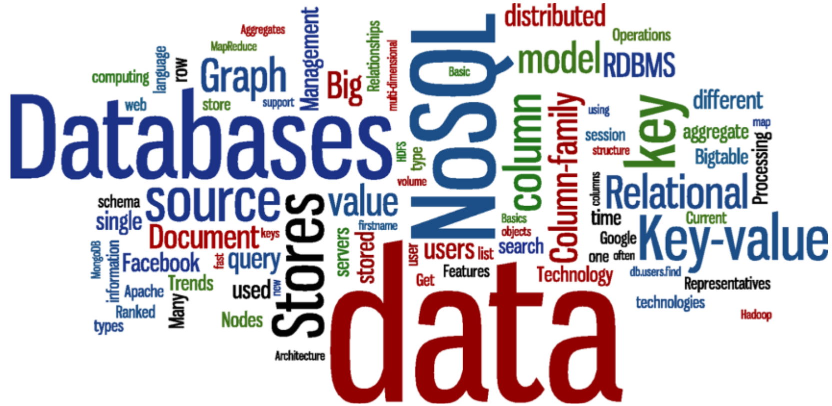


Why NoSQL, Principles, Overview

Lecture 1 of *NoSQL Databases* (PA195)

David Novak & Vlastislav Dohnal
Faculty of Informatics, Masaryk University, Brno

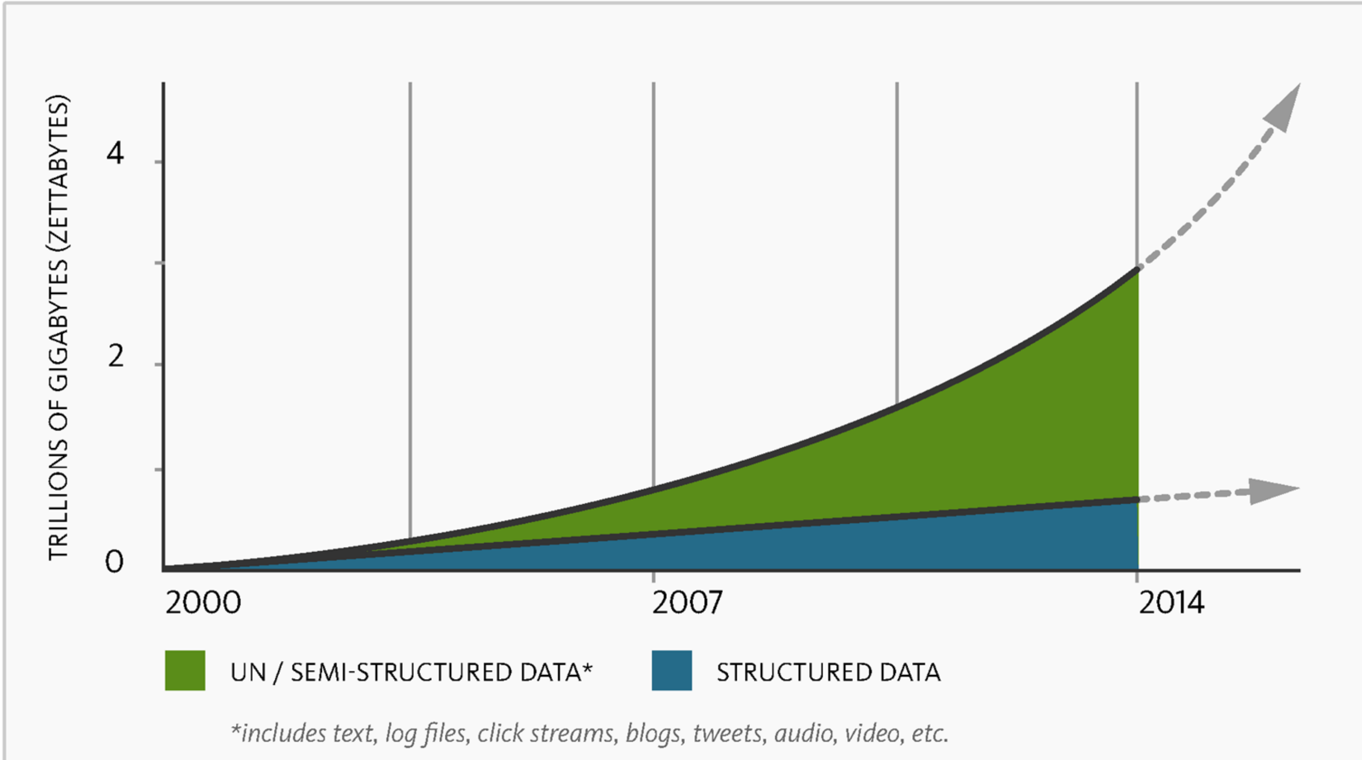
Agenda

- 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

Agenda

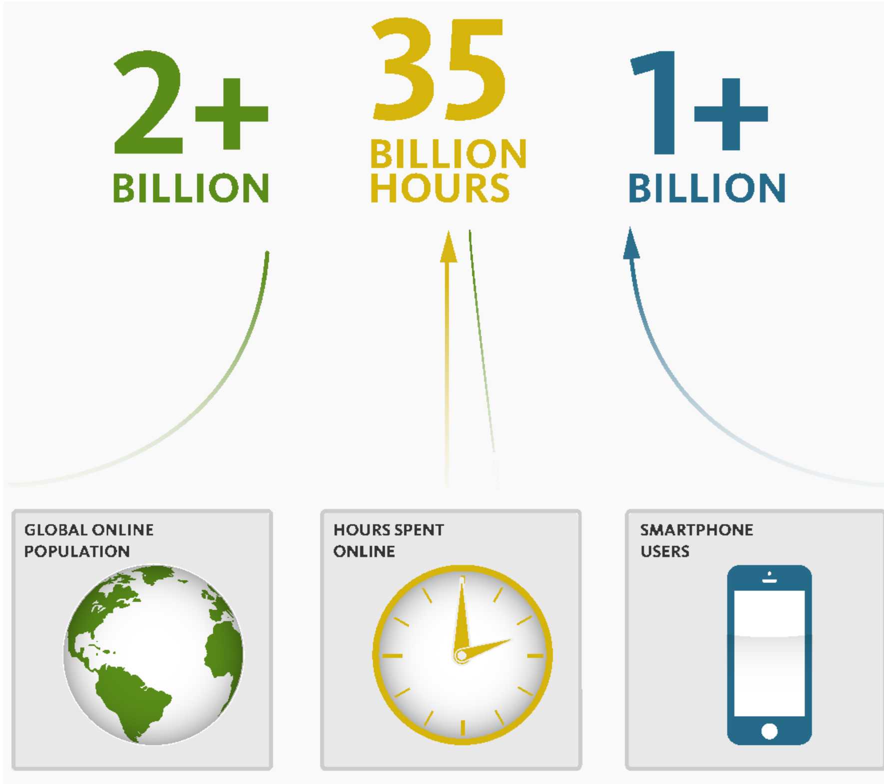
- Current trends in data management & computing
- Big Data
- Relational vs. NoSQL databases
    - the value of relational databases
    - new requirements
    - NoSQL features, strengths and challenges
- Types of NoSQL databases
    - key-value stores, document databases, column-family databases, graph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

Current Trends: Big Data



- Volume, Velocity and Variety of data

Current Trends: Big Users



- It is common to start a Web-based system and have millions of users within a few months

Current Trends: Cloud Computing

- Everything is in Cloud
  - flexibility and distributed nature of the systems

Agenda

- 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

Big Data

“Big data is 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)

veracity – precision vs. uncertainty of data
value – information extraction needed to get a value

Sources of Big Data

- Social networks
  - this data is huge, but volumes can be relatively limited
- Logs of various web/email servers or routers
  - growing beyond limits
- Sensor networks
  - this sector is expected to grow even faster
- Internet of things (IoT)
- Computer-driven machines, like airplanes:
  - one overseas flight of Boeing generates 640 TB of data
- etc.

Processing (Traditional) Data

- OLTP: Online Transaction Processing
    - Standard databases (DBMSs) and 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

Agenda

- 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

Relational Database Management Systems

- RDBMS are predominant database technologies
  - first defined in 1970 by Edgar Codd, IBM 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
  - each tuple identified by a key
    - field (or a set of fields) that uniquely identifies a row
  - tables and objects “interconnected” via foreign keys
- Relational algebra, 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, supported,...

Data Management: Trends & Requirements

Trends

- **Volume** of data
- **Cloud** comp. (IaaS)
- **Velocity** of data
- **Many** 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. Good 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 to store 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 wilder
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.

https://redis.io/docs/about/users/

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 under different circumstances [Sadalage & Fowler: NoSQL Distilled, 2012]

Two trends:

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

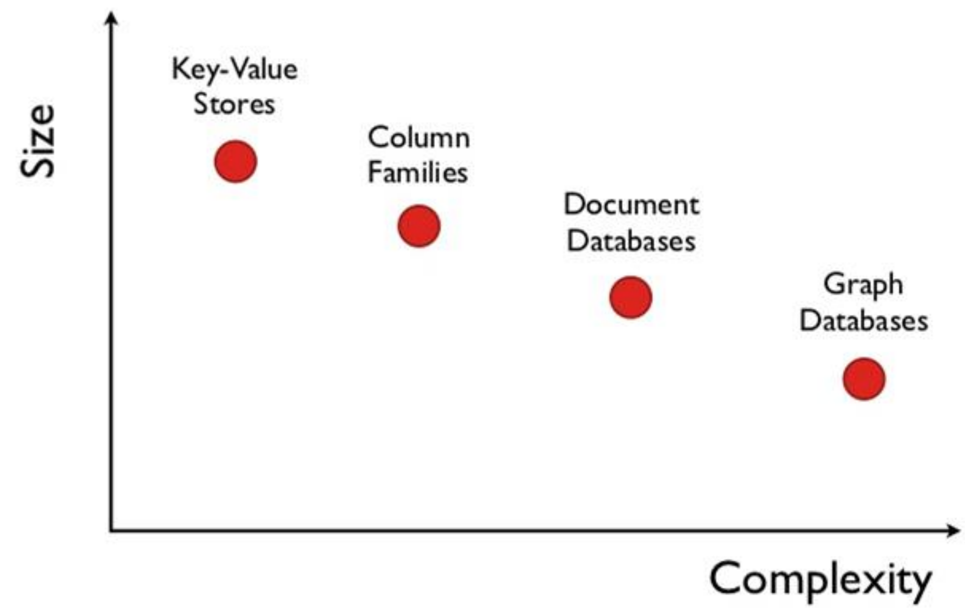
Agenda

- 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

NoSQL Technologies

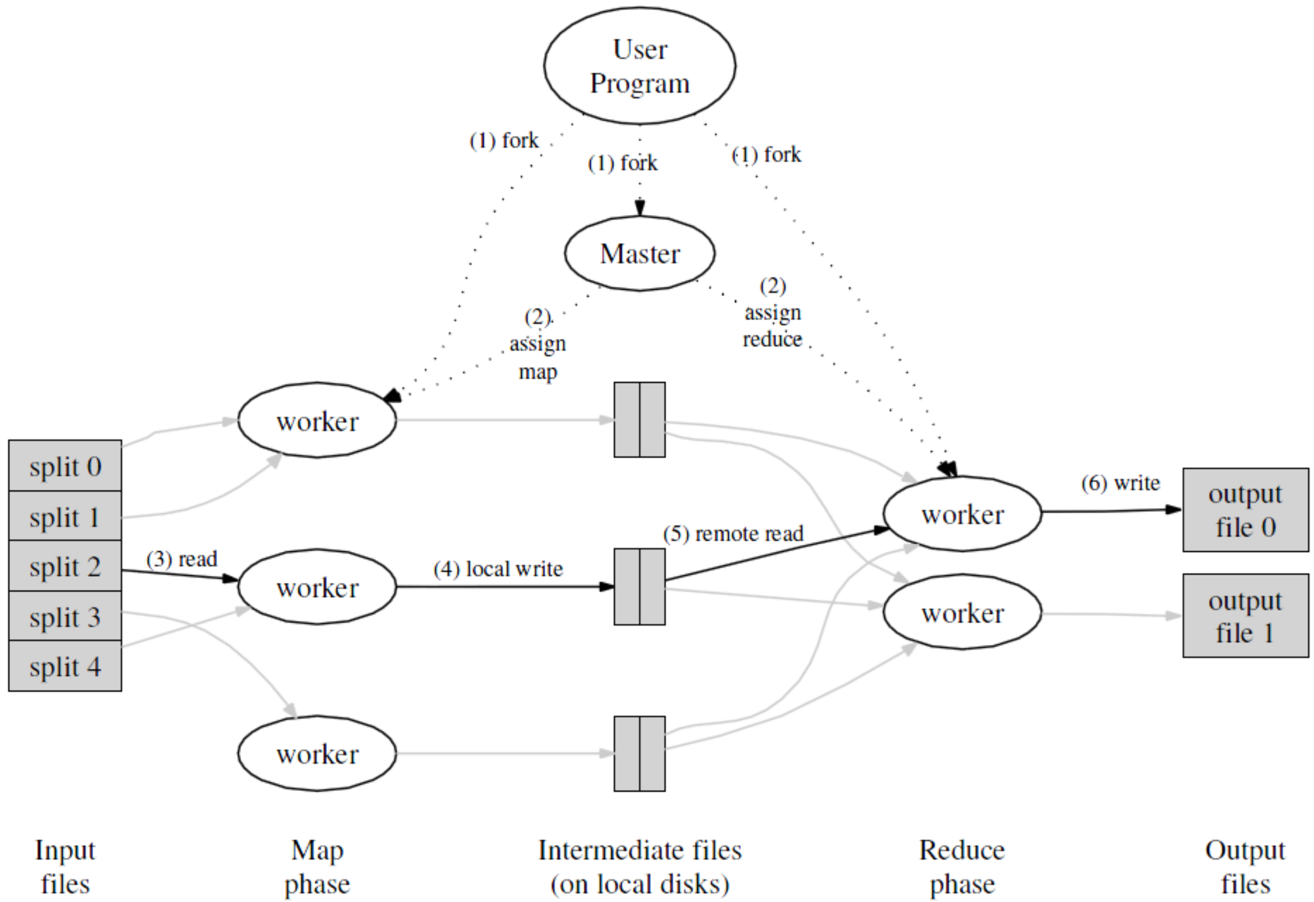
- MapReduce programming model
  - running over a distributed file system

- 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

redis

Cassandra

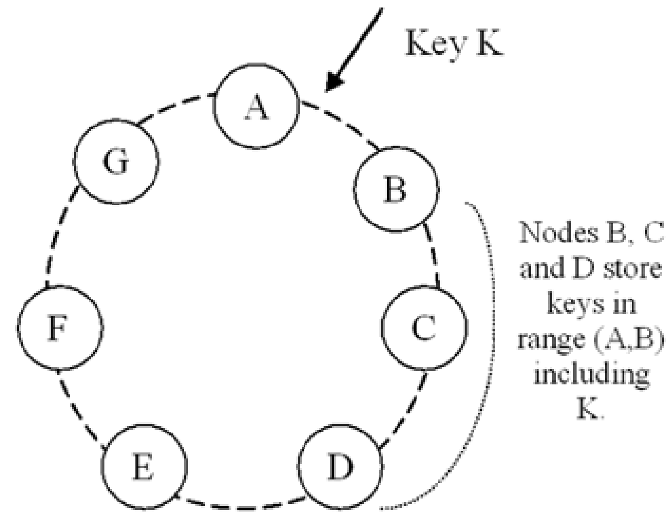
Key-value Stores: Basics

- A simple hash table (map), primarily used when all accesses to the database are via primary key
  - key-value mapping
- In RDBMS world: A table with two columns:
  - `ID` column (`primary key`)
  - `DATA` column storing the value (unstructured `BLOB`)
- Basic operations:
  - Put a value for a key `put(key, value)`
  - Get the value for the key `value:= get(key)`
  - Delete a key-value `delete(key)`

Key-value Stores: Architecture

1. Embedded systems
   - the system is a library and the DB runs within your system

2. Large-scale Distributed stores



Architecture often as a distributed hash table (DHT)

Features: it is simple

- great performance, easily scaled

source: http://www.allthingsdistributed.com/2007/10/amazons_dynamo.html

Key-value Stores: Representatives

amazon DynamoDB

redis

levelDB

ORACLE BERKELEY DB 12c

Infinispan

RocksDB

Project Voldemort

MEMCACHED

MapDB

hazelcast

EHCACHE

Ranked list: http://db-engines.com/en/ranking/key-value+store

Document Databases: Basics

- Basic concept of data: *Document*
- Documents are self-describing pieces of data
  - Hierarchical tree data structures
  - Nested associative arrays (maps), collections, scalars
  - XML, JSON (JavaScript Object Notation), BSON, …
- Documents in a collection should be "similar"
  - Their schema can differ
- Documents stored in the value part of key-value
  - Key-value stores where the values are examinable
  - Building search indexes on various keys/fields

Document Databases: Data Example

```
key=3 ->  { "personID": 3,
            "firstname": "Martin",
            "likes": [ "Biking","Photography" ],
            "lastcity": "Boston",
            "visited": [ "NYC", "Paris" ] }

key=5 ->  { "personID": 5,
            "firstname": "Pramod",
            "citiesvisited": [ "Chicago", "London","NYC" ],
            "addresses": [
               { "state": "AK",
                 "city": "DILLINGHAM" },
               { "state": "MH",
                 "city": "PUNE" }  ],
            "lastcity": "Chicago“ }
```

source: Sadalage & Fowler: NoSQL Distilled, 2012

Document Databases: Queries

Example in MongoDB syntax

- Query language expressed via JSON
- clauses: where, sort, count, sum, etc.

SQL: `SELECT * FROM users`

MongoDB: `db.users.find()`

```
SELECT * FROM users WHERE personID = 3
db.users.find( { "personID": 3 } )
```

```
SELECT firstname, lastcity FROM users WHERE personID = 5
db.users.find( { "personID": 5}, {firstname:1, lastcity:1} )
```

Document Databases: Representatives

mongoDB

Apache CouchDB relax

OrientDB

databricks

RAVENDB

Azure Cosmos DB

amazon DynamoDB

PostgreSQL

Ranked list: http://db-engines.com/en/ranking/document+store

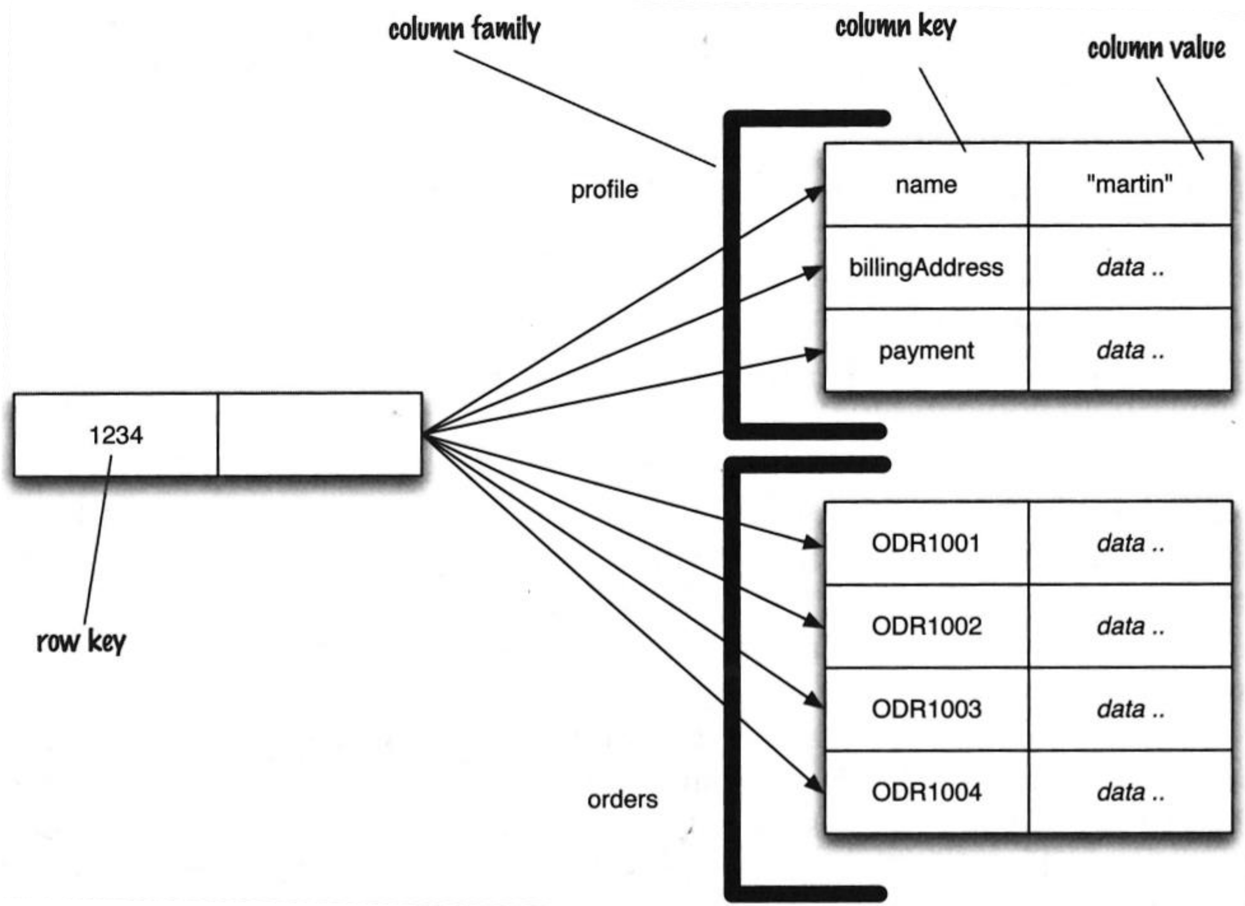
Column-family Stores: Basics

- AKA: wide-column, columnar

- Data model: rows (each identified with a row key) each row can have many columns

- Column families are groups of related data (columns) that are often accessed together
  - e.g., for a customer we typically access all profile information at the same time, but not customer's orders

Column-family Stores: Example



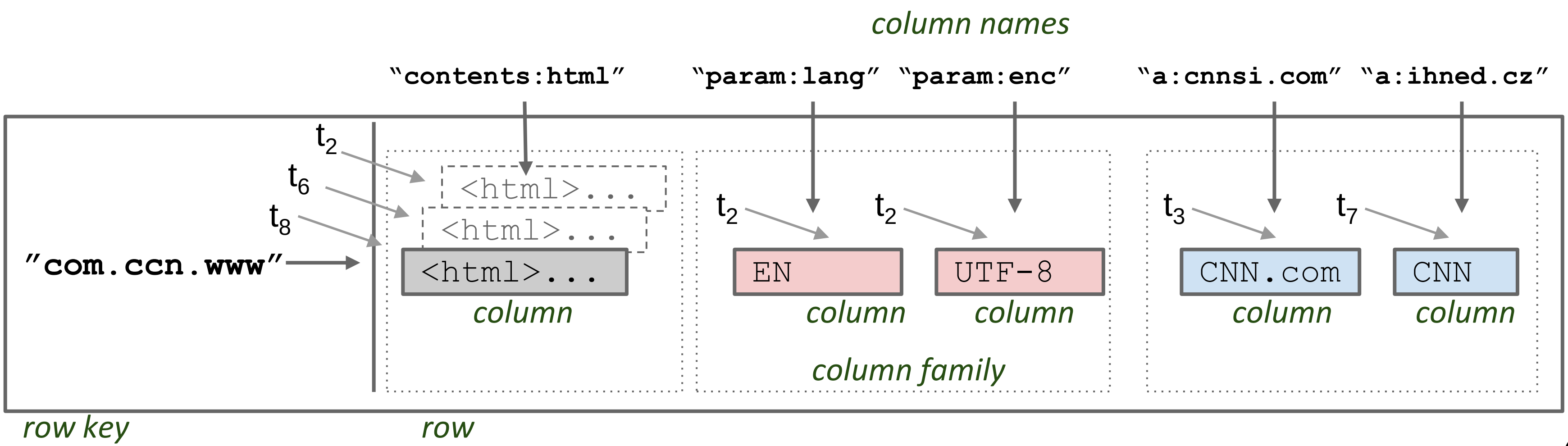
source: Sadalage & Fowler: NoSQL Distilled, 2012

Column-family Stores: BigTable

- 2008: Google publishes Bigtable Paper
- “BigTable = sparse, distributed, persistent, multi-dimensional sorted map indexed by *(row_key, column_key, timestamp)”*

Column-family Stores: BigTable

- 2008: Google publishes Bigtable Paper
- “BigTable = sparse, distributed, persistent, multi-dimensional sorted map indexed by *(row_key, column_key, timestamp)*”



Column-family Stores: Representatives

Cassandra

Google BigTable

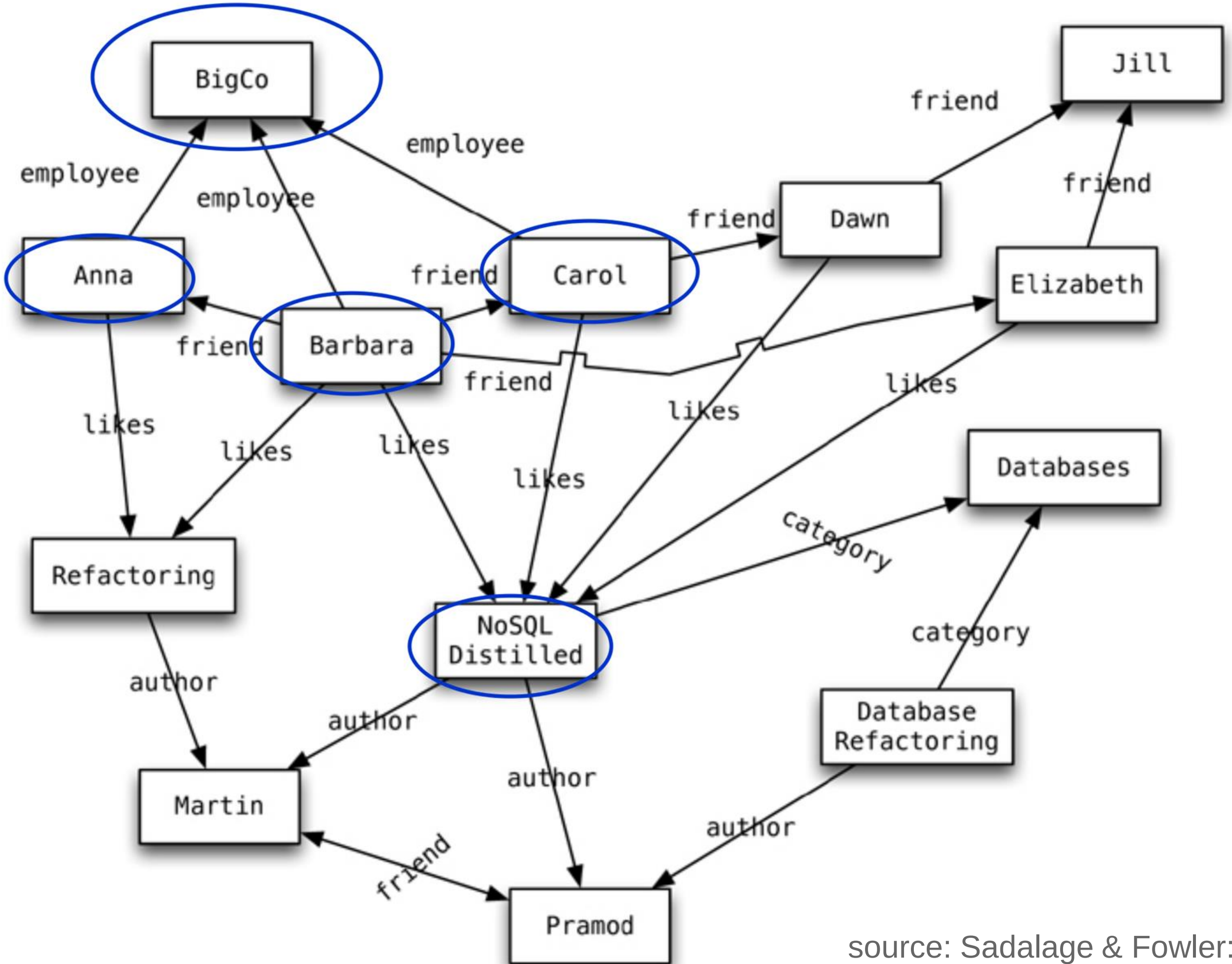
H·BASE

Azure Cosmos DB

SCYLLA

Ranked list: http://db-engines.com/en/ranking/wide+column+store

Graph Databases: Example



source: Sadalage & Fowler: NoSQL Distilled, 2012

Graph Databases: Mission

- To store entities and relationships between them
  - Nodes are instances of objects
  - Nodes have properties, e.g., name
  - Edges have directional significance
  - Edges have types e.g., likes, friend, …

- Nodes are organized by relationships
  - Allow to find interesting patterns
  - Example: Get all nodes that are “employee” of “Big Company” and that “likes” “NoSQL Distilled”

Graph Databases: Graphs in RDBMS

- When we store a graph-like structure in RDBMS, it is for a single type of relationship
  - “Who is my manager”
- Adding another relationship usually means a lot of schema changes
- In RDBMS, we model the graph beforehand based on the traversal we want
  - If the traversal changes, the data will have to change
  - Graph DBs: the relationship is not calculated but persisted

Graph Databases: Representatives

Neo4j

OrientDB®

APACHE GIRAPH

Azure Cosmos DB

MEMGRAPH

Ranked list: http://db-engines.com/en/ranking/graph+dbms

One Example: Facebook

Facebook statistics (2016)

- 1.86 billion monthly active users
- 4 million 'likes' per minute
- 250 billion stored photos (350 million uploaded daily)
- 300 PB of user data stored (2014)

2009: 10,000 servers

2010: 30,000 servers

2012: 180,000 servers (estimated)

source: http://expandedramblings.com/index.php/by-the-numbers-17-amazing-facebook-stats/
https://www.brandwatch.com/blog/47-facebook-statistics-2016/

Facebook: Database Tech. Behind

Apache Hadoop http://hadoop.apache.org/

- Hadoop File System (HDFS)
  - over 100 PB in a single HDFS cluster
- an open source implementation of MapReduce:
  - Enables efficient calculations on massive amounts of data

Apache Hive http://hive.apache.org/

- SQL-like access to Hadoop-stored data
- integration of MapReduce query evaluation

sources: http://goo.gl/SZ6jia http://royal.pingdom.com/2010/06/18/the-software-behind-facebook/

Facebook: Database Tech. Behind (2)

Apache HBase http://hbase.apache.org/

- a Hadoop column-family database
- used for e-mails, instant messaging and SMS
- replacement for MySQL and Cassandra
- but Instagram uses Cassandra

Memcached http://memcached.org/

- distributed key-value store
- used as a cache between web servers
  and MySQL servers in the beginning of FB

sources: http://goo.gl/SZ6jia http://royal.pingdom.com/2010/06/18/the-software-behind-facebook/

Facebook: Database Tech. Behind (3)

Graph database in MySQL

- 5+ billion entries
- in two tables (vertices and edges)
- Apache Giraph was previously used.

RocksDB http://rocksdb.org/

- high-performance key-value store
- developed internally in FB, now open-source
- now, LSM-Tree is used

sources: https://code.facebook.com/posts/509727595776839/scaling-apache-giraph-to-a-trillion-edges/ http://goo.gl/XNtG6p

Questions?

Please, any questions? Good question is a gift...

Found a bug/imprecision?

Please, report it by email.

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.