Bases de Dados

Universidade do Porto

Faculdade de Engenharia

FEUP

NoSQL

Observação: baseado em http://www.stanford.edu/class/cs145/ppt/cs145nosql.pptx

SQL Characteristics

- Data stored in columns and tables
- Relationships represented by data
- Data Manipulation Language
  - Select T1.Column1, T2.Column2 …
    From Table1, Table2 …
    Where T1.Column1 = T2.Column1 …
- Data Definition Language
  - Schema defined at the start
  - Create Table (Column1 Datatype1, Column2 Datatype 2, …)
  - Constraints to define and enforce relationships
    - Primary Key
    - Foreign Key
    - Etc.
  - Triggers to respond to Insert, Update , & Delete
  - Stored Modules
  - Alter …
  - Drop …
  - Security and Access Control
- Transactions: ACID properties
- Abstraction from physical layer

NoSQL Definition

From www.nosql-database.org:

> Next Generation Databases mostly addressing some of the points: being **non-relational, distributed, open-source** and **horizontal scalable**. The original intention has been **modern web-scale databases**. The movement began early 2009 and is growing rapidly. Often more characteristics apply as: **schema-free, easy replication support, simple API, eventually consistent / BASE** (not ACID), a **huge data amount**, and more.

Why NoSQL?

- Not every data management/analysis problem is best solved using a traditional relational DBMS
  - e.g., remember text search?
- "NoSQL" = "Not only SQL"
  - … = Not using traditional relational DBMS

NoSQL Systems

- Alternative to traditional relational DBMS
  - Flexible schema
    - including unstructured documents, images, videos
  - Quicker/cheaper to set up
  - Massive scalability
  - Relaxed consistency
    - higher performance & availability but fewer guarantees
  - No declarative query language
    - more programming

Example of Advanced Query (1/2)

- Social-network graph
  - Each record: UserID1, UserID2
  - Separate records: UserID, name, age, gender, …
- Task
  - Find all friends of friends of friends of … friends of given user

Example of Advanced Query (2/2)

- Wikipedia pages
  - Large collection of documents
  - Combination of structured and unstructured data

- Task
  - Retrieve introductory paragraph of all pages about U.S. presidents before 1900

Example in MongoDB

- find all orders shipped by CTT-Expresso

```
db.order.find( { shipping: {carrier: “ctt-expresso”} } )
```

- … and process results

```
var cursor = db.order.find( { shipping: {carrier: “ctt-expresso”} } );
cursor.hasNext();
cursor.forEach(
        function(item) {
                print(tojson(item))
                });
```

Types of NoSQL Systems

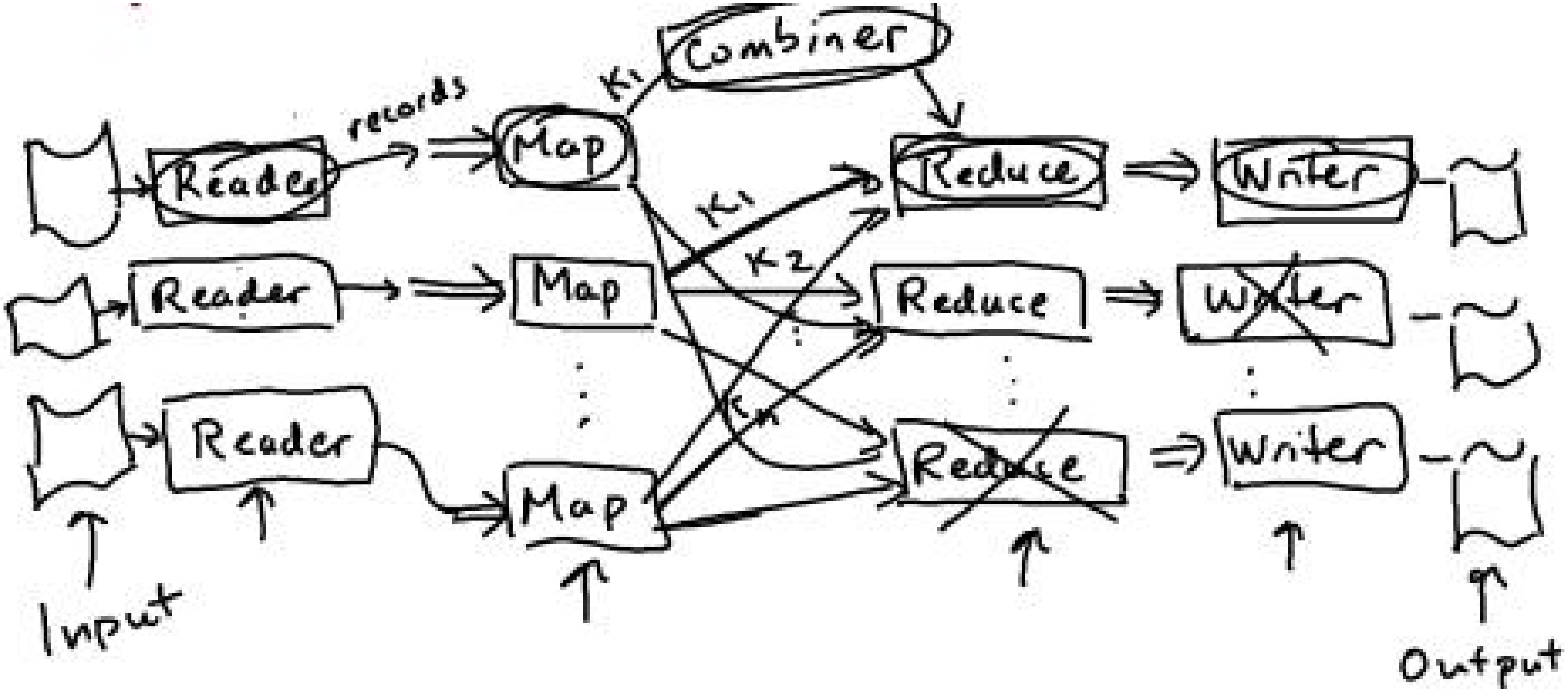
- MapReduce framework
- Key-value stores
- Document stores
- Graph database systems

MapReduce

- Originally from Google, open source Hadoop
  - No data model, data stored in files
  - User provides specific functions
    - map()
    - reduce()

- System provides
  - data processing “glue”
  - fault-tolerance
  - scalability

MapReduce Architecture



MapReduce Framework

- Schemas and declarative queries are missed
  - Hive
    - schemas
    - SQL-like query language
  - Pig
    - more imperative but with relational operators
- Both compile to “workflow” of Hadoop (MapReduce) jobs

Key-Value Stores

- Extremely simple interface
  - Data model: (key, value) pairs
  - Operations:
    - Insert(key,value)
    - Fetch(key)
- Implementation
  - efficiency
  - scalability
  - fault-tolerance

- Example systems
  - Google BigTable
  - Amazon Dynamo
  - Cassandra
  - Voldemort
  - HBase

Document Stores

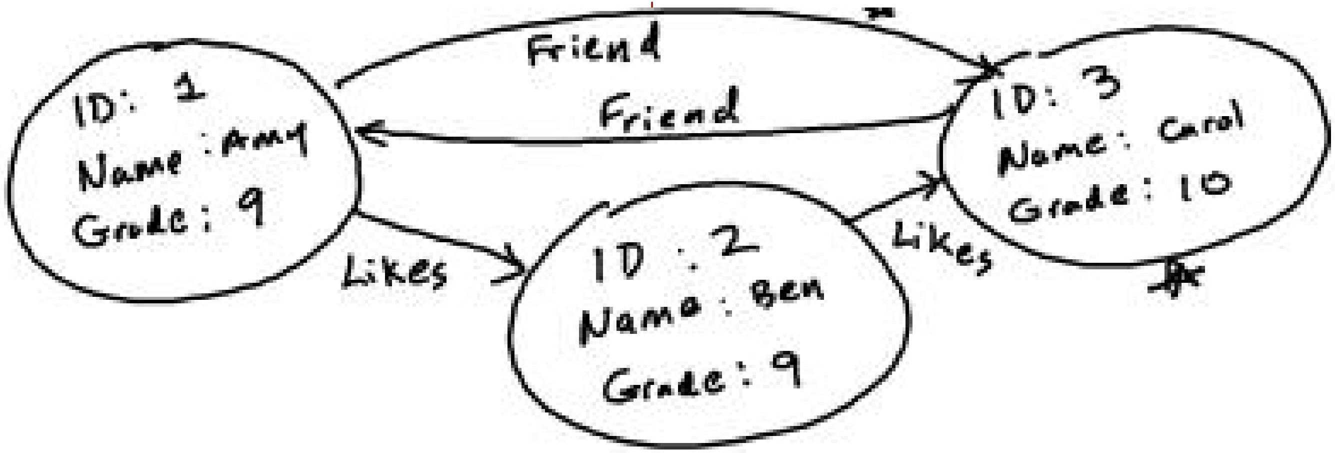
- Key-Value Stores where value is a document
  - Data model: (key, document) pairs
  - Document: JSON, XML, other semistructured formats
- Basic operations:
  - Insert(key,document)
  - Fetch(key)
- Also Fetch based on document contents

- Example systems
  - CouchDB
  - MongoDB
  - SimpleDB

Graph Database Systems

- Data model: nodes and edges
  - Nodes may have properties (including ID)
  - Edges may have labels or roles

- Example systems
  - Neo4j
  - FlockDB
  - Pregel



Distributed systems

A **distributed system** should have the following characteristics:

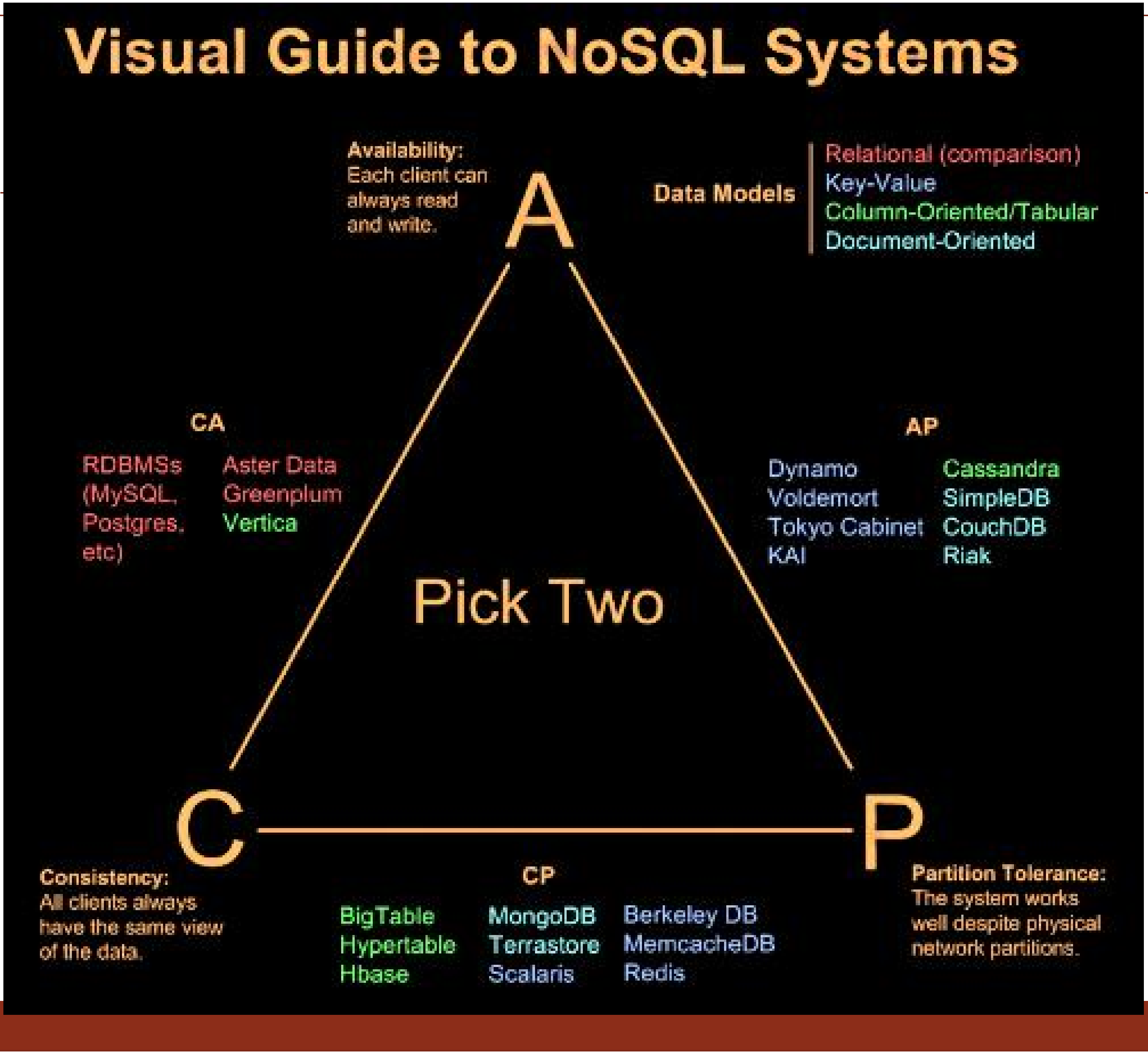
- Consistency
  - All nodes see the same data at the same time – Wikipedia
  - Client perceives that a set of operations has occurred all at once – Pritchett
  - More like Atomic in ACID transaction properties
- Availability
  - Node failures do not prevent survivors from continuing to operate – Wikipedia
  - Every operation must terminate in an intended response – Pritchett
- Partition tolerance
  - The system continues to operate despite arbitrary message loss – Wikipedia
  - Operations will complete, even if individual components are unavailable – Pritchett

http://queue.acm.org/detail.cfm?id=1394128

Brewer’ s CAP Theorem

A **distributed system** can support only two of the following characteristics:

- **Consistency**
- **Availability**
- **Partition tolerance**



Visual Guide to NoSQL Systems

Availability:
Each client can always read and write.

Data Models
- Relational (comparison)
- Key-Value
- Column-Oriented/Tabular
- Document-Oriented

CA

RDBMSs (MySQL, Postgres, etc)

Aster Data
Greenplum
Vertica

AP

Dynamo
Voldemort
Tokyo Cabinet
KAI

Cassandra
SimpleDB
CouchDB
Riak

Consistency:
All clients always have the same view of the data.

CP

BigTable
Hypertable
Hbase

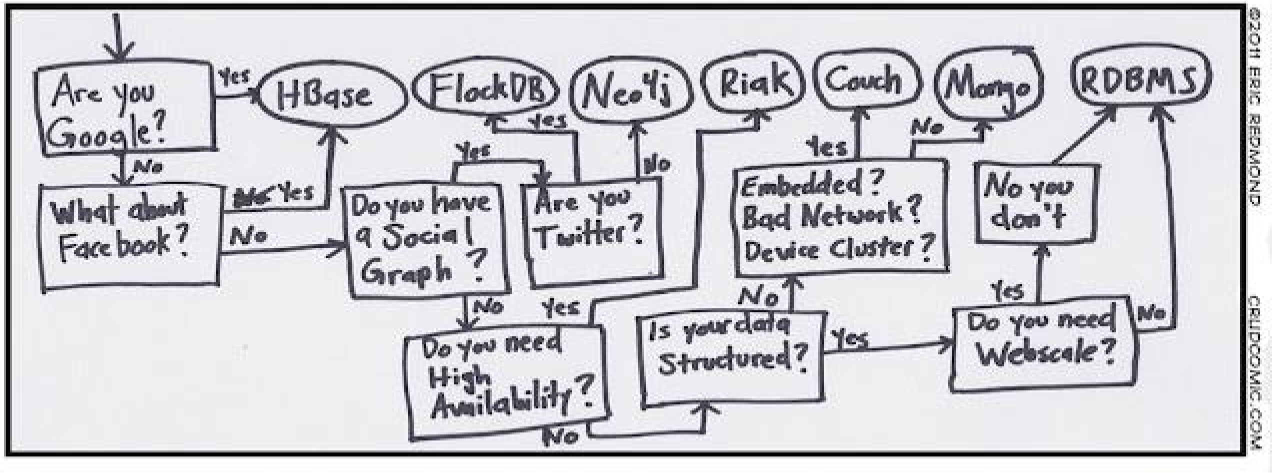
MongoDB
Terrastore
Scalaris

Berkeley DB
MemcacheDB
Redis

Partition Tolerance:
The system works well despite physical network partitions.

Summary

- "NoSQL" = "Not Only SQL"
  - Not every data management/analysis problem is best solved exclusively using a traditional DBMS
- Current incarnations
  - MapReduce framework
  - Key-value stores
  - Document stores
  - Graph database systems



©2011 ERIC REDMOND

CRUDCOMIC.COM