Big Data Analytics

6. NoSQL Databases

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Computer Science
University of Hildesheim, Germany

original slides by Lucas Rego Drumond, ISMLL

Outline

Outline

Relational vs Big-Data Technologies

- **Structure:** In relational data bases, the data is structured but in big data technologies the data is raw.
- **Process:** Relational data bases were initially created for transactional processing, but big data technologies are for data analysis.
- **Entities:** A relational data base is big if it has many entities and many rows per each entity. But in big data technologies, the number of entities is not necessary high. The amount of information that exist for entities is big.
- **horizontal scaling:** Adding new nodes in big data technologies can be done with low cost while in relational data base, either it fixed or it is expensive.

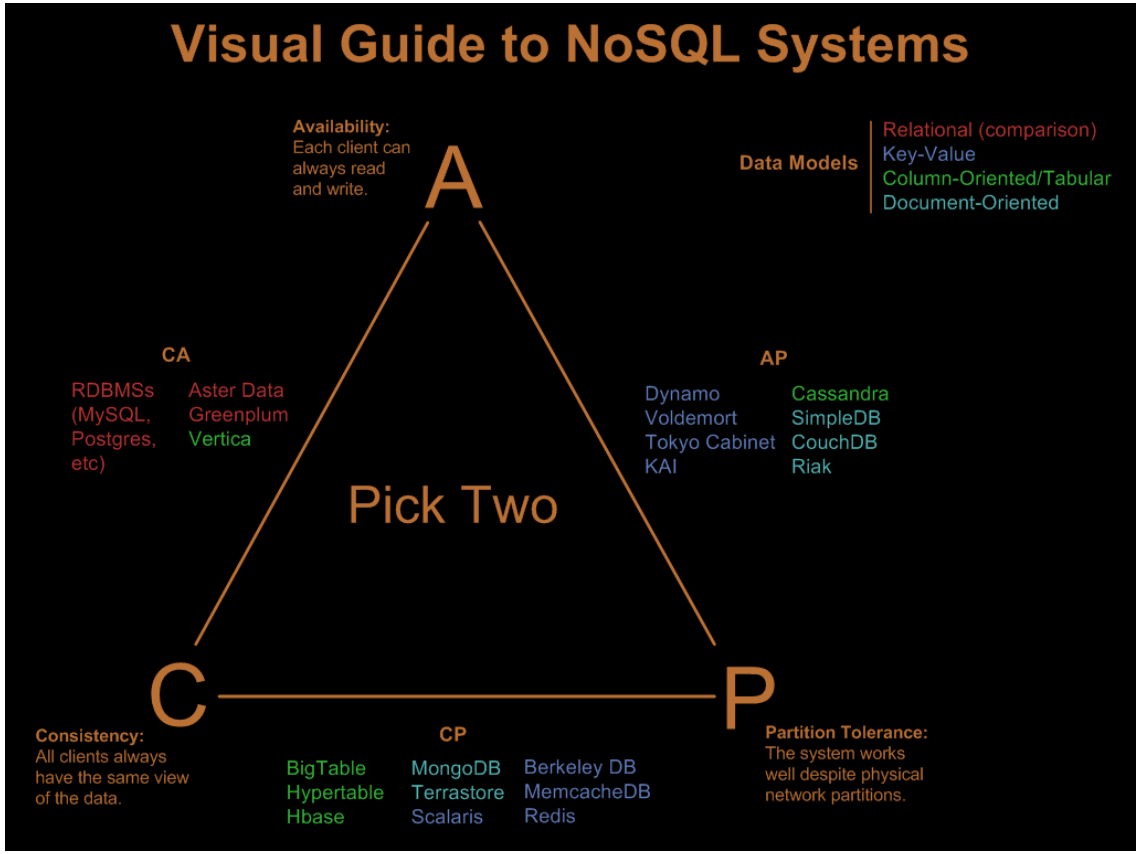
NoSQL Databases

- A NoSQL or Not Only SQL database provides a mechanism for storage and retrieval of data that is modeled in means **other than** the tabular relations used in relational databases.
- Motivations for this approach include simplicity of design and horizontal scaling.
- The data structure differs from the RDBMS, and therefore some operations are faster in NoSQL and some in RDBMS.
- Most NoSQL stores lack true ACID transactions.

NoSQL Databases / Types and Implementations

- **Key-value:** Dynamo, FoundationDB, MemcacheDB, Redis, Riak
- **Document:** Clusterpoint, Couchbase, MarkLogic, MongoDB
- **Graph:** Allegro, Neo4J, OrientDB, Virtuoso
- **Column:** Accumulo, Cassandra, HBase
- **Object:** Db4o, ZODB

...



http://blog.scottlogic.com/2014/08/04/mongodb-vs-couchdb.html

Outline

Key–Value stores

- Key–Value stores use the associative array as their fundamental data model.
- In this model, data is represented as a collection of key–value pairs.
- The key–value model is one of the simplest non-trivial data models.

Example:

```
{
"Great Expectations": "John",
"Pride and Prejudice": "Alice",
"Wuthering Heights": "Alice"
}
```

Outline

Document Databases

- A document-oriented database is a computer program designed for storing, retrieving, and managing document-oriented information.
- In contrast to relational databases and their notions of "Relations" (or "Tables"), these systems are designed around an abstract notion of a "Document".
- Documents inside a document-oriented database are not required to have all the same sections, slots, parts, or keys.

Example 1 / Schema-less

```
{
  _id: ObjectId(7df78ad8902c),
  FirstName: "Bob",
  Age: 35,
  Address: "5 Oak St.",
  Hobby: "sailing"
}
```

```
{
  _id: ObjectId(5df78ad8902c),
  FirstName: "Jonathan",
  Age: 37,
  Address: "15 Wanamassa Point Road",
  Languages: [ 'English', 'German' ]
}
```

Example 2 / Foreign Keys

```
{
  _id: ObjectId(5df78ad8902c),
  FirstName: "Jonathan",
  Age: 37,
  Address: "15␣Wanamassa␣Point␣Road",
  Children: [ ObjectId(5df78ad89020), ObjectId(5df78ad89021),
              ObjectId(5df78ad89022), ObjectId(5df78ad89023) ]
},{
  _id: ObjectId(5df78ad89020),
  FirstName: "Michael",
  Age: 10,
},{
  _id: ObjectId(5df78ad89021),
  FirstName: "Jennifer",
  Age: 8,
},{
  _id: ObjectId(5df78ad89022),
  FirstName: "Samantha",
  Age: 5,
},{
  _id: ObjectId(5df78ad89023),
  FirstName: "Elena",
  Age: 2,
}
```

Example 3 / Embedded Documents

```
{
  _id: ObjectId(5df78ad8902c),
  FirstName: "Jonathan",
  Age: 37,
  Address: "15 Wanamassa Point Road",
  Children: {
     { FirstName: "Michael", Age: 10 },
     { FirstName: "Jennifer", Age: 8 },
     { FirstName: "Samantha", Age: 5 },
     { FirstName: "Elena", Age: 2}
  }
}
```

Document Databases

- Documents are addressed in the database via a unique key that represents that document.
- Documents can be retrieved by their
  - key
  - content
- Documents are organized through
  - Collections
  - Tags
  - Non-visible Metadata
  - Directory hierarchies
  - Buckets

RDBMS vs. Document Database Terminology

relational database	document database
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Primary key	Primary key (e.g., default key _id in mongodb)
Foreign key	Foreign key

- In document databases, join operations often are replaced by **embedding documents**.
- Document databases are **schema-less**
  - one can add a new field at any time
  - a document basically is a dictionary (like Python objects)

Big Data Document Databases

Document databases are useful for big data for two main reasons:

1. Document databases can be **horizontally partitioned** / **sharded**.
   - Documents are distributed over different nodes.
   - Using a **partition/sharding function**
     - e.g., as hash function
   - works exactly the same ways as for RDBMS.

Big Data Document Databases

Document databases are useful for big data for two main reasons:

1. Document databases can be **horizontally partitioned** / **sharded**.
   - Documents are distributed over different nodes.
   - Using a **partition/sharding function**
     - e.g., as hash function
   - works exactly the same ways as for RDBMS.

2. Documents are **sparse representations**,
   - only some fields/keys have values,

   while tuples/rows are **dense**.
   - all columns have values stored explicitely
     - also NULL is explicitely stored.

Outline

Graph databases

A graph database is a database that uses graph structures with nodes, edges, and properties to represent and store data.

Graph databases

- Nodes represent entities such as people, businesses, accounts, or any other item you might want to keep track of.
- Properties are relevant information that relate to nodes.
- Edges are the lines that connect nodes to nodes
- Most of the important information is really stored in the edges.
- Meaningful patterns emerge when one examines the connections and interconnections of nodes, properties, and edges.

Graph databases

- Compared with relational databases, graph databases are often faster for associative data sets
- They map more directly to the structure of object-oriented applications.
- As they depend less on a rigid schema, they are more suitable to manage ad hoc and changing data with evolving schemas.
- Graph databases are a powerful tool for graph-like queries.

Graph queries

- Reachability queries
- shortest path queries
- Pattern queries

Outline

Column Databases

- Column databases are a tuple (a key-value pair) consisting of three elements:
  - Unique name: Used to reference the column
  - Value: The content of the column.
  - Timestamp: The system timestamp used to determine the valid content.
- Differences to a relational database

Example

```
{
    street: name: "street", value: "1234 x street", timestamp: 123456789,
    city: name: "city", value: "san francisco", timestamp: 123456789,
    zip: name: "zip", value: "94107", timestamp: 123456789,
}
```

Object Database

- An object database is a database management system in which information is represented in the form of objects as used in object-oriented programming.
- Most object databases also offer some kind of query language, allowing objects to be found using a declarative programming approach (OQL)
- Access to data can be faster because joins are often not needed.
- Many object databases offer support for versioning.
- They are specially suitable in applications with complex data.