NoSQL Databases

Why NoSQL, Principles, Overview

NoSQL

Lecture 1 - Introduction

Lecture 2 - Document Databases, Data Formats

Lecture 3 - Graph Databases

Lab 1 - MongoDB

Lab2 - Neo4j

Lab3 - Evaluation : **Lab & QCM** on the last session

Professors :

Lectures - Nistor GROZAVU : nistor.grozavu@gmail.com

Labs - Issam FALIH : issam.falih@gmail.com

Lecture plan

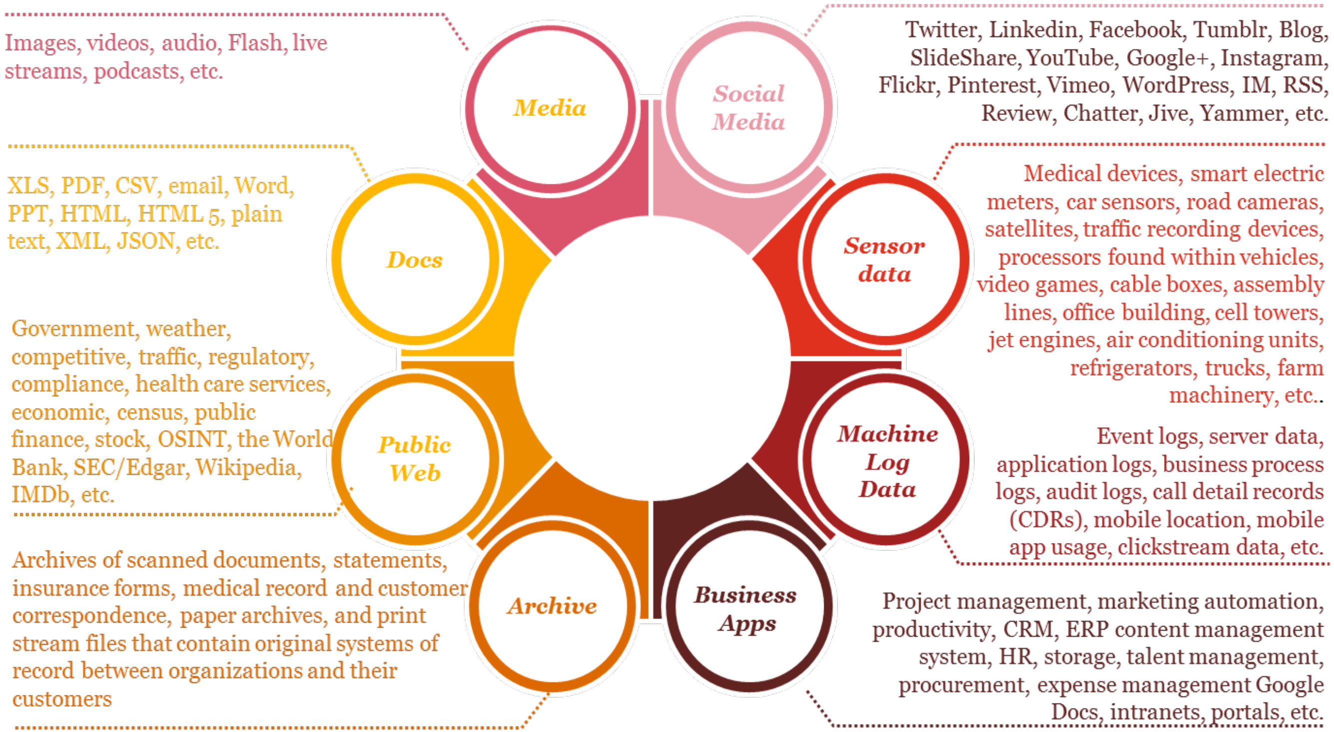
- Current trends in data management & computing
- Big Data
- Relational vs. NoSQL databases
  - the value of relational databases
  - new requirements
  - NoSQL features, strengths and challenges
- Types of NoSQL databases
  - key-value stores, document databases, column-family databases, graph databases
  - principles and examples

Data verses Information

Data is raw, unorganized facts that need to be processed. Data can be something simple and seemingly random and useless until it is organized.

When data is processed, organized, structured or presented in a given context so as to make it useful.

Who’s generating Big Data ?

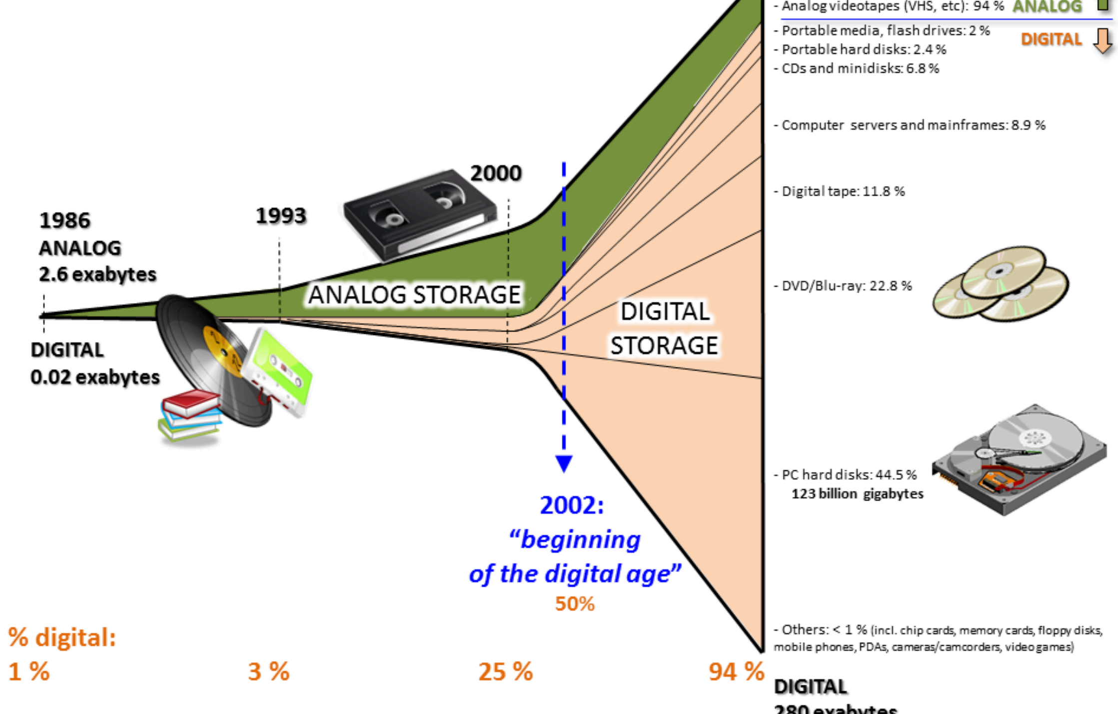


Volume of data (1)

- *The volume of data available to organizations today is on the rise, while the percent of data they can analyze is on the decline.*

Global Information Storage Capacity

in optimally compressed bytes



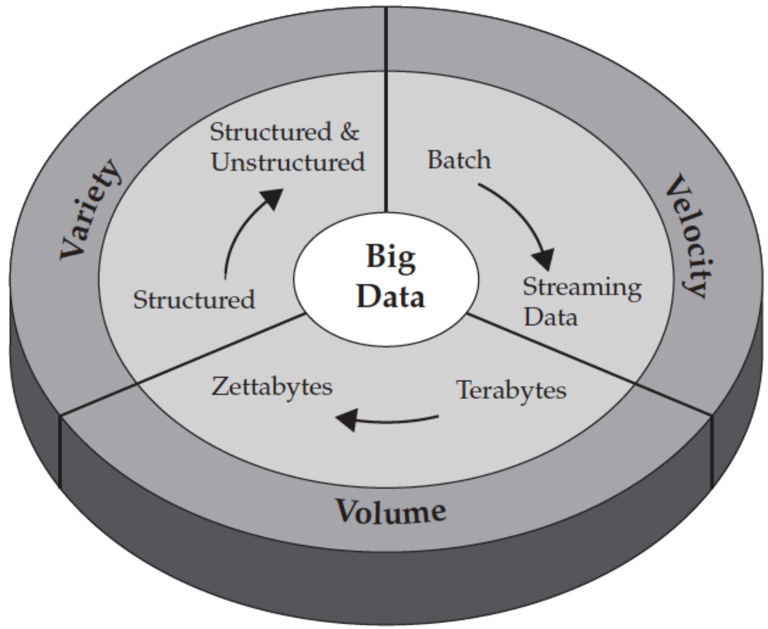
Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

Type of data sources & storage

- Data capture and collection
  - Sensor data, Mobile device, Social Network, Web clickstream,
  - Traffic monitoring, Multimedia content, Smart energy meters,
  - DNA analysis, Industry machines in the age of Internet of Things, Consumer activities – communicating, browsing, buying, sharing, searching – create enormous trails of data.

- Data Storage
  - Cost of storage is reduced tremendously
  - Seagate 3 TB Barracuda @ $149.99 from Amazon.com (4.9¢/GB)

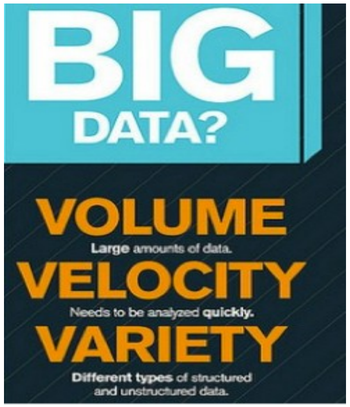
Big Data: 3V

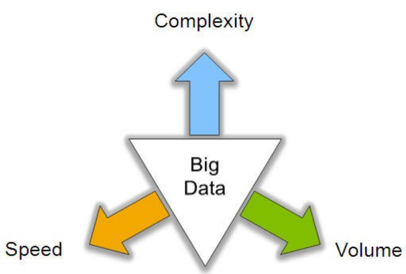


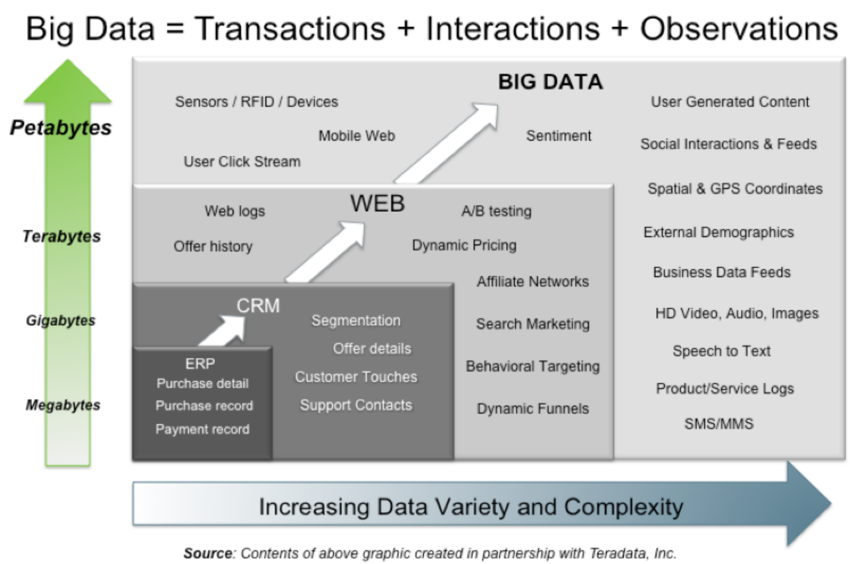
IBM characterizes Big Data by its volume, velocity, and variety—or simply, V^3.

- Gartner’s definition of the 3Vs is still widely used, and in agreement with a consensual definition that states that "Big Data represents the Information assets characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value”. The 3Vs have been expanded to other complementary characteristics of big data:
- Volume: big data doesn't sample. It just observes and tracks what happens
- Velocity: big data is often available in real-time
- Variety: big data draws from text, images, audio, video; plus it completes missing pieces through data fusion
- Machine Learning: big data often doesn't ask why and simply detects patterns
- Digital footprint: big data is often a cost-free byproduct of digital interaction

Big Data Characteristics: 3V







Big Data Characteristics

How big is the Big Data?

- What is big today maybe not big tomorrow

- Any data that can challenge our current technology in some manner can consider as Big Data
  - Volume
  - Communication
  - Speed of Generating
  - Meaningful Analysis

Big Data Vectors (3Vs)

"Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization"
Gartner 2012

Big Data Characteristics

Big Data Vectors (3Vs)

- high-volume
  amount of data

- high-velocity
  Speed rate in collecting or acquiring or generating or processing of data

- high-variety
  different data type such as audio, video, image data (mostly unstructured data)

Big Data: 5V…

- Big data can be described by the following characteristics:
- **Volume -** The quantity of generated data is important in this context. The size of the data determines the value and potential of the data under consideration, and whether it can actually be considered big data or not. The name 'big data' itself contains a term related to size, and hence the characteristic.
- **Variety -** The type of content, and an essential fact that data analysts must know. This helps people who are associated with and analyze the data to effectively use the data to their advantage and thus uphold its importance.
- **Velocity** In this context, the speed at which the data is generated and processed to meet the demands and the challenges that lie in the path of growth and development.
- **Variability** The inconsistency the data can show at times—-which can hamper the process of handling and managing the data effectively.
- **Veracity** The quality of captured data, which can vary greatly. Accurate analysis depends on the veracity of source data. Complexity Data management can be very complex, especially when large volumes of data come from multiple sources. Data must be linked, connected, and correlated so users can grasp the information the data is supposed to convey.

Factory work and Cyber-physical systems may have a 6C system:

- Connection (sensor and networks)
- Cloud (computing and data on demand)
- Cyber (model and memory)
- Content/context (meaning and correlation)
- Community (sharing and collaboration)
- Customization (personalization and value)
- Data must be processed with advanced tools (analytics and algorithms) to reveal meaningful information. Considering visible and invisible issues in, for example, a factory, the information generation algorithm must detect and address invisible issues such as machine degradation, component wear, etc. on the factory floor.

Introduction: Explosion in Quantity of Data

1946	2012
Eniac	LHC

X 6000000 = 1 (40 TB/S)

Air Bus A380

- 1 billion line of code
- each engine generate 10 TB every 30 min

640TB per Flight

Twitter Generate approximately 12 TB of data per day

New York Stock Exchange 1TB of data everyday

---

storage capacity has doubled roughly every three years since the 1980s

Our Data-driven World

- Science
  - Data bases from astronomy, genomics, environmental data, transportation data, …
- Humanities and Social Sciences
  - Scanned books, historical documents, social interactions data, new technology like GPS …
- Business & Commerce
  - Corporate sales, stock market transactions, census, airline traffic, …
- Entertainment
  - Internet images, Hollywood movies, MP3 files, …
- Medicine
  - MRI & CT scans, patient records, …

Our Data-driven World

- Fish and Oceans of Data

What we do with these amount of data?

Ignore

Cost Problem (example)

Cost of processing 1 Petabyte of data with 1000 node ?

1 PB = 10^{15} B = 1 million gigabytes = 1 thousand terabytes

- 9 hours for each node to process 500GB at rate of 15MB/S
- 15*60*60*9 = 486000MB ~ 500 GB
- 1000 * 9 * 0.34$ = 3060$ for single run

- 1 PB = 1000000 / 500 = 2000 * 9 = 18000 h /24 = 750 Day
- The cost for 1000 cloud node each processing 1PB
  2000 * 3060$ = 6,120,000$

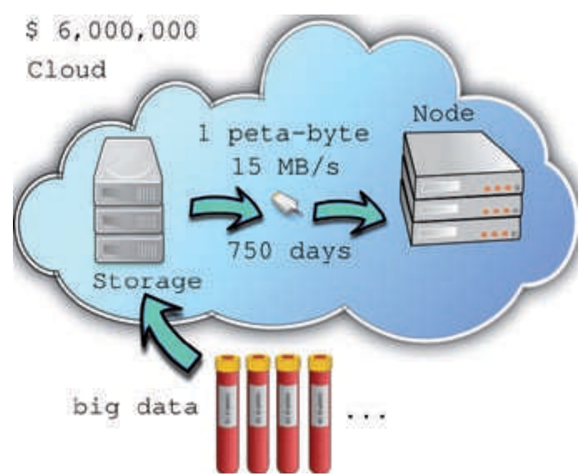


Table 1.1 Contrasting Approaches in Adopting High-Performance Capabilities

Aspect	Typical Scenario	Big Data
Application development	Applications that take advantage of massive parallelism developed by specialized developers skilled in high-performance computing, performance optimization, and code tuning	A simplified application execution model encompassing a distributed file system, application programming model, distributed database, and program scheduling is packaged within Hadoop, an open source framework for reliable, scalable, distributed, and parallel computing
Platform	Uses high-cost massively parallel processing (MPP) computers, utilizing high-bandwidth networks, and massive I/O devices	Innovative methods of creating scalable and yet elastic virtualized platforms take advantage of clusters of commodity hardware components (either cycle harvesting from local resources or through cloud-based utility computing services) coupled with open source tools and technology
Data management	Limited to file-based or relational database management systems (RDBMS) using standard row-oriented data layouts	Alternate models for data management (often referred to as NoSQL or "Not Only SQL") provide a variety of methods for managing information to best suit specific business process needs, such as in-memory data management (for rapid access), columnar layouts to speed query response, and graph databases (for social network analytics)
Resources	Requires large capital investment in purchasing high-end hardware to be installed and managed in-house	The ability to deploy systems like Hadoop on virtualized platforms allows small and medium businesses to utilize cloud-based environments that, from both a cost accounting and a practical perspective, are much friendlier to the bottom line

Big Data services (by Wikibon)

- Wikibon includes the following products and services under the umbrella of Big Data:
- Hadoop software and related hardware and services;
- NoSQL database software and related hardware and services;
- Next-generation data warehouses/analytic database software and related hardware and services;
- Non-Hadoop Big Data platforms, software, and related hardware and services;
- In-memory – both DRAM and flash – databases as applied to Big Data workloads;
- Data integration and data quality platforms, tools and services as applied to Big Data deployments;
- Advanced analytics and data science platforms, tools and services;
- Application development platforms, tools and services as applied to Big Data use cases;
- Business intelligence and data visualization platforms, tools and services as applied to Big Data use cases;
- Analytic and transactional applications and services as applied to Big Data use cases;
- Cloud-based Big Data services including infrastructure, platform and software delivers as a service.
- Other Big Data support, training, and professional services.

Techniques towards Big Data

- Massive Parallelism
- Huge Data Volumes Storage
- Data Distribution
- High-Speed Networks
- High-Performance Computing
- Task and Thread Management
- Data Mining and Analytics
- Data Retrieval
- Machine Learning
- Data Visualization

- ➔ Techniques exist for years to decades. Why did Big
- Data become hot now?
  - More data are being collected and stored
  - Open source code
  - Commodity hardware

Big Data scenario tools

- NoSQL
  - DatabasesMongoDB, CouchDB, Cassandra, Redis, BigTable, Hbase, Hypertable, Voldemort, Riak, ZooKeeper
- MapReduce
  - Hadoop, Hive, Pig, Cascading, Cascalog, mrjob, Caffeine, S4, MapR, Acunu, Flume, Kafka, Azkaban, Oozie, Greenplum
- Storage
  - S3, Hadoop Distributed File System
- Servers
  - EC2, Google App Engine, Elastic, Beanstalk, Heroku
- Processing
  - R, Yahoo! Pipes, Mechanical Turk, Solr/Lucene, ElasticSearch, Datameer, BigSheets, Tinkerpop

Types of Data

- Data can be broadly classified into four types:
    1. Structured Data:
        - Have a predefined model, which organizes data into a form that is relatively easy to store, process, retrieve and manage
        - E.g., relational data

    2. Unstructured Data:
        - Opposite of structured data
        - E.g., Flat binary files containing text, video or audio
        - Note: data is not completely devoid of a structure (e.g., an audio file may still have an encoding structure and some metadata associated with it)

Types of Data

- Data can be broadly classified into four types:
  3. Dynamic Data:
     - Data that changes relatively frequently
     - E.g., office documents and transactional entries in a financial database
  4. Static Data:
     - Opposite of dynamic data
     - E.g., Medical imaging data from MRI or CT scans

Why Classifying Data?

- Segmenting data into one of the following 4 quadrants can help in designing and developing a pertaining storage solution

	Dynamic	Static
Unstructured	Media Production, eCAD, mCAD, Office Docs	Media Archive, Broadcast, Medical Imaging
Structured	Transaction Systems, ERP, CRM	BI, Data Warehousing

- Relational databases are usually used for structured data

- File systems or *NoSQL databases* can be used for (static), unstructured data (*more on these later*)

Processing (Traditional) Data

- OLTP: Online Transaction Processing (DBMSs)
  - Database applications
  - Storing, querying, multi-user access
- OLAP: Online Analytical Processing (Warehousing)
  - Answer multi-dimensional analytical queries
  - Financial/marketing reporting, budgeting, forecasting, …
- RTAP: Real-Time Analytic Processing (Big Data Architecture & Technology)
  - Data gathered & processed in real-time (streaming)
  - Real-time and history data combined

Technologies for Big Data

- Distributed file systems (GFS, HDFS, etc.)
- MapReduce
  - and other models for distributed programming
- NoSQL databases
- Data Warehouses
- Grid computing, cloud computing
- Large-scale machine learning

Relational Database Management Systems

- RDBMS are predominant database technologies
  - first defined in 1970 by Edgar Codd of IBM's Research Lab
- Data modeled as relations (tables)
  - object = tuple of attribute values
    - each attribute has a certain domain
  - a table is a set of objects (tuples, rows) of the same type
    - relation is a subset of cartesian product of the attribute domains
  - tables and objects “interconnected” via (foreign) keys
    - field (or a set of fields) that uniquely identifies a row in another table
- Relational calculus, SQL query language

RDBMS Example

Students

ID#	Name	Phone	DOB
500	Matt	555-4141	06/03/70
501	Jenny	867-5309	3/15/81
502	Sean	876-9123	10/31/82

ID#	ClassID	Sem
500	1001	Fall02
501	1002	Fall02
501	1002	Spr03
502	1003	S203

Takes_Course

ClassID	Title	ClassNum
1001	Intro to Informatics	I101
1002	Data Mining	I400
1003	Internet and Society	I400

Courses

SELECT Name **FROM** Students **NATURAL JOIN** Takes_Course **WHERE** ClassID = 1001

sources: https://github.com/talhafazal/DataBase/wiki/Home-Work-%23-3-Relational-Data-vs-Non-Relational-Database...

The Value of Relational Databases

- A (mostly) standard data model
- Many well developed technologies
  - physical organization of the data, search indexes, query optimization, search operator implementations
- Good concurrency control (ACID)
  - transactions: atomicity, consistency, isolation, durability
- Many reliable integration mechanisms
  - “shared database integration” of applications
- Well-established: familiar, mature, support,...

Data Management: Trends & Requirements

Trends

- **Volume** of data
- **Cloud** comp. (IaaS)
- **Velocity** of data
- **Big** users
- **Variety** of data

Requirements

- Real database **scalability**
  - massive database distribution
  - dynamic resource management
  - horizontally scaling systems
- Frequent **update** operations
- Massive **read** throughput
- **Flexible** database schema
  - semi-structured data

RDBMS for Big Data

- relational schema
  - data in tuples
  - a priori known schema
- schema normalization
  - data split into tables (3NF)
  - queries merge the data
- transaction support
  - trans. management with ACID
  - Atomicity, Consistency, Isolation, Durability
  - safety first

- but current data are naturally flexible
- inefficient for large data
- slow in distributed environment
- full transactions very inefficient in distributed envir.

NoSQL Databases

- What is "NoSQL"?
    - term used in late 90s for a different type of technology: Carlo Strozzi: http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/
    - "Not Only SQL"?
        - but many RDBMS are also "not just SQL"

"NoSQL is an accidental term with no precise definition"

- first used at an informal meetup in 2009 in San Francisco (presentations from Voldemort, Cassandra, Dynomite, HBase, Hypertable, CouchDB, and MongoDB)

[Sadalage & Fowler: NoSQL Distilled, 2012]

NoSQL Databases (cont.)

- NoSQL: Database technologies that are (mostly):
  - Not using the relational model (nor the SQL language)
  - Designed to run on large clusters (horizontally scalable)
  - No schema - fields can be freely added to any record
  - Open source
  - Based on the needs of 21st century web estates

[Sadalage & Fowler: NoSQL Distilled, 2012]

- Other characteristics (often true):
  - easy replication support (fault-tolerance, query efficiency)
  - simple API
  - eventually consistent (not ACID)

Just Another Temporary Trend?

- There have been other trends here before
  - object databases, XML databases, etc.

- But NoSQL databases:
  - are answer to real practical problems big companies have
  - are often developed by the biggest players
  - outside academia but based on solid theoretical results
    - e.g. old results on distributed processing
  - widely used

NoSQL Properties in Detail

1. Flexible scalability
   - horizontal scalability instead of vertical
2. Dynamic schema of data
   - different levels of flexibility for different types of DB
3. Efficient reading
   - spend more time storing the data, but read fast
   - keep relevant information together
4. Cost saving
   - designed to run on commodity hardware
   - typically open-source (with a support from a company)

Challenges of NoSQL Databases

1. Maturity of the technology
    - it’s getting better, but RDBMS had a lot of time
2. User support
    - rarely professional support as provided by, e.g. Oracle
3. Administration
    - massive distribution requires advanced administration
4. Standards for data access
    - RDBMS have SQL, but the NoSQL world is more wild
5. Lack of experts
    - not enough DB experts on NoSQL technologies

...but

More and more companies accept the weak points and choose NoSQL databases for their strengths.
NoSQL technologies are also often used as secondary databases for specific data processing.

http://basho.com/about/customers/

https://www.mongodb.com/who-uses-mongodb

http://planetcassandra.org/companies/

http://neo4j.com/customers/

The End of Relational Databases?

- Relational databases are not going away
  - are ideal for a lot of structured data, reliable, mature, etc.
- RDBMS became one option for data storage

Polyglot persistence – using different data stores in different circumstances
[Sadalage & Fowler: NoSQL Distilled, 2012]

Two trends:

1. NoSQL databases implement standard RDBMS features
2. RDBMS are adopting NoSQL principles

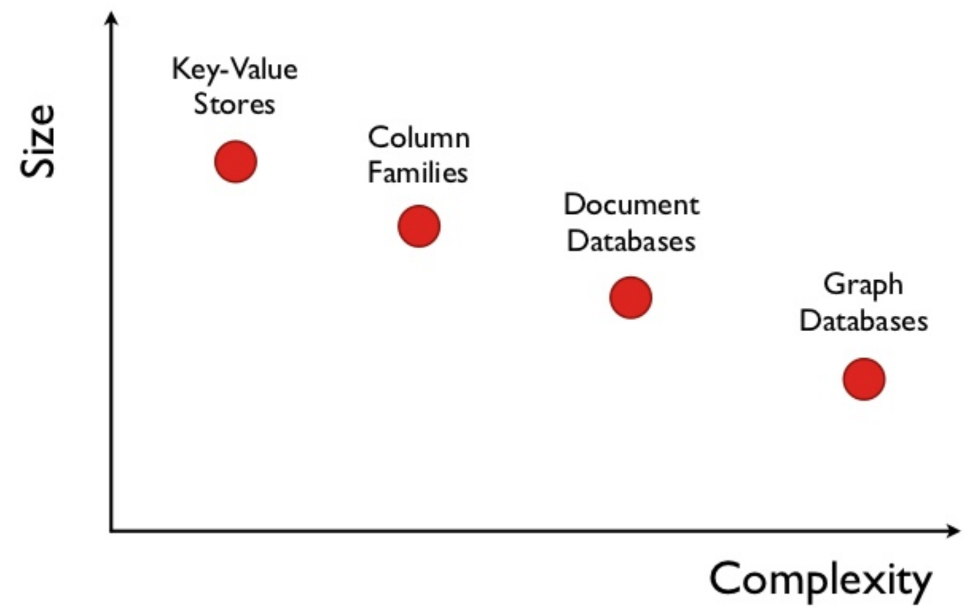
NoSQL Databases

- To this end, a new class of databases emerged, which mainly follow the BASE properties
  - These were dubbed as NoSQL databases
  - E.g., Amazon's Dynamo and Google's Bigtable
- Main characteristics of NoSQL databases include:
  - No strict schema requirements
  - No strict adherence to ACID properties
  - Consistency is traded in favor of Availability

NoSQL Technologies

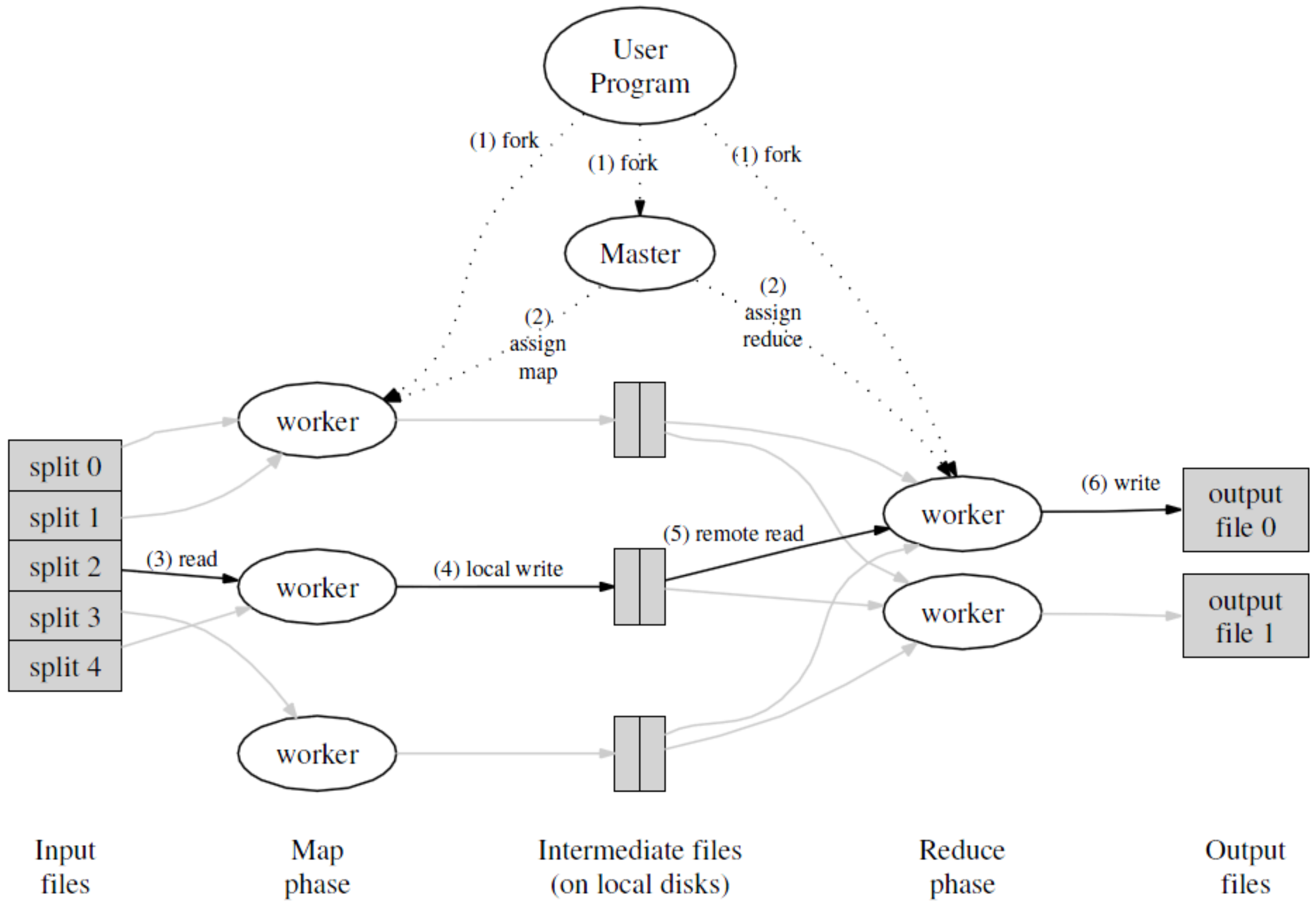
- MapReduce programming model
  - running over a distributed file system

- Key-value stores
- Document databases
- Column-family stores
- Graph databases



source: http://www.slideshare.net/emileifrem/nosql-east-a-nosql-overview-and-the-benefits-of-graph-databases

MapReduce: Principles



source: Dean, J. & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters

MapReduce: Features

- MapReduce is a generic approach for distributed processing of large data collections

- Requires a way to distribute the data
  - and to collect the results back after the processing

- The user must only specify two functions:
  map & **reduce**

MapReduce: Implementation

hadoop MapReduce

Spark

mongoDB

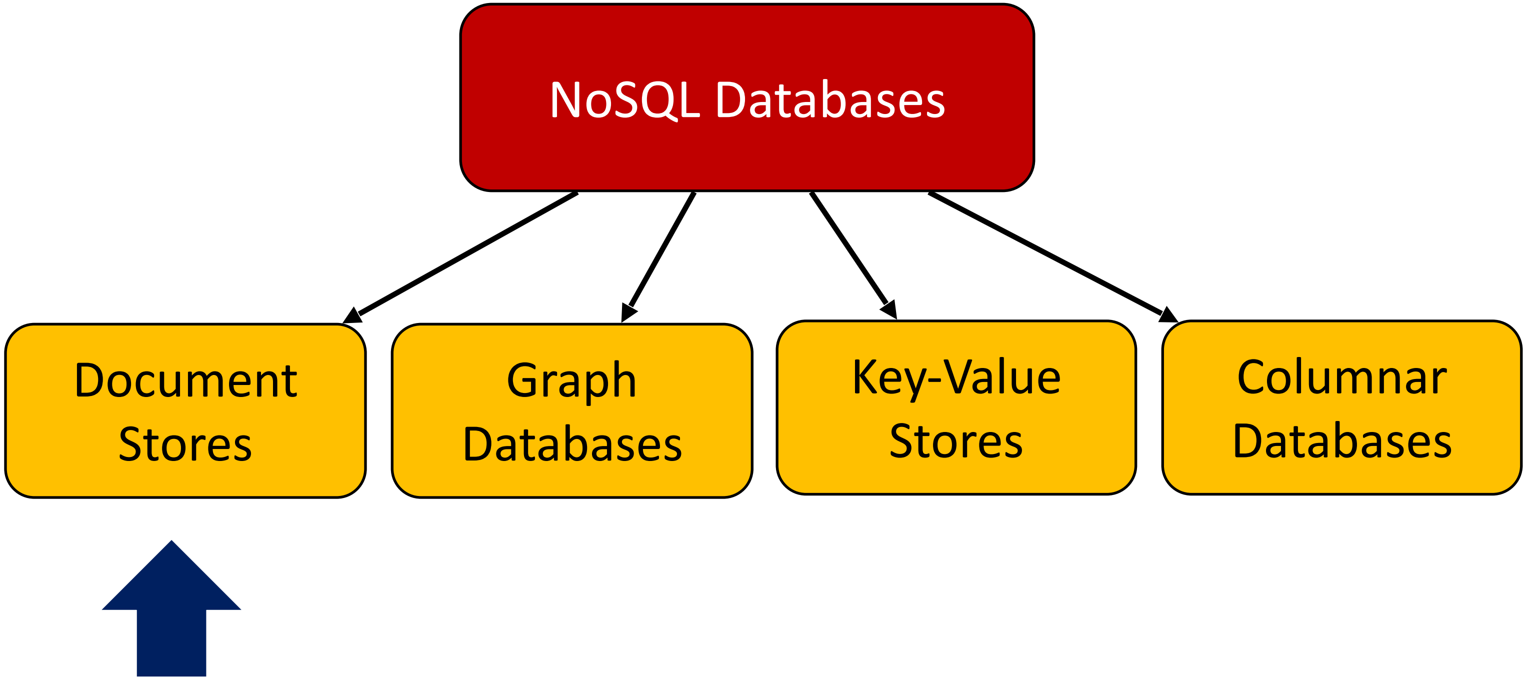
Amazon Elastic MapReduce

riak

Cassandra

Types of NoSQL Databases

- Here is a limited taxonomy of NoSQL databases:



Document Stores

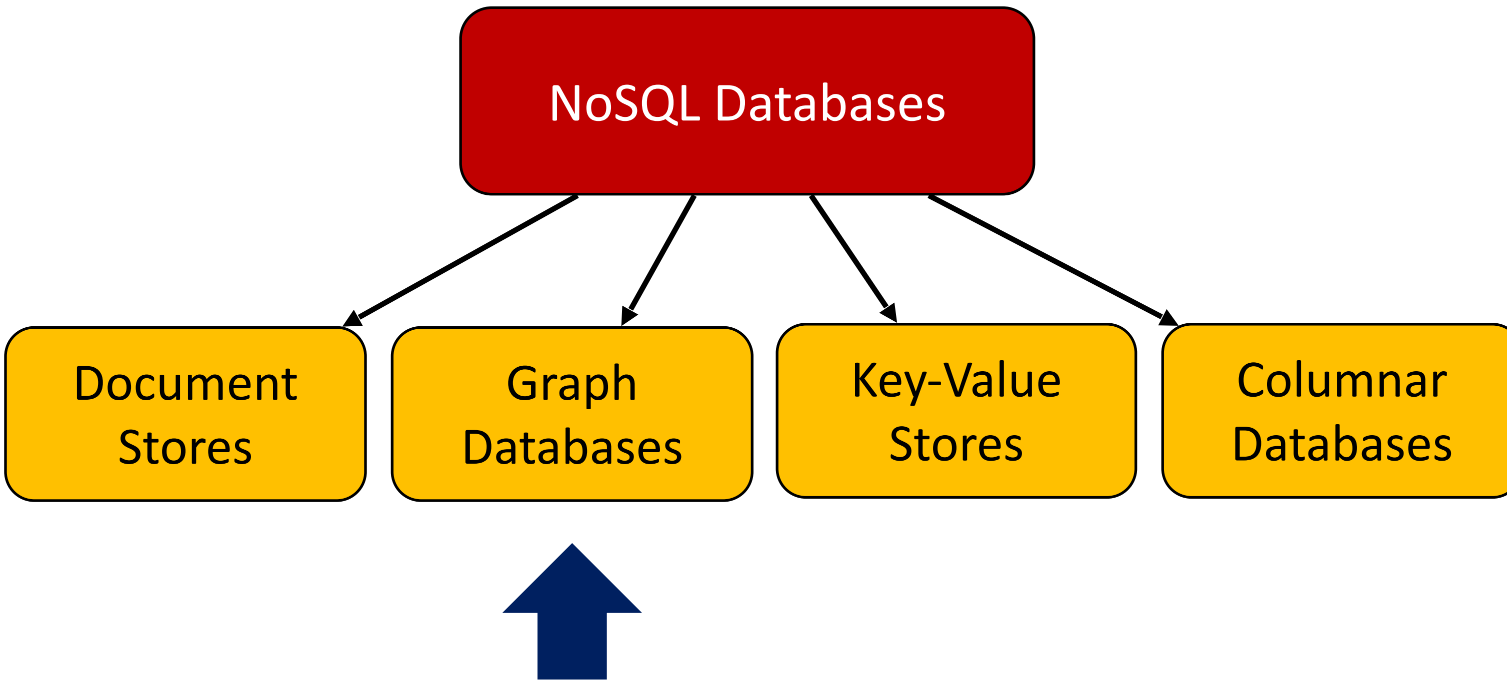
- Documents are stored in some standard format or encoding (e.g., XML, JSON, PDF or Office Documents)
  - These are typically referred to as Binary Large Objects (BLOBs)

- Documents can be indexed
  - This allows document stores to outperform traditional file systems

- E.g., MongoDB and CouchDB (both can be queried using MapReduce)

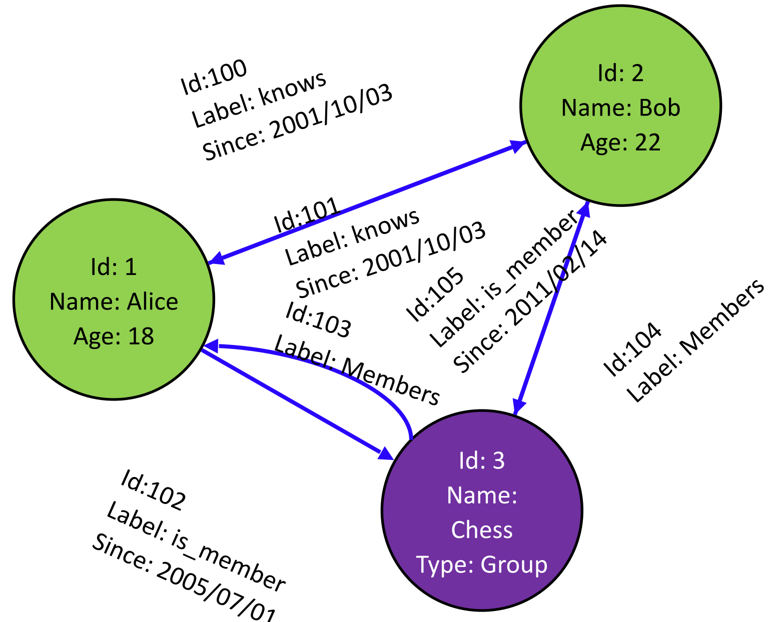
Types of NoSQL Databases

- Here is a limited taxonomy of NoSQL databases:



Graph Databases

- Data are represented as vertices and edges

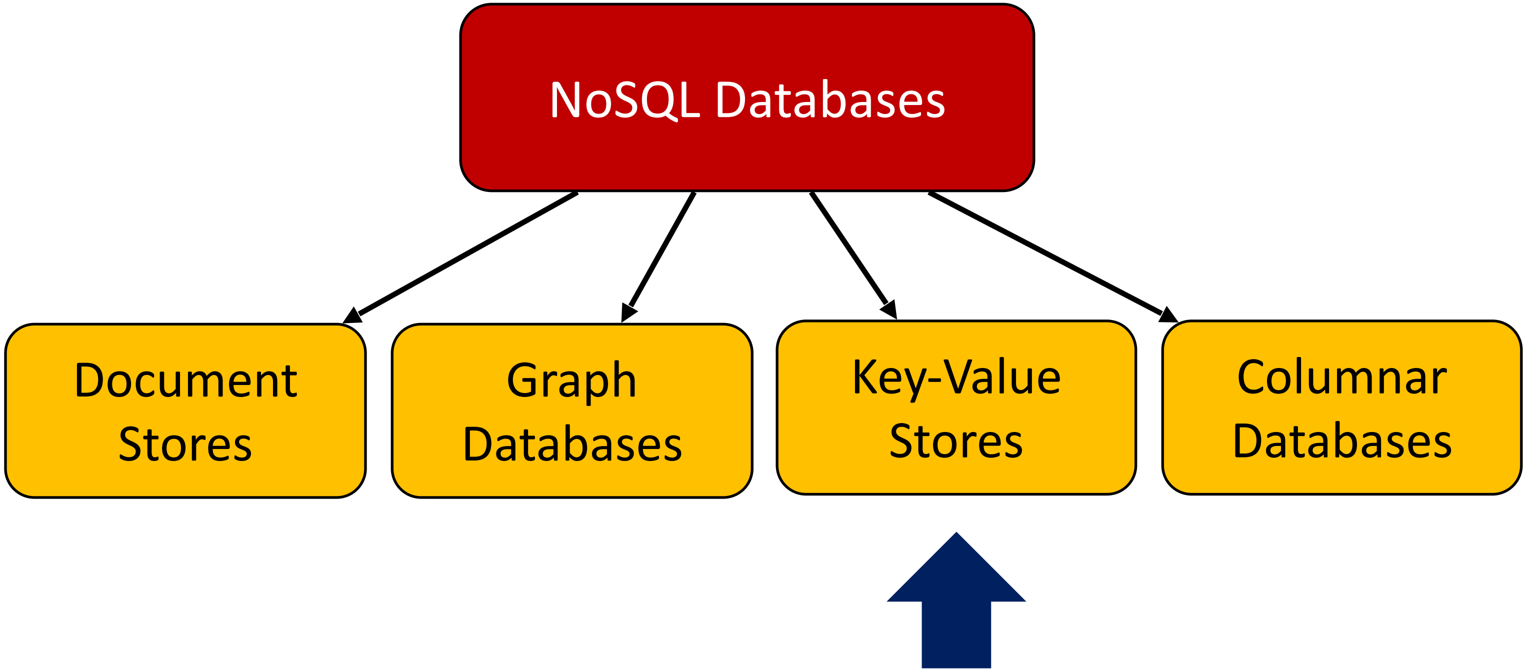


- Graph databases are powerful for graph-like queries (e.g., find the shortest path between two elements)

- E.g., Neo4j and VertexDB

Types of NoSQL Databases

- Here is a limited taxonomy of NoSQL databases:



Key-Value Stores

- Keys are mapped to (possibly) more complex value (e.g., lists)

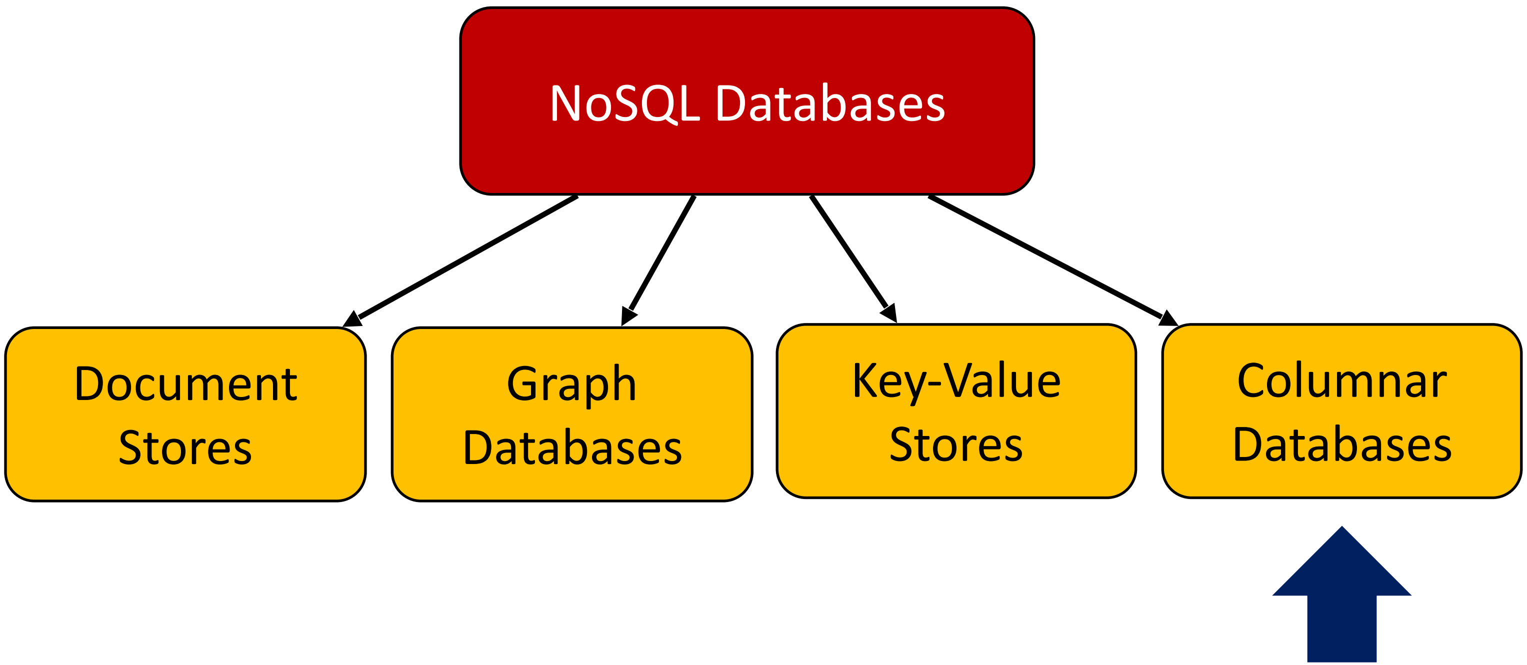
- Keys can be stored in a hash table and can be distributed easily

- Such stores typically support regular CRUD (create, read, update, and delete) operations
    - That is, no joins and aggregate functions

- E.g., Amazon DynamoDB and Apache Cassandra

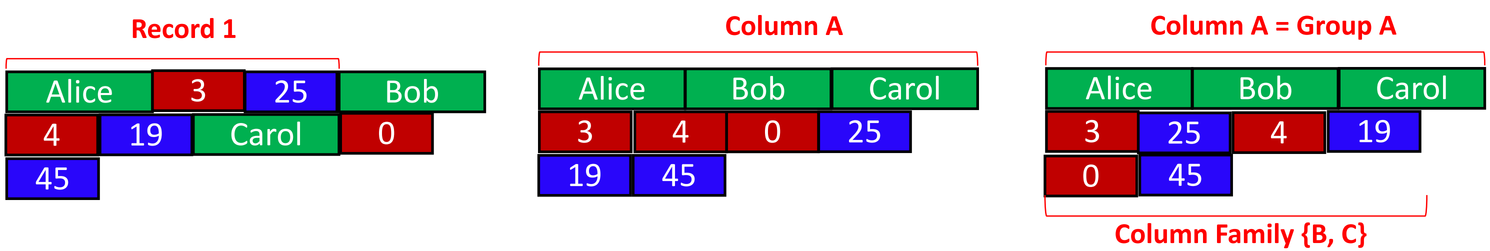
Types of NoSQL Databases

- Here is a limited taxonomy of NoSQL databases:



Columnar Databases

- Columnar databases are a hybrid of RDBMSs and Key-Value stores
  - Values are stored in groups of zero or more columns, but in Column-Order (as opposed to Row-Order)



Row-Order | *Columnar (or Column-Order)* | *Columnar with Locality Groups*

  - Values are queried by matching keys

- E.g., HBase and Vertica

Summary

- Data can be classified into 4 types, *structured*, *unstructured*, *dynamic* and *static*
- Different data types usually entail different database designs
- Databases can be scaled *up* or *out*
- The *2PC protocol* can be used to ensure strict consistency
- Strict consistency limits scalability

Summary (*Cont'd*)

- The *CAP theorem* states that any distributed database with shared data can have at most two of the three desirable properties:
  - **<u>C</u>**onsistency
  - **<u>A</u>**vailability
  - **<u>P</u>**artition Tolerance

- The CAP theorem lead to various designs of databases with *relaxed* ACID guarantees

Summary (*Cont'd*)

- *NoSQL* (or *Not-Only-SQL*) databases follow the *BASE properties*:
  - **B**asically **A**vailable
  - **S**oft-State
  - **E**ventual Consistency

- NoSQL databases have different types:
  - Document Stores
  - Graph Databases
  - Key-Value Stores
  - Columnar Databases

Questions?

Please, any questions?

References

- I. Holubová, J. Kosek, K. Minařík, D. Novák. Big Data a NoSQL databáze. Praha: Grada Publishing, 2015. 288 p.
- Sadalage, P. J., & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley Professional, 192 p.
- RNDr. Irena Holubova, Ph.D. MMF UK course NDBI040: Big Data Management and NoSQL Databases
- *Why NoSQL.* White paper. http://www.couchbase.com/
- http://db-engines.com/en/ranking
- http://nosql-database.org/
- Chang, F. et al. (2008). Bigtable: A Distributed Storage System for Structured Data. ACM TOCS, 26(2), pp 1–26.