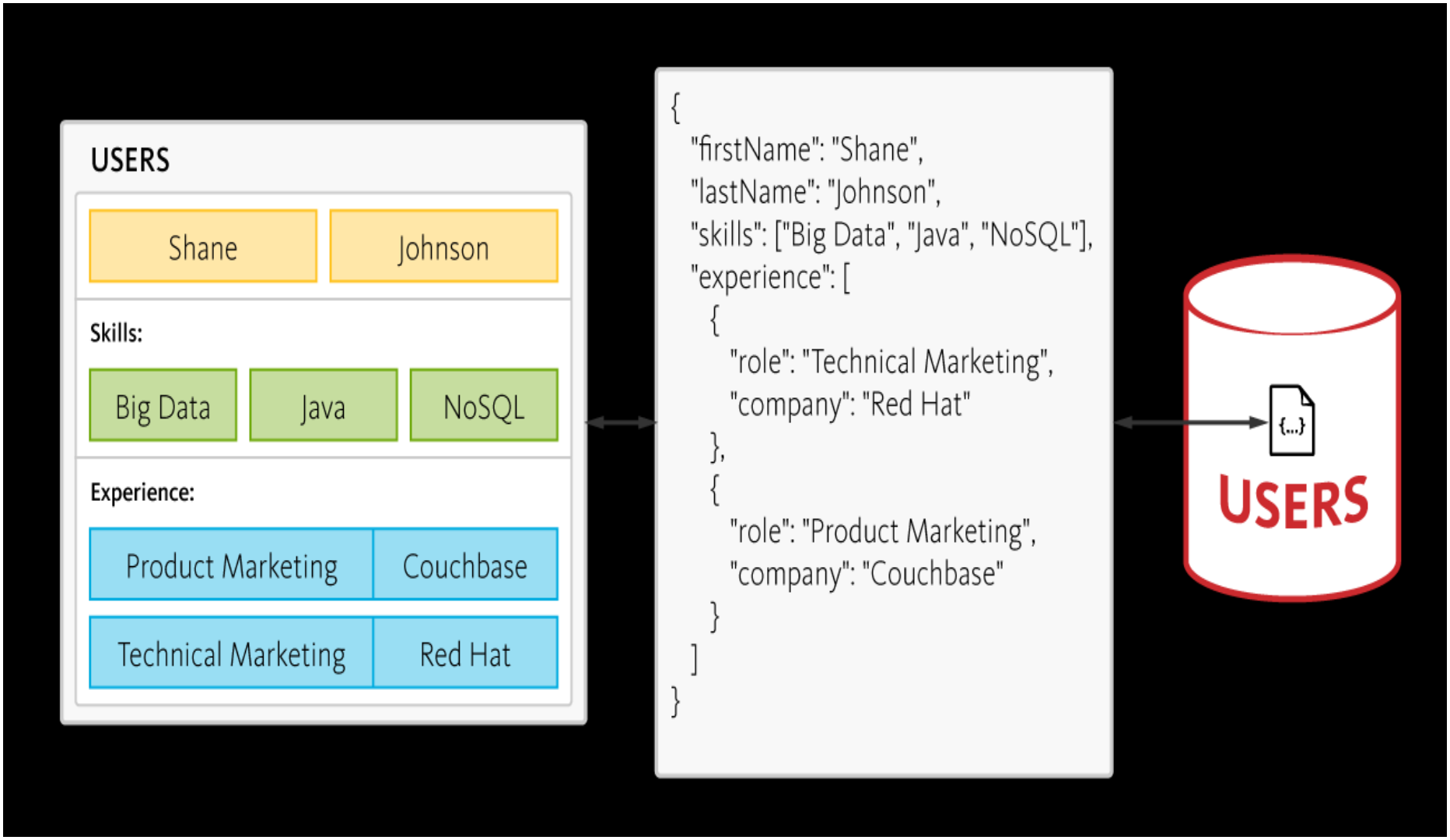
- These are stored as Key-Value pairs

Example (MongoDB)

1. . {_id:101, “Book Name”: ”Fundamentals of business analytics”, “Author Name”: “Seema Acharya”, “Publisher” :”Wiley India”}
2. {_id:102, “Book name”: “Big data and analytics”}

- These are stored as Key-Value pairs

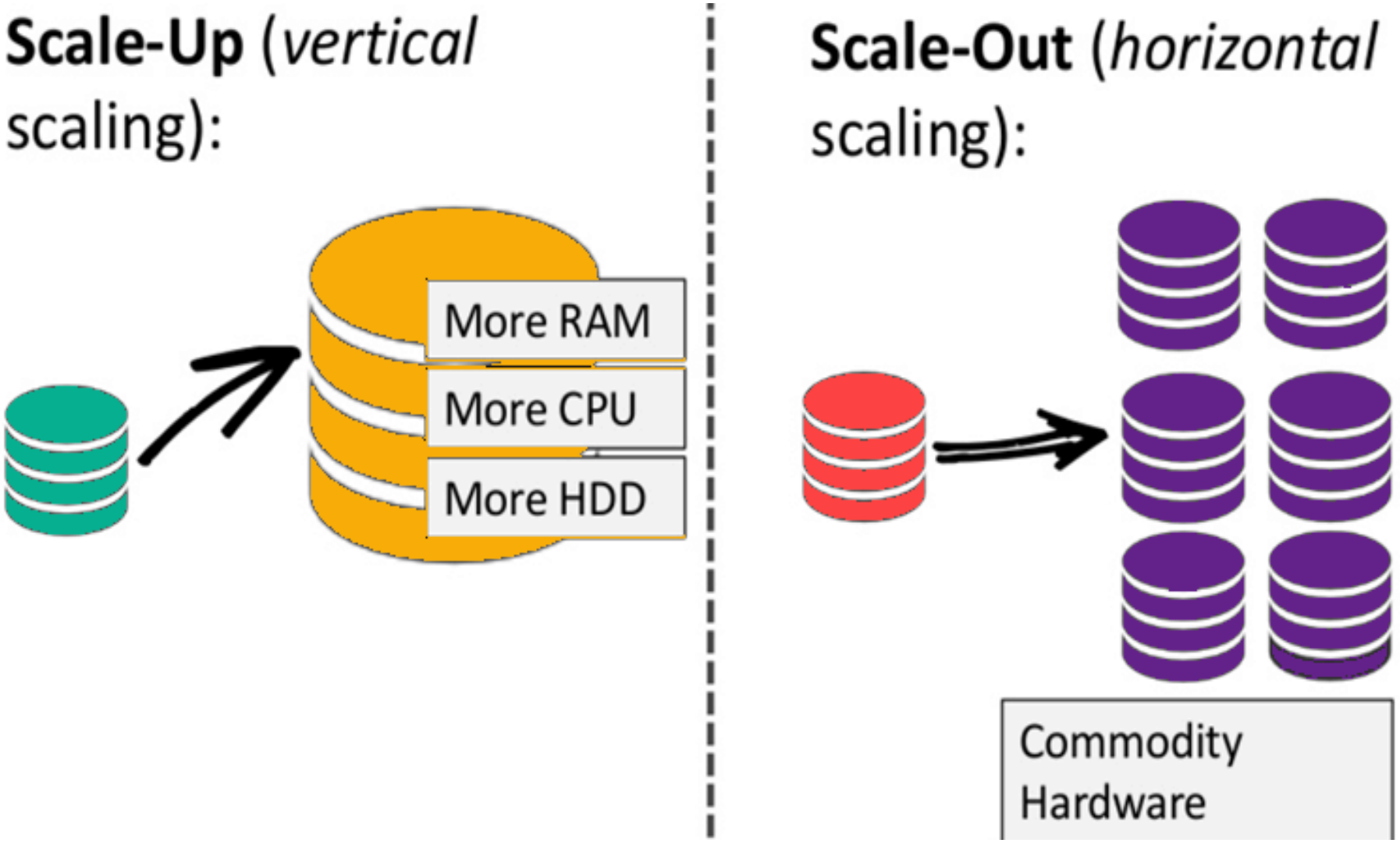
Example (MongoDB)



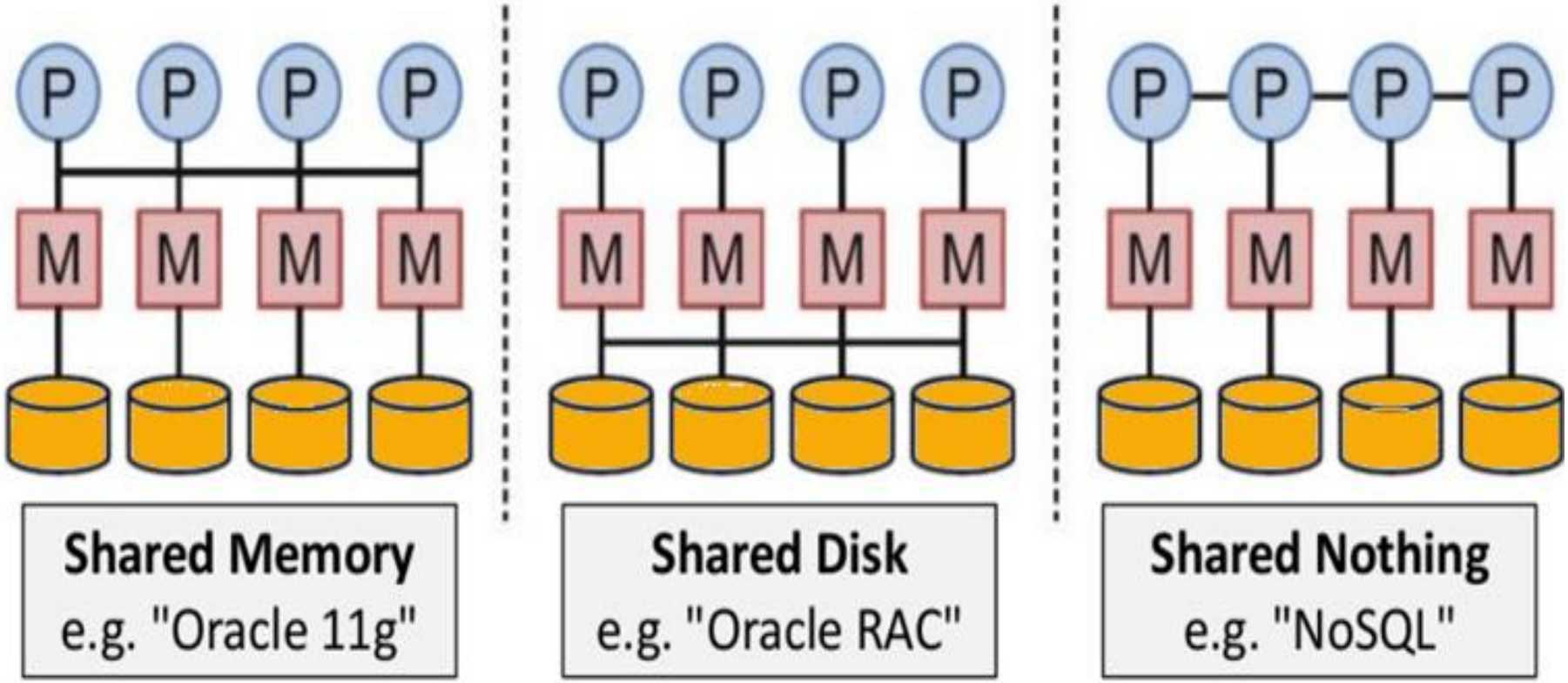
Advantages of NoSQL

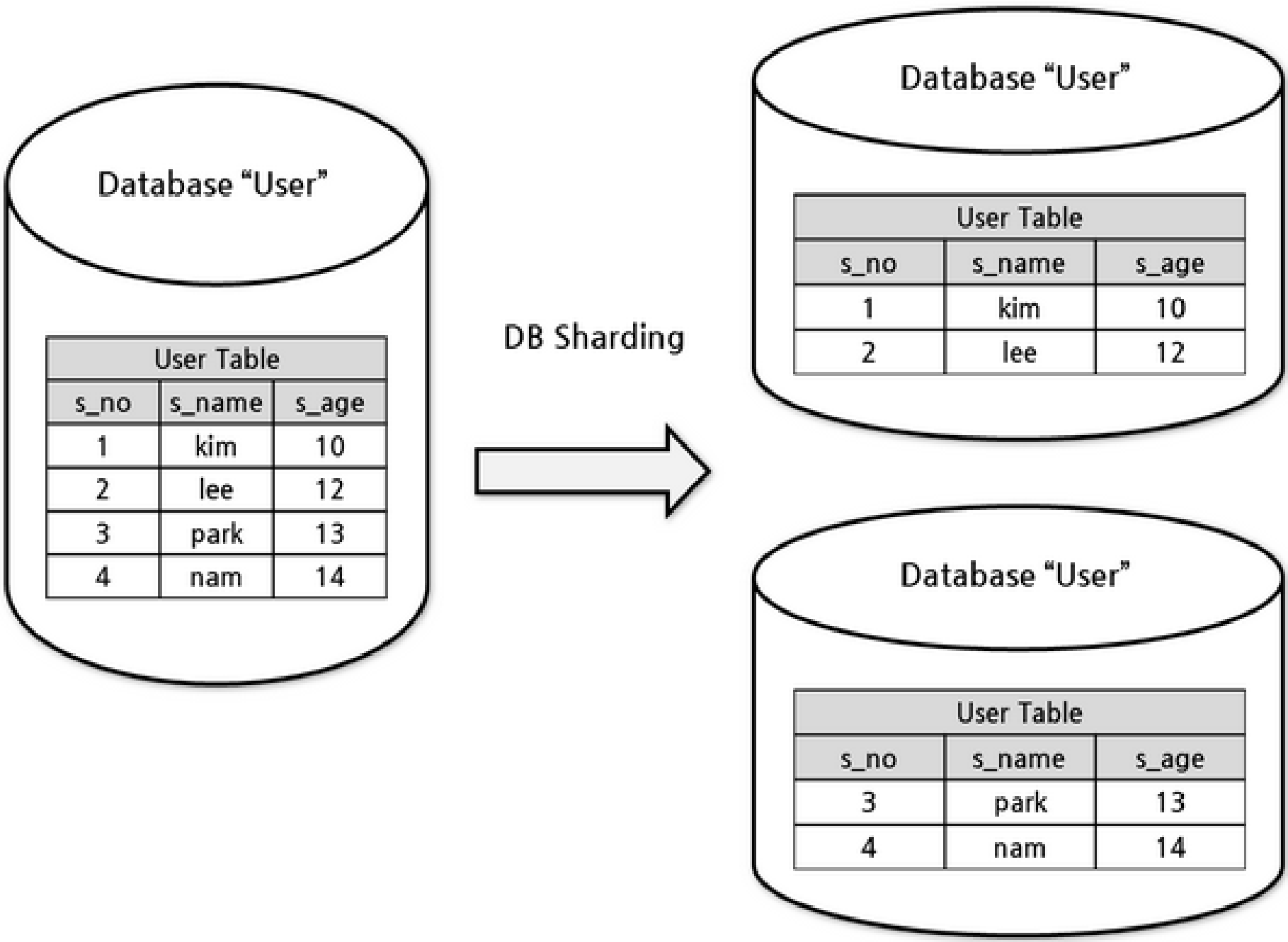
3. Cheap and easy to implement and supports benefits of scale, high availability, fault tolerance in low cost.
4. Relaxes the data consistency requirements: Adopts CAP theorem
5. Data can be replicated on multiple nodes and can be distributed:
   a) Sharding: Automatically spread data across an arbitrary number of servers
   b) Replication: Multiple copies of the data across the cluster.

Difference between Scale-up and Scale-out



NoSQL is Shared Nothing.




Database "User"
User Table
s_no s_name s_age
1 kim 10
2 lee 12
3 park 13
4 nam 14
DB Sharding
Database "User"
User Table
s_no s_name s_age
1 kim 10
2 lee 12
Database "User"
User Table
s_no s_name s_age
3 park 13
4 nam 14

Sharding

- **Sharding** is a type of database partitioning that splits very large databases the into smaller, faster, more easily managed parts called data shards.
- The word **shard** means a small part of a whole.

What we miss with NoSQL

- Does not support Joins.
- Group by
- ACID properties
- Does not have standard SQL interface. But supports MQL and CQL) [M- MongoDB, C- Cassandra]
- Does not support easy integration with other applications that support SQL.

Use of NoSQL in industry

- NoSQL is being used in varied industries. They are used to support analysis for applications such as:
  - Web user data analysis
  - Log analysis
  - Sensor feed analysis
  - Making recommendations for upsell and cross-sell, etc.

NoSQL Vendors

Company	Product	Most widely used by
Amazon	DynamoDB	LinkedIn, Mozilla
Facebook	Cassandra	Netflix, Twitter, eBay
Google	BigTable	Adobe Photoshop

SQL	NoSQL
Relational database	Non-relational, distributed database
Relational model	Model-less approach
Pre-defined schema	Dynamic schema for unstructured data
Table based databases	Document-based or graph-based or wide column store or key-value pairs databases
Vertically scalable (by increasing system resources)	Horizontally scalable (by creating a cluster of commodity machines)
Uses SQL	Uses UnQL (Unstructured Query Language)
Not preferred for large datasets	Largely preferred for large datasets
Not a best fit for hierarchical data	Best fit for hierarchical storage as it follows the key-value pair of storing data similar to JSON (Java Script Object Notation)
Emphasis on ACID properties	Follows Brewer's CAP theorem
Excellent support from vendors	Relies heavily on community support
Supports complex querying and data keeping needs	Does not have good support for complex querying
Can be configured for strong consistency	Few support strong consistency (e.g., MongoDB), few others can be configured for eventual consistency (e.g., Cassandra)
Examples: Oracle, DB2, MySQL, MS SQL, PostgreSQL, etc.	MongoDB, HBase, Cassandra, Redis, Neo4j, CouchDB, Couchbase, Riak, etc.

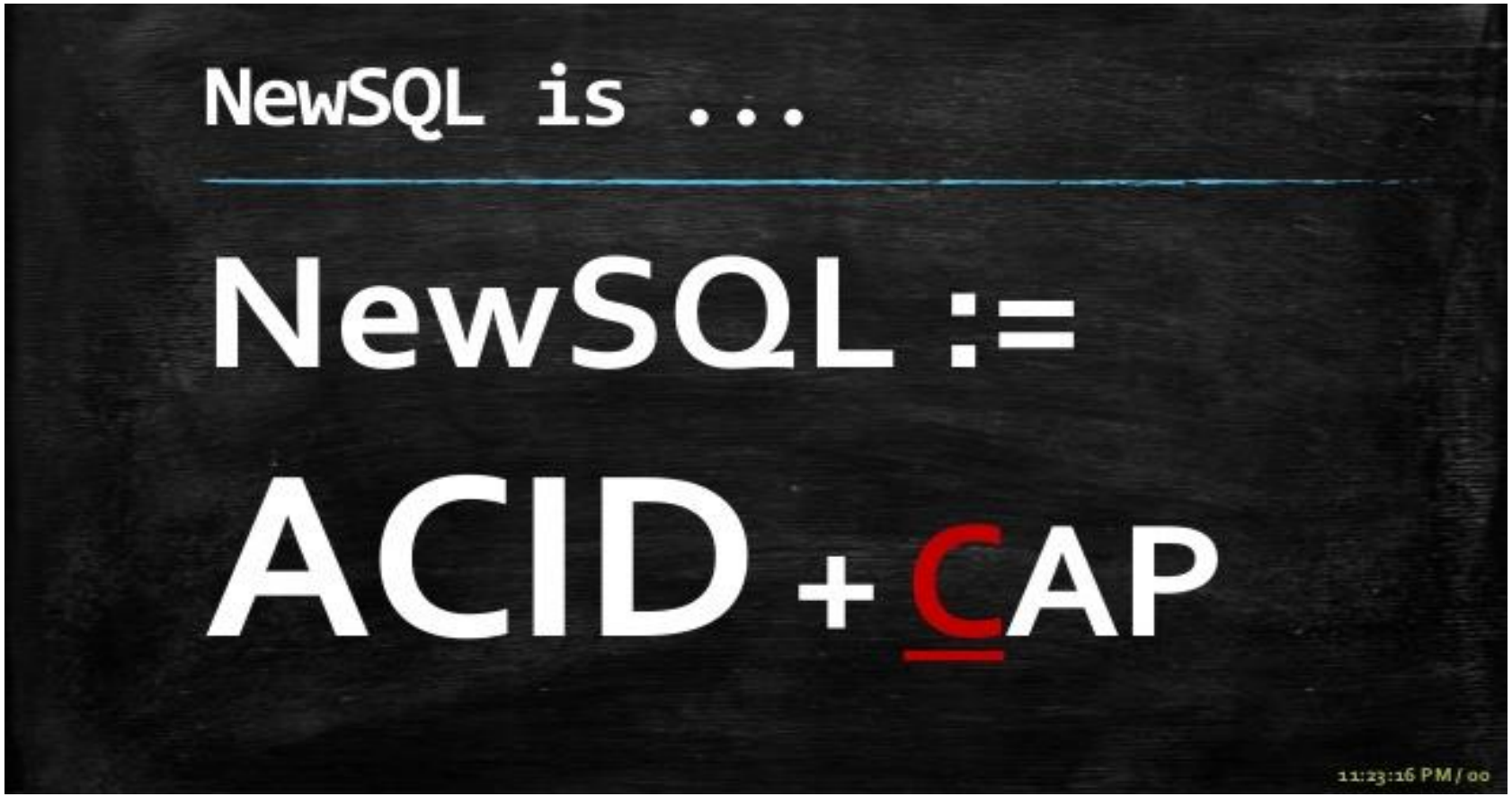
NewSQL

- It is a class of modern relational database management systems that seek to provide the scalable performance of NoSQL systems for online transaction processing (OLTP) while still maintaining the ACID guarantees of a traditional database system.
- NewSQL is a type of database language that incorporates and builds on the concepts and principles of Structured Query Language (SQL) and NoSQL languages.

NewSQL

- By combining the reliability of SQL with the speed and performance of NoSQL, NewSQL provides improved functionality and services.

NewSQL



NewSQL Examples

memsql

xeround

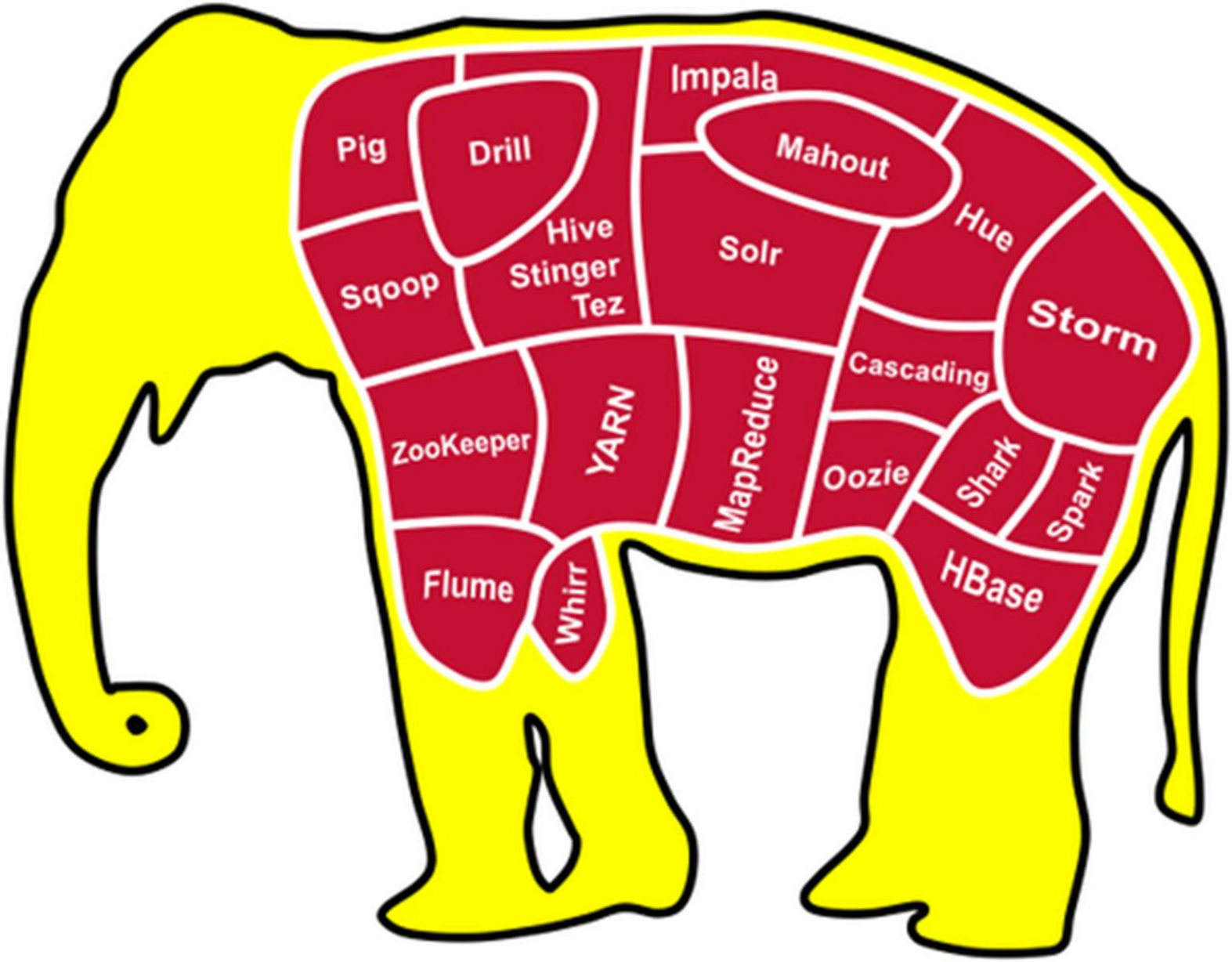
Characteristics of NewSQL

- Based on shared nothing architecture.
- Supports SQL interface for application interaction.
- Supports ACID properties.
- Supports scale out/horizontal scalability.

SQL Vs. NoSQL Vs. NewSQL

	SQL	NoSQL	NewSQL
Adherence to ACID properties	Yes	No	Yes
OLTP/OLAP	Yes	No	Yes
Schema rigidity Adherence to data model	Yes Adherence to relational model	No	Maybe
Data Format Flexibility	No	Yes	Maybe
Scalability	Scale up Vertical Scaling	Scale out Horizontal Scaling	Scale out
Distributed Computing	Yes	Yes	Yes
Community Support	Huge	Growing	Slowly growing

Apache Hadoop Ecosystem



Hadoop

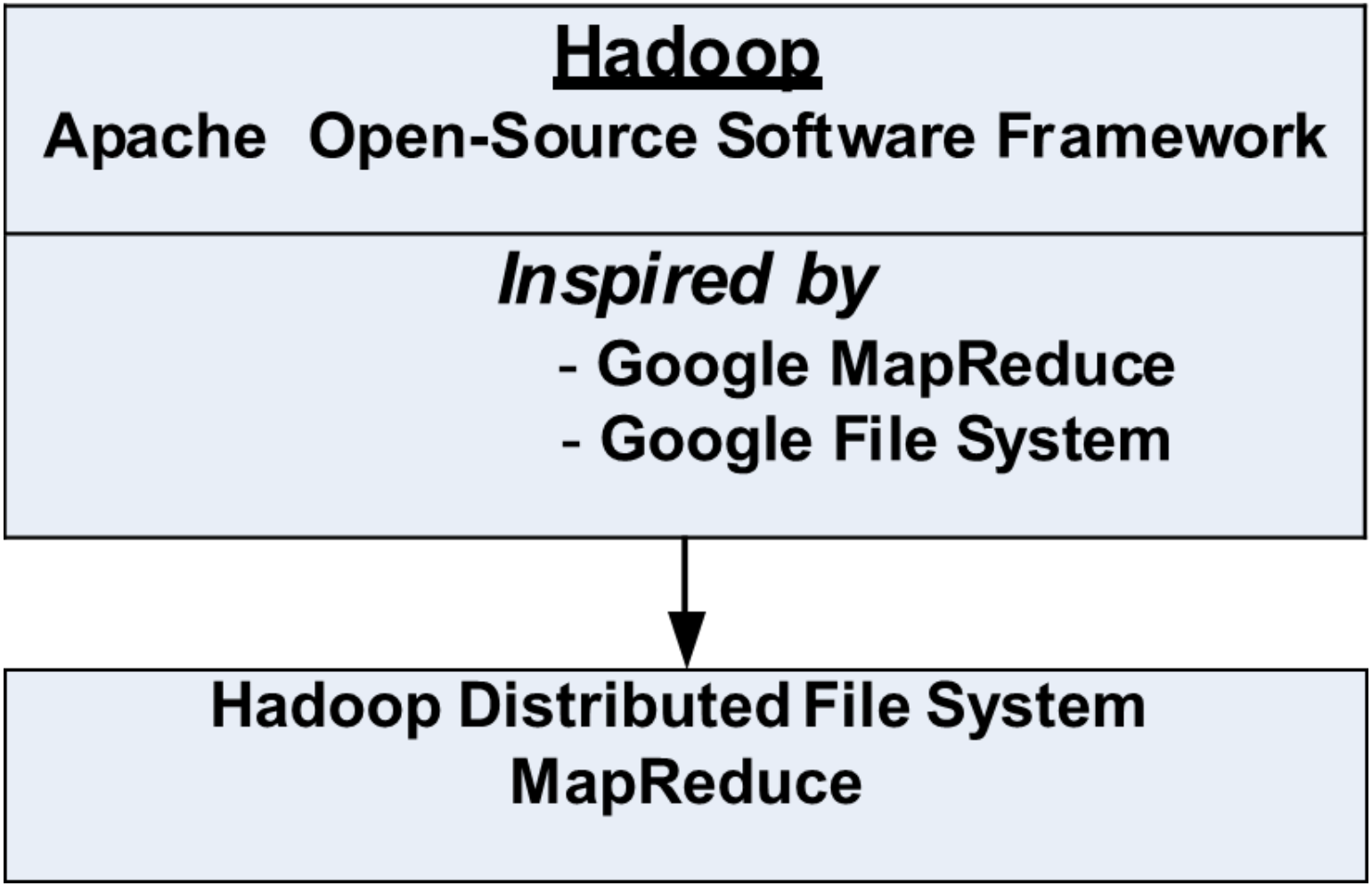
- **Hadoop** is an open source, Java-based programming framework used for storing and processing Big data in a distributed manner on large clusters of commodity hardware.
- It is part of the Apache project sponsored by the Apache Software Foundation.
- **Hadoop** was created by Doug Cutting in 2005, the creator of Apache Lucene, the widely used text search library. He named it Hadoop which is the name of his son's toy elephant.

Hadoop the Elephant

Hadoop

- It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.
- It allows to store and process big data in distributed environment across clusters of computers (1000 /More) .
- It is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Hadoop


Hadoop
Apache Open-Source Software Framework
Inspired by
- Google MapReduce
- Google File System
Hadoop Distributed File System
MapReduce

Features of Hadoop

1. It is optimized to handle massive quantities of SD, SSD and USD using commodity hardware (inexpensive computers).
2. It has shared nothing architecture.
3. It replicates its data across multiple computers.
4. It is for high throughput rather than low latency or response time because handling massive quantities of data is a batch operation.

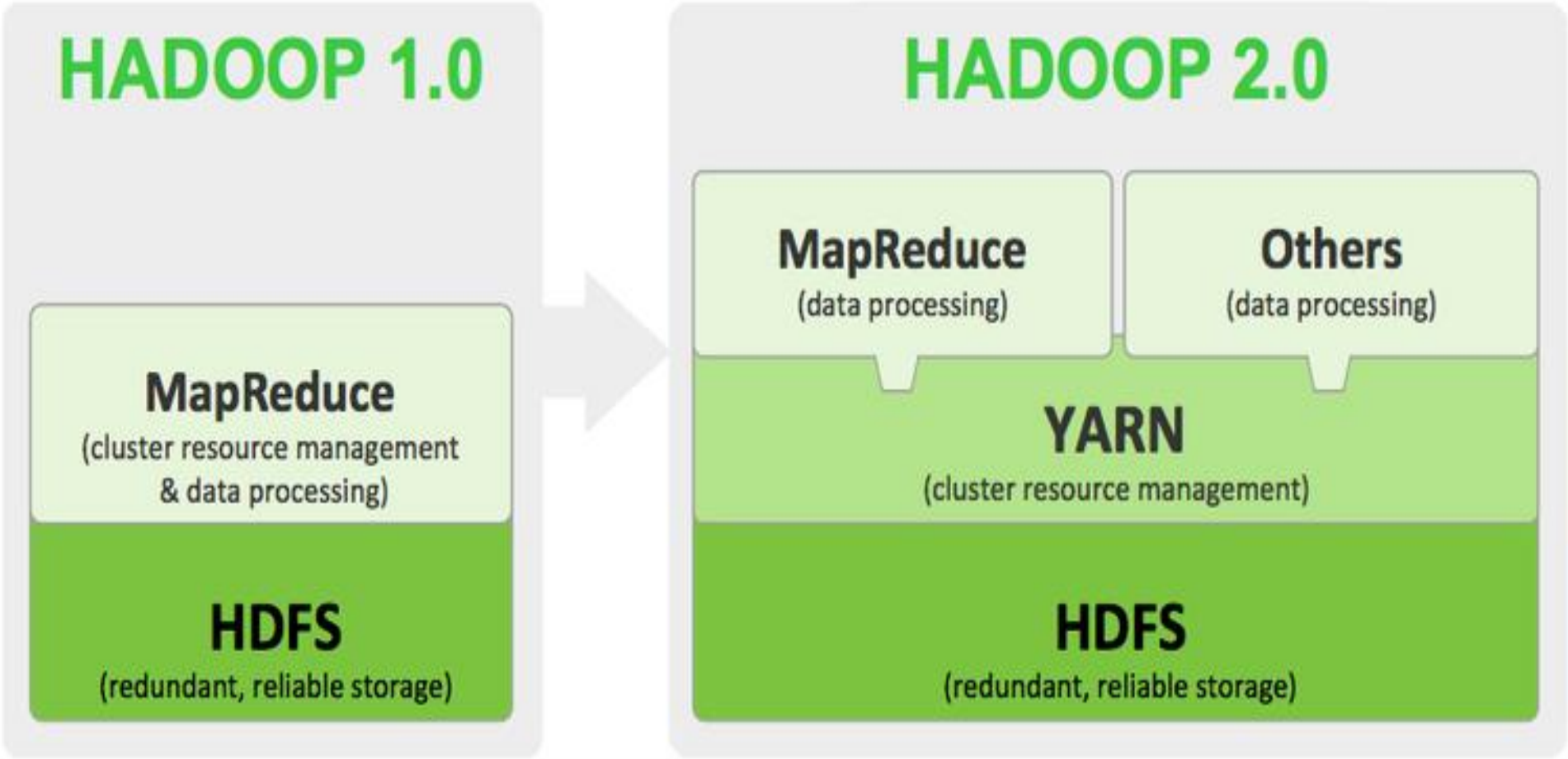
Features of Hadoop

5. It supplements (Complements) OLTP and OLAP.
6. It is not good for non-parallel tasks and processing small files. (Works best for huge data files & data sets).
7. Easy to use
8. Data locality and Reliability
9. Distributed Processing & Storage
10. Robust Ecosystem

Key advantages of Hadoop

1. Stores data in its native format (HDFS).
2. Scalable : store and distribute very large clusters
3. Cost effective: reduced cost/TB of storage and processing.
4. Resilient to failure : Fault tolerant due to data replication on multiple nodes in the cluster.
5. Flexibility: supports any data (SD, SSD and USD) analysis such as email conversations, social media data analysis, click-stream data analysis, log analysis, data mining, market campaign analysis, etc.
6. Fast: move code to data paradigm. [Process Migration]

Versions of Hadoop



YARN -> Yet Another Resource Negotiator

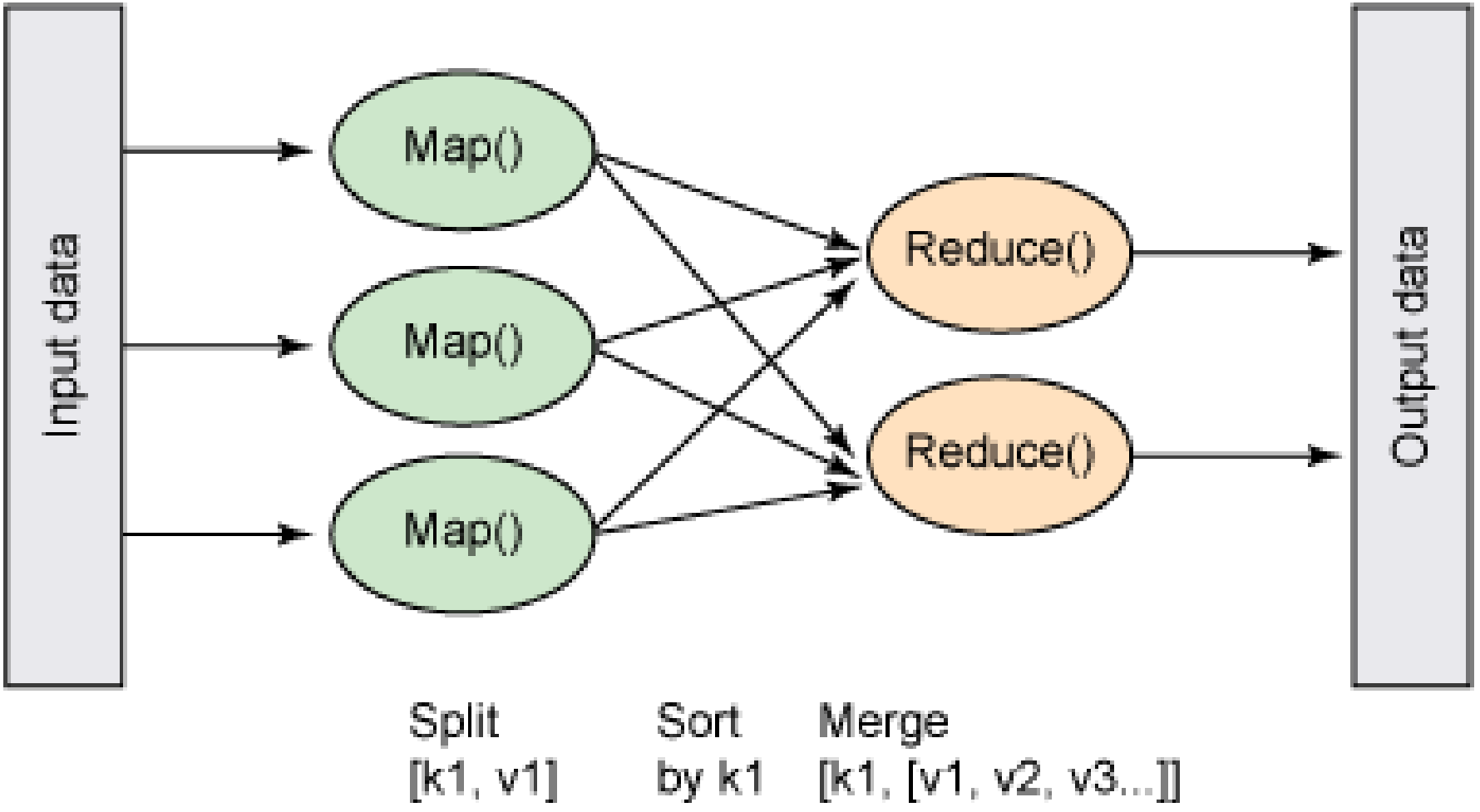
Hadoop 1.0 parts

1. Data storage framework (HDFS):
   - General purpose file system.
   - It is schema-less and stores data files of any format.
   - Stores data files as close to their original format and this provides needed flexibility and agility.
2. Data processing framework:
   - MapReduce model (Google's popular model)
   - Uses two functions: Map and Reduce functions to process the data.

Data processing framework

- The “Mapper” take in set of key-value pairs and generate intermediate data which is another list of key-value pairs.
- The “Reducers” acts on intermediate data and produce the output data.
- The two functions work in isolation from one another, thus enabling the processing to be highly distributed in a highly-parallel, fault tolerant and scalable way.

Data processing framework



Limitations of Hadoop 1.0

1. Requires expertise in MapReduce programming and Java.
2. It supports only batch processing.
3. It is tightly coupled with MapReduce and hence every data for analysis has to be transformed into MapReduce structure.

Hadoop 2.0

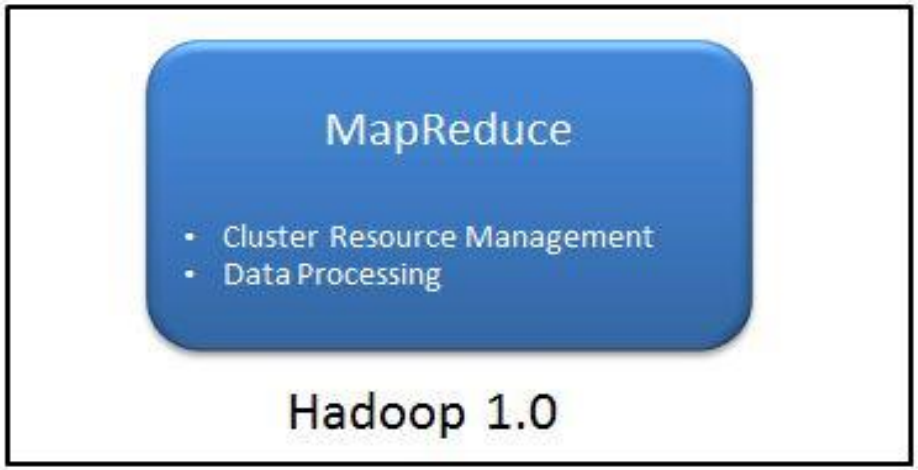
- Apache **Hadoop 2** (**Hadoop 2.0**) is the second iteration of the **Hadoop** framework for distributed data processing.
- **Hadoop 2** adds support for running non-batch applications through the introduction of YARN, a redesigned cluster resource manager that eliminates **Hadoop's** sole reliance on the MapReduce programming model.

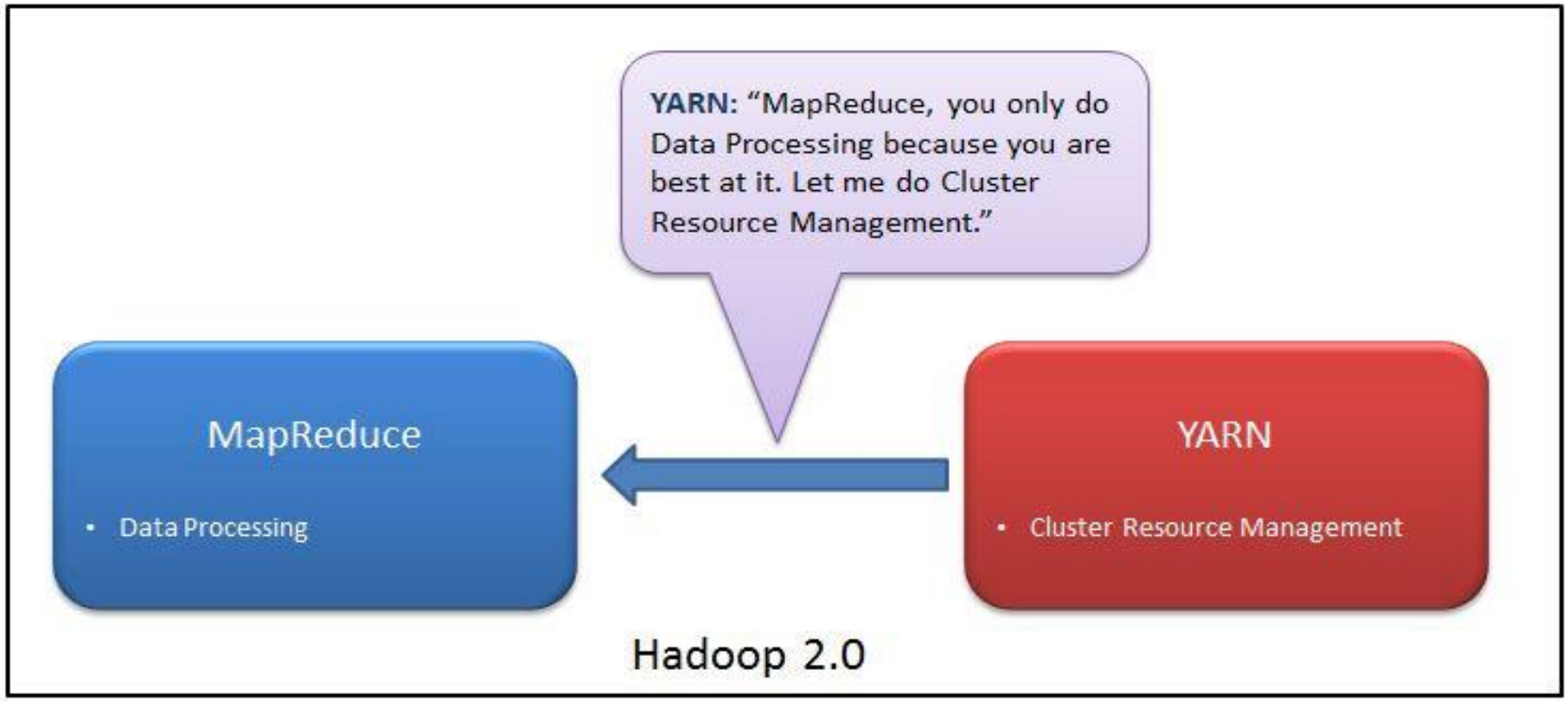
YARN

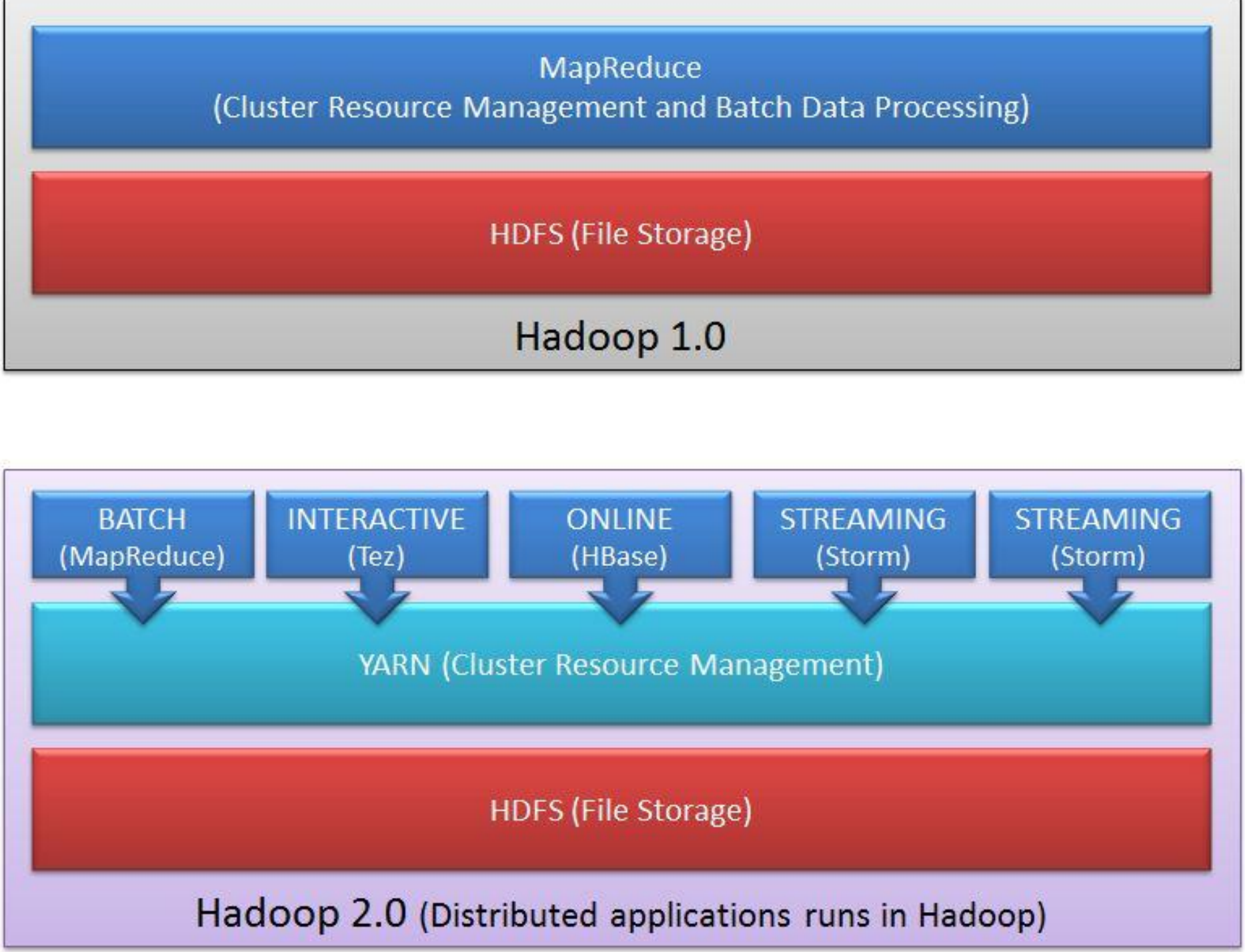
- YARN framework is responsible for Cluster resource management.
- Cluster resource management means managing the resources of the Hadoop Clusters. Resources means Memory, CPU etc.
- YARN took over task of cluster management from MapReduce and MapReduce is streamlined to perform Data Processing only in which it is best.

YARN

- YARN is the brain of Hadoop Ecosystem. It performs all the processing activities by allocating resources and scheduling tasks.
- YARN co-ordinates the allocation of subtasks of the submitted applications, thus enhances flexibility, scalability and efficiency of the applications.


MapReduce
• Cluster Resource Management
• Data Processing
Hadoop 1.0


YARN: "MapReduce, you only do Data Processing because you are best at it. Let me do Cluster Resource Management."
MapReduce
• Data Processing
YARN
• Cluster Resource Management
Hadoop 2.0


MapReduce
(Cluster Resource Management and Batch Data Processing)
HDFS (File Storage)
Hadoop 1.0
BATCH
(MapReduce)
INTERACTIVE
(Tez)
ONLINE
(HBase)
STREAMING
(Storm)
STREAMING
(Storm)
YARN (Cluster Resource Management)
HDFS (File Storage)
Hadoop 2.0 (Distributed applications runs in Hadoop)

Apache Hadoop Ecosystem

Ambari
Provisioning, Managing and Monitoring Hadoop Clusters

Sqoop Data Exchange

Flume Log Collector

Zookeeper Coordination

Oozie Workflow

Pig Scripting

Mahout Machine Learning

R Connectors Statistics

Hive SQL Query

Hbase Columnar Store

YARN Map Reduce v2
Distributed Processing Framework

HDFS
Hadoop Distributed File System

Overview of Hadoop Ecosystem

1. HDFS: It stores different types of large data sets (i.e. structured, unstructured and semi structured data) as close to original form.
2. Hbase (Hadoop's database): HBase is an open source, non-relational distributed database. In other words, it is a NoSQL database.
3. Hive(Facebook): HIVE is a data warehousing component which performs reading, writing and managing large data sets in a distributed environment (Hadoop Cluster) using SQL-like interface. (***HIVE + SQL = HQL)***

Overview of Hadoop Ecosystem

4. Pig: It gives a platform for building data flow for ETL processing and analyzing huge data sets.
   - It is also known as Data Flow language.
   - PIG has two parts: **Pig Latin**, the language and **the pig runtime,** for the execution environment. It is similar to Java and JVM.
   - ***10 line of pig latin = approx. 200 lines of Map-Reduce Java code***
5. ZooKeeper: is the coordinator of any Hadoop job which includes a combination of various services in a Hadoop Ecosystem for distributed applications.

Overview of Hadoop Ecosystem

6. Oozie: clock and alarm (scheduler) service inside Hadoop Ecosystem.
   - It schedules Hadoop jobs and binds them together as one logical work.
7. Mahout: It provides an environment for creating machine learning applications which are scalable.
8. Flume/Chukwa : which helps in storing unstructured and semi-structured data into HDFS.
   - It is data collection system

Overview of Hadoop Ecosystem

9. Sqoop: Import and export structured data from RDBMS or Enterprise data warehouses to HDFS or vice versa.
10. Ambari: It aims at making Hadoop ecosystem more manageable.
    - It is web based tool for **provisioning, managing and monitoring** Apache Hadoop clusters.

Hadoop distributions

- Open source Apache project (free download).
- The core aspects of Hadoop includes:
  - Hadoop Common
  - HDFS
  - Hadoop YARN
  - Hadoop MapReduce


1010data

Actian

amazon
web services

cloudera
hp

Hortonworks

IBM

InfiniDB

INFOBRIGHT

kognitio

MAPR
TECHNOLOGIES

Microsoft

ORACLE

Pivotal

SAP

TERADATA

Hadoop versus SQL

Hadoop	RDBMS
Scale out	Scale Up
Key-Value Pair	Record
MapReduce (Functional Style)	SQL (Declarative)
De-normalized	Normalized
All varieties of Data	Structured Data
OLAP/Batch/Analytical Queries	OLTP/Real time/ Queries

Integrated Hadoop systems

- EMC Greenplum.
- Oracle Big data Appliance
- Microsoft Big data solutions
- IBM InfoSphere
- HP Big data solutions, etc.

Cloud-based Hadoop Solutions

- Amazon Web services (AWS)
- Google BigQuery

AWS

- **Amazon Web Services** offers reliable, scalable, and inexpensive cloud computing services.
- Free to join, pay only for what you use.
- **AWS** is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis with a free-tier option available for 12 months.

Google BigQuery

- Google BigQuery is a cloud-based big data analytics web service for processing very large read-only data sets.
- BigQuery was designed for analyzing data on the order of billions of rows, using a SQL-like syntax. It runs on the Google Cloud Storage infrastructure.


Thank you!