PEOPLE COME FIRST
INFORMATIKAI SZAKÉRTŐK EGYESÜLETE

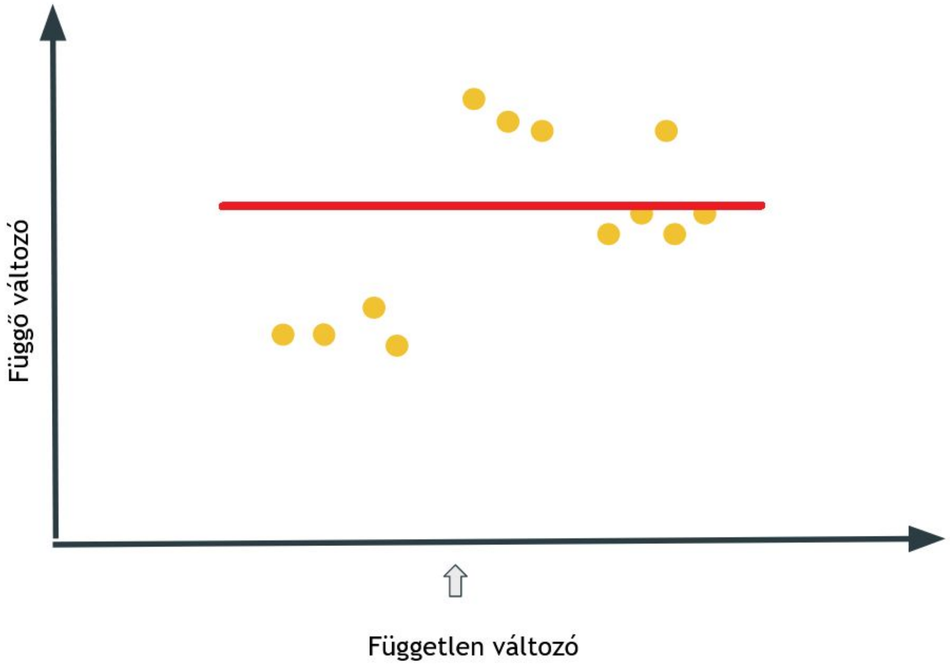
Big data/NoSQL

Oracle Junior Program - IT technológiák és architektúrák nagyvállalati környezetben

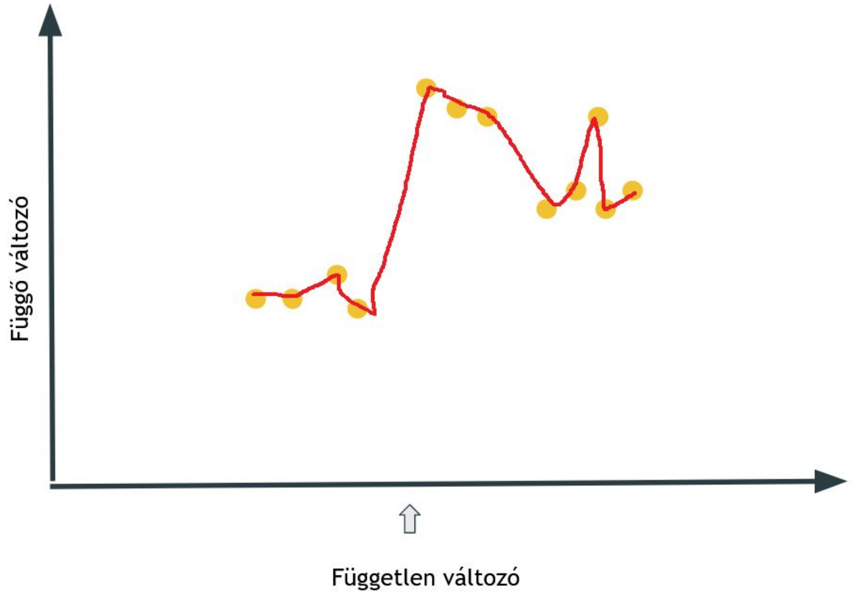
Szakal Ádám
(Webváltó)

Előző óráról

High bias

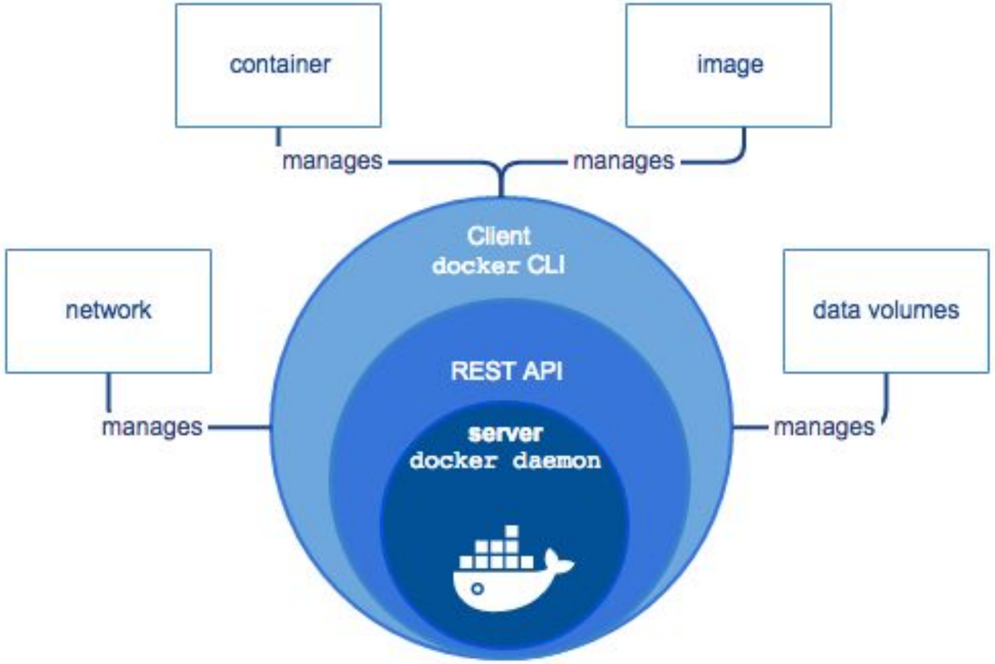


High Variance



Kitekintés Docker

- NoSQL-Big data technológiák kipróbálásához



Big data

Egyszerűen megfogalmazva

- „Olyan adatmennyiség, amitől az Excel már crash-el.”

Oxford dictionary

- „data sets that are too large and complex to manipulate or interrogate with standard methods or tools”

Dictionary.com

- „data sets, typically consisting of billions or trillions of records, that are so vast and complex that they require new and powerful computational resources to process

O’Reilly Media (M. Loukides)

- „As storage capacity continues to expand, today’s “big” is certainly tomorrow’s “medium” and next week’s “small.” The most meaningful definition I’ve heard: “big data” is when the size of the data itself becomes part of the problem.”

Big data

Gartner definíció (D. Laney):

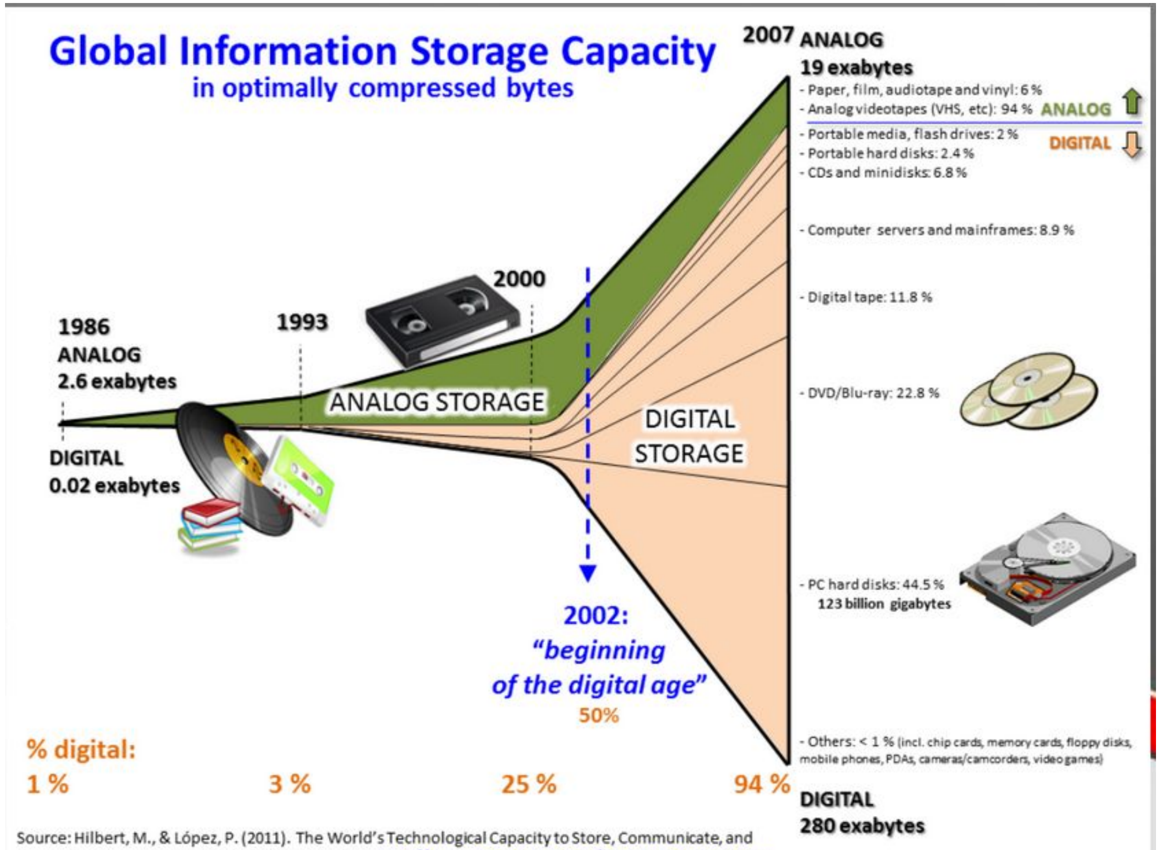
- „Big data are high **volume**, high **velocity**, and/or high **variety** information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.”

Fentit szokás 3-10V-nek is nevezni:

- Volume – méret
- Velocity – adatsebesség
- Variety – sokféle típusú adat
- …
  - Veracity – igazságtartalom, tisztaság; Validity – helyesség; Variability – egyre rugalmasabb struktúrák; Value – nagy értékű; Visualization – vizualizálhatóság; …

Big data - Volume

- The capacity to store information doubles roughly every 3.5 years
- Roughly 2,5 exabytes of data are generated every day



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

pcf

Big data Velocity

- So much so that the MetLife executive stressed that: “Velocity can be more important than volume because it can give us a bigger competitive advantage. Sometimes it’s better to have limited data in real time than lots of data at a low speed.”
- Standard relational databases are not capable of handling this amount of data
- Parallel computation

Big data - Variety

- Different data types are used: text, pictures,videos
- The capability to process important information is crucial
- Data come from different sources

Big data - Veracity

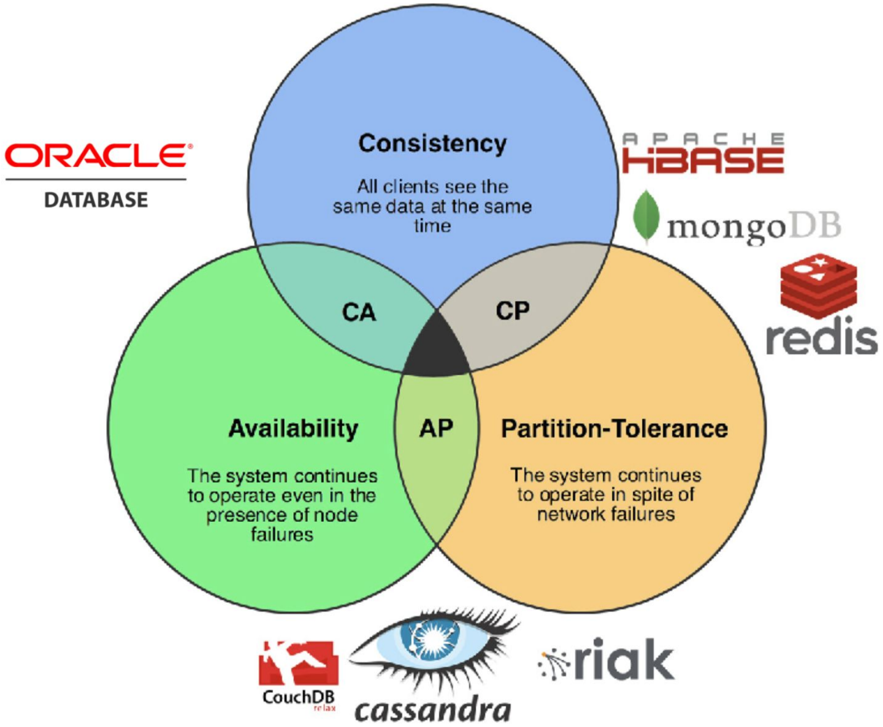
- Data source trustworthy
- The actual data makes sense based on business needs
- The only measure that decreases over time

Distributed Data Storing and Processing

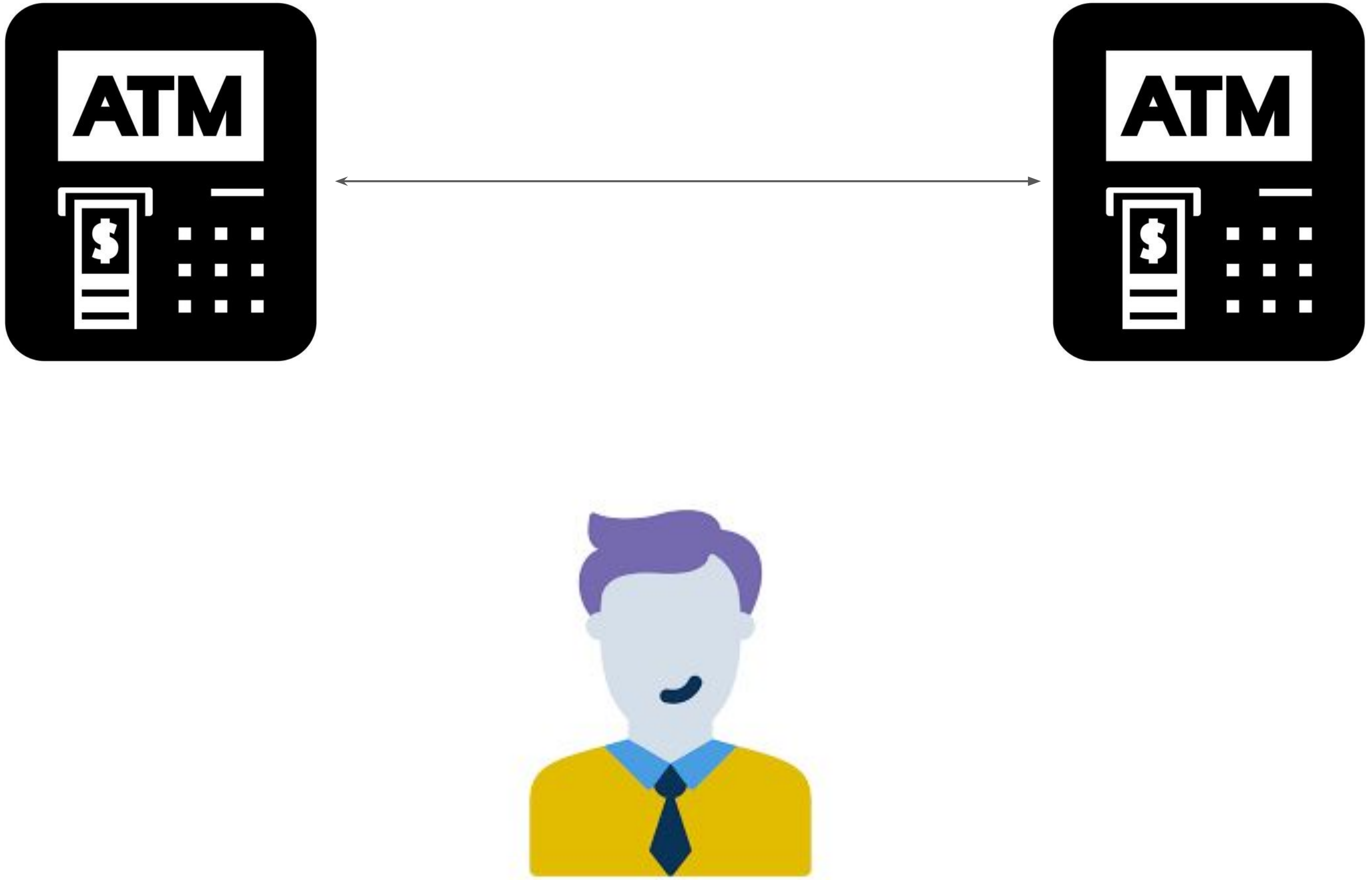
- CAP Theorem
- NoSQL database structures
- Distributed file systems
- Hadoop
  - HDFS
  - MapReduce

CAP Theorem

Impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees.


ORACLE DATABASE
Consistency
All clients see the same data at the same time
CA
CP
Availability
The system continues to operate even in the presence of node failures
AP
Partition-Tolerance
The system continues to operate in spite of network failures
APACHE HBASE
mongoDB
redis
CouchDB relax
cassandra
riak

CAP Theorem



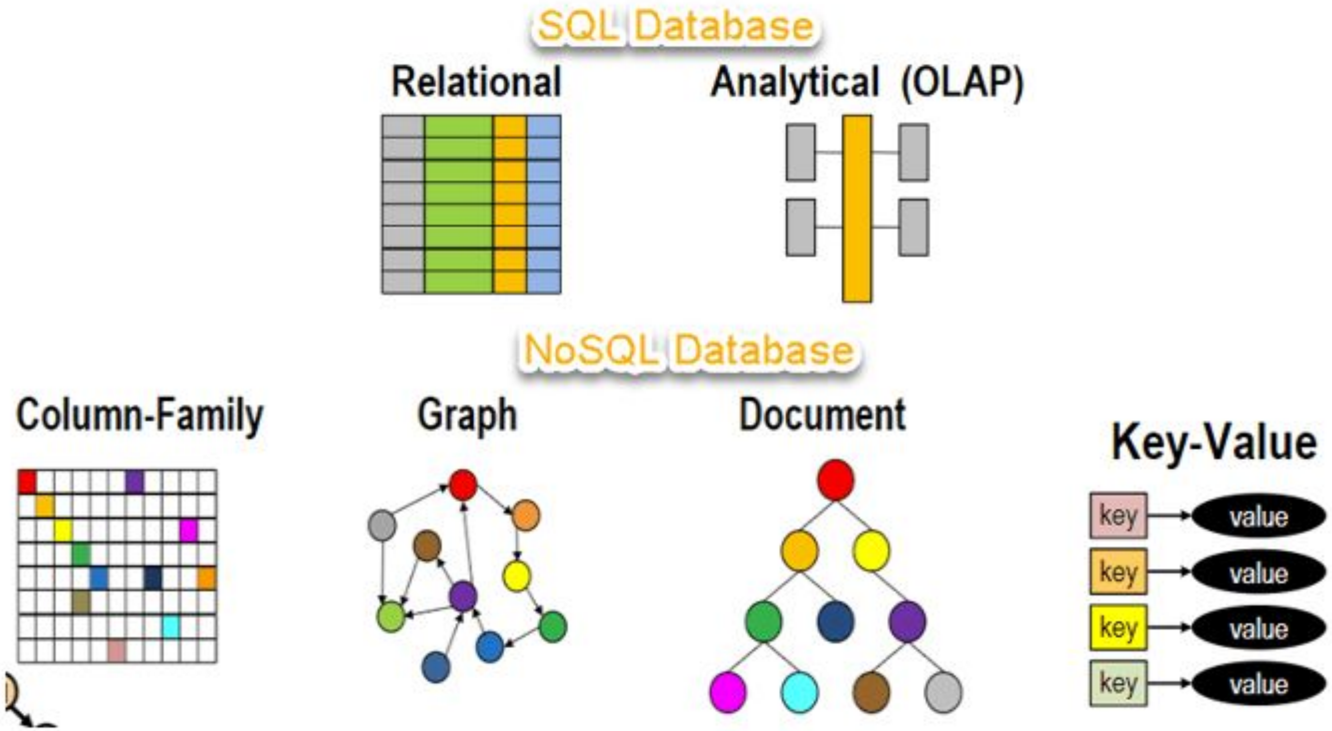
Distributed data storing(BASE)

Eventually-consistent services are often classified as providing BASE (Basically Available, Soft state, Eventual consistency) semantics, in contrast to traditional ACID (Atomicity, Consistency, Isolation, Durability) guarantees. In chemistry BASE is opposite to ACID, which helps remembering the acronym. According to the same resource, these are the rough definitions of each term in BASE.

Distributed data storing(BASE)

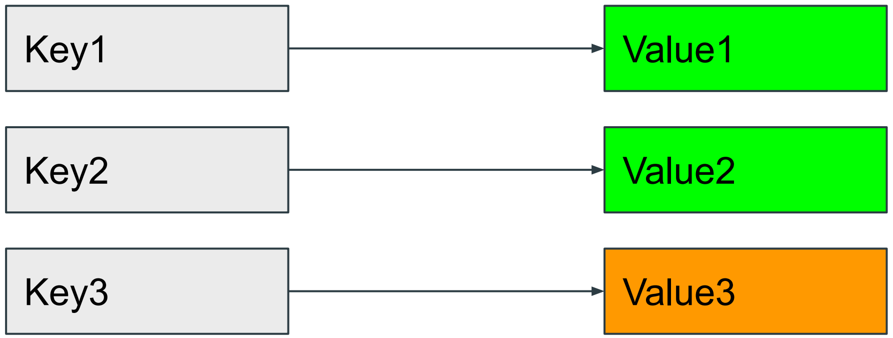
- (B)asically (A)vailable: basic reading and writing operations are available as much as possible (using all nodes of a database cluster), but without any kind of consistency guarantees (the write may not persist after conflicts are reconciled, the read may not get the latest write)
- (S)oft state: without consistency guarantees, after some amount of time, we only have some probability of knowing the state, since it may not yet have converged
- (E)ventually consistent: If the system is functioning and we wait long enough after any given set of inputs, we will eventually be able to know what the state of the database is, and so any further reads will be consistent with our expectations

NoSQL


SQL Database
Relational
Analytical (OLAP)
NoSQL Database
Column-Family
Graph
Document
Key-Value
key
value
key
value
key
value
key
value

Key-value database

- Key based search
- Hash function
- Value
  - Various type of objects
  - Don’t search in value
- Redis, Dynamo, Riak, Oracle NoSQL database



Key1

Value1

Key2

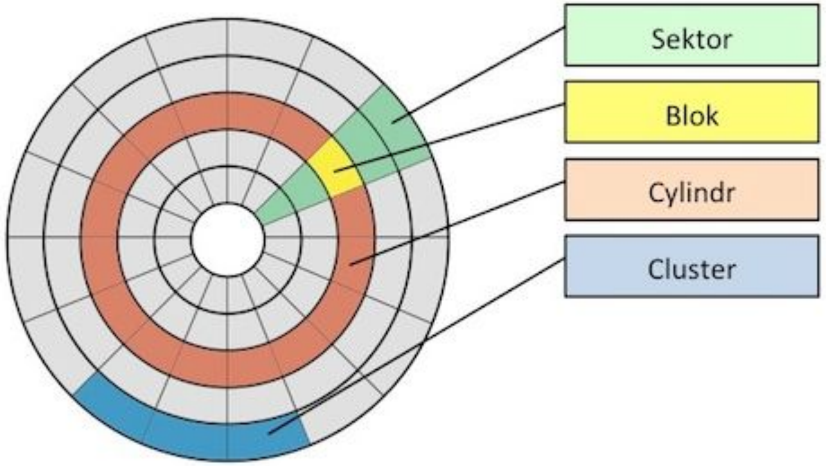
Value2

Key3

Value3

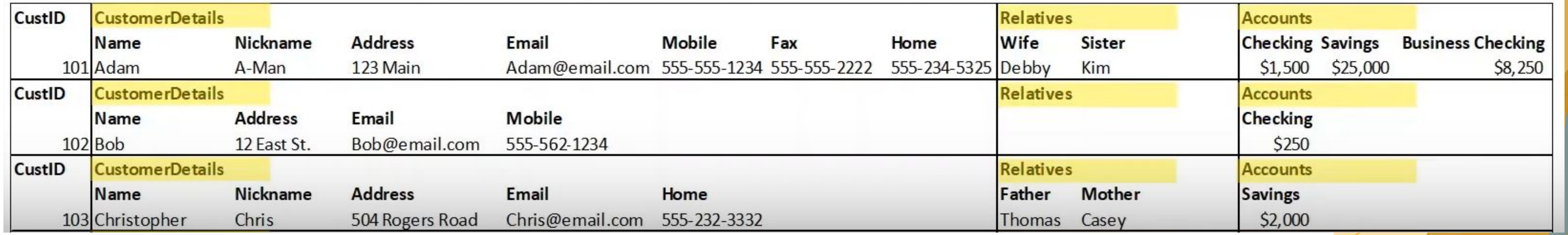
Column oriented database

- Stores data tables by column rather than by row
- SQL syntax
- Data compression
- Fast column functions: SUM, COUNT, AVG, MIN



Column family database

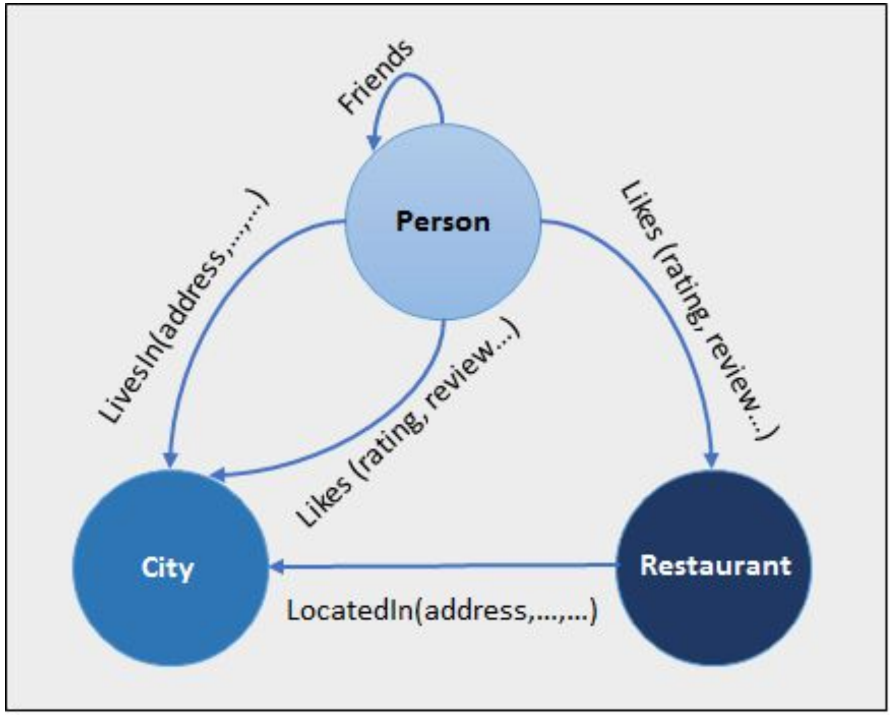
- Loose schema and various data types
- Google Bigtable, Apache Cassandra



CustID	CustomerDetails							Relatives		Accounts		
	Name	Nickname	Address	Email	Mobile	Fax	Home	Wife	Sister	Checking	Savings	Business Checking
101	Adam	A-Man	123 Main	Adam@email.com	555-555-1234	555-555-2222	555-234-5325	Debby	Kim	$1,500	$25,000	$8,250
CustID	CustomerDetails							Relatives		Accounts		
	Name	Address	Email	Mobile						Checking		
102	Bob	12 East St.	Bob@email.com	555-562-1234						$250		
CustID	CustomerDetails							Relatives		Accounts		
	Name	Nickname	Address	Email	Home			Father	Mother	Savings		
103	Christopher	Chris	504 Rogers Road	Chris@email.com	555-232-3332			Thomas	Casey	$2,000		

Graph database

- Node: Data
- Edge: stored relation
- Fast join!
- Neo4J, Infinite Graph, OrientDB, FlockDB


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

Documentum oriented database

- Like key-value database
- Document is organized structure (JSON)
- Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB,

NoSQL Advantages

- Can be used as Primary or Analytic Data Source
- **No Single Point of Failure**
- **Easy Replication**
- No Need for Separate Caching Layer
- It provides fast performance and horizontal scalability.
- Can handle structured, semi-structured, and unstructured data with equal effect
- Object-oriented programming which is easy to use and flexible
- **NoSQL databases don't need a dedicated high-performance server**
- Support Key Developer Languages and Platforms
- Simple to implement than using RDBMS
- It can serve as the primary data source for online applications.
- **Handles big data which manages data velocity, variety, volume, and complexity**
- Excels at distributed database and multi-data center operations
- Eliminates the need for a specific caching layer to store data
- Offers a flexible schema design which can easily be altered without downtime or service disruption

NoSQL Disadvantages

- **No standardization rules**
- **Limited query capabilities**
- RDBMS databases and tools are comparatively mature
- It does not offer any traditional database capabilities, like **consistency** when multiple transactions are performed simultaneously.
- When the volume of data increases it is difficult to maintain unique values as keys become difficult
- Doesn't work as well with relational data
- The learning curve is stiff for new developers
- **Open source options so not so popular for enterprises.**

Distributed file systems

- Uniform interface
  - Naming conventions
  - Mapping scheme
  - Can be reached like local files (mounting, unmounting)
  - Permission handling
- Clients should have a same view
- Authorization
- Goal maximization of local reads

Distributed file systems

- Access transparency: clients are unaware that files are distributed and can access them in the same way as local files are accessed.
- Location transparency: a consistent namespace exists encompassing local as well as remote files. The name of a file does not give its location.
- Concurrency transparency: all clients have the same view of the state of the file system. This means that if one process is modifying a file, any other processes on the same system or remote systems that are accessing the files will see the modifications in a coherent manner.
- Failure transparency: the client and client programs should operate correctly after a server failure.
- Heterogeneity: file service should be provided across different hardware and operating system platforms.

Distributed file systems

- Scalability: the file system should work well in small environments (1 machine, a dozen machines) and also scale gracefully to bigger ones (hundreds through tens of thousands of systems).
- Replication transparency: Clients should be unaware of the file replication performed across multiple servers to support scalability.
- Migration transparency: files should be able to move between different servers without the client's knowledge.

Hadoop Distributed file systems (HDFS)

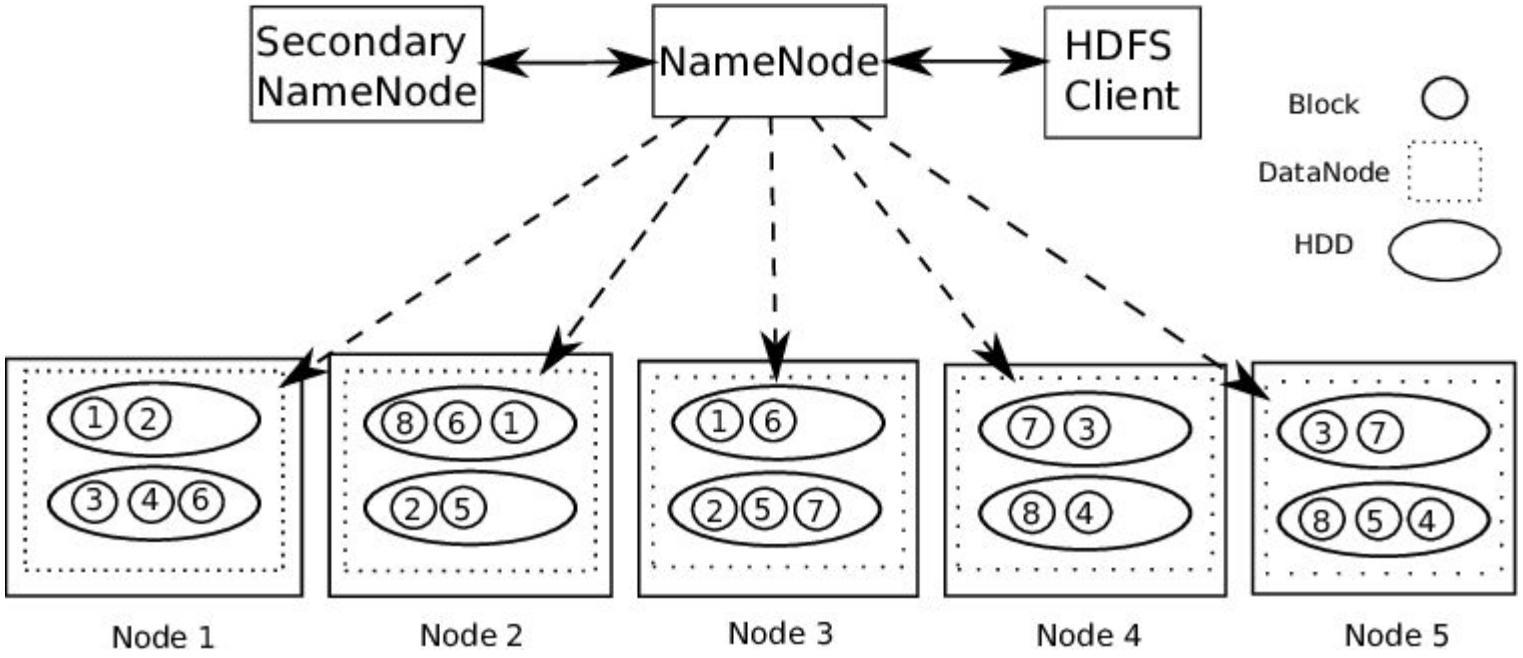
Written in JAVA

Lack of POXIS(Portable Operating System Interface) interface

Shell commands/JAVA API

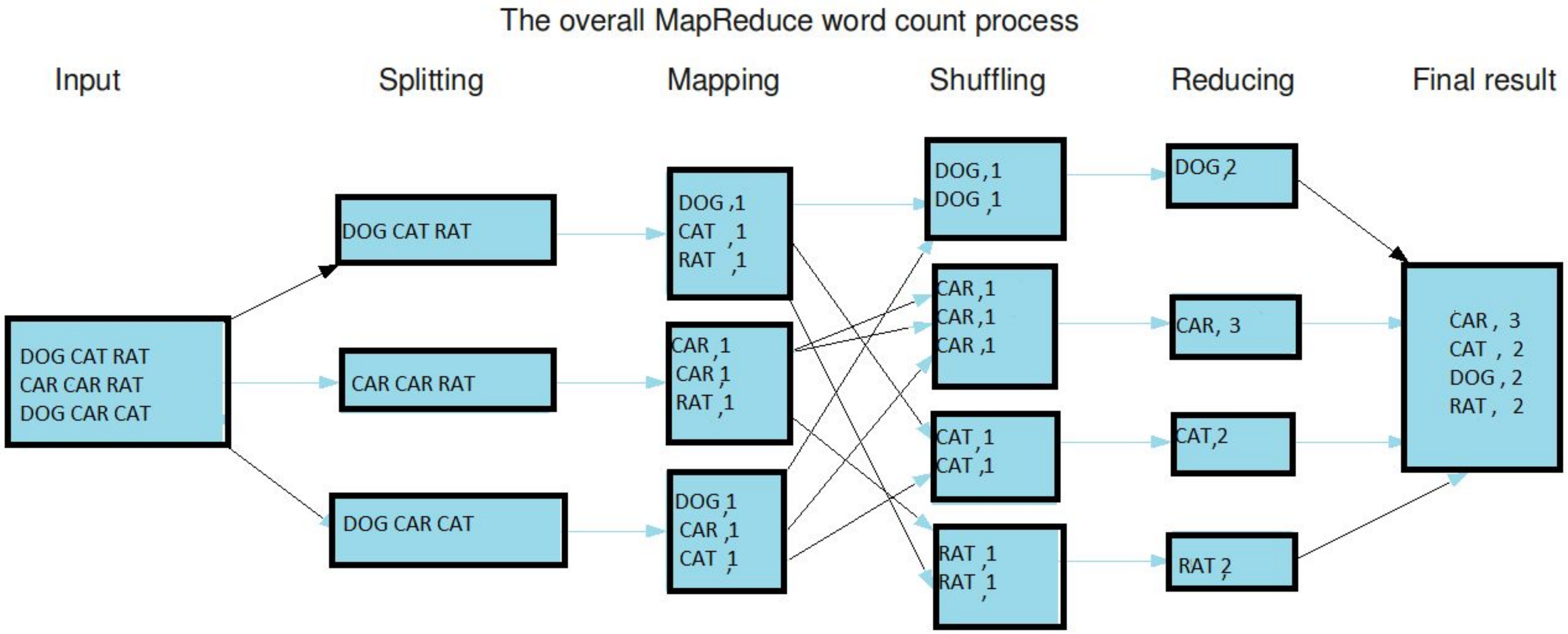
- Master elements
  - Name Node
  - Secondary Name Node
  - Job tracker
- Slave elements
  - Data Node
  - Task Tracker

Hadoop Distributed file systems (HDFS)



Hadoop mapreduce

Done by Job and Task trackers



The overall MapReduce word count process

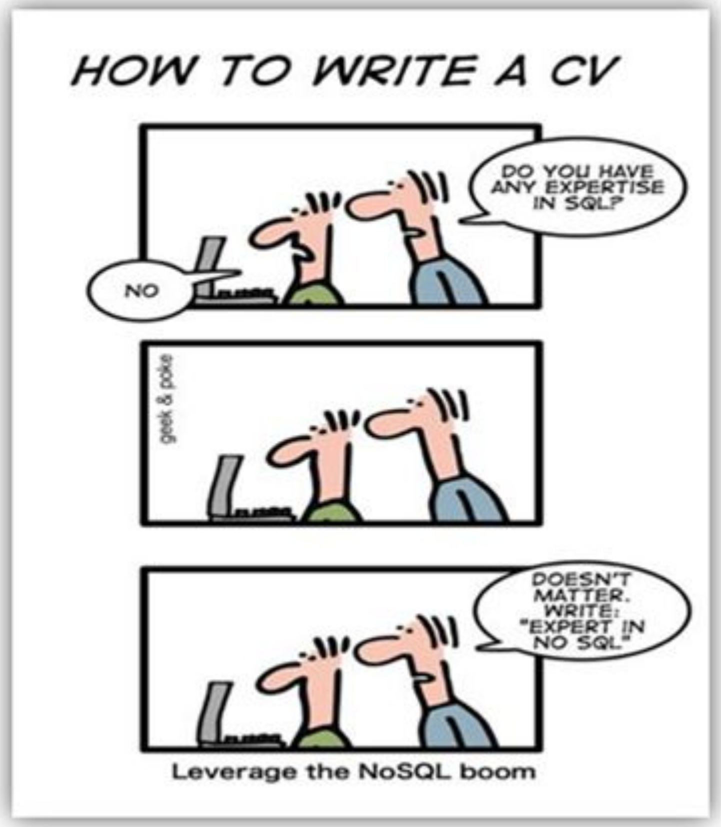
More to observe

Apache Spark

Apache Casandra

Apache Kafka

Köszönöm a figyelmet!


HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"
Leverage the NoSQL boom