Graph & Modern Databases

COMP1835

1

1

NoSQL Data Stores

2

2

Previously…

- ✓ Discussed challenges of RDMS
- ✓ Introduced NoSQL
- ✓ Compared RDMS and NoSQL
- ✓ Discussed BASE vs ACID
- ✓ Introduced CAP theorem

3

3

Objectives

- To discuss different NoSQL storage types
- To introduce key-value stores
- To discuss column-oriented stores
- To describe document-oriented data stores
- To introduce graph databases
- To discuss Polyglot Persistence

4

4

Part 1

5

5

Overview

- Data is like oil: we sit upon a vast ocean of data
- But until it's refined into **information,** it is unusable.
- Which database to choose for collecting, storing, mining and refining the data?
  - The decision is more complex than just considering which genre maps best to a given domain data
- For example, Facebook has far too much data and very different data to choose just one database.
  - The best would be to choose a 'Big Data' implementation, such as HBase or DynamoDB
- For bank transactions probably a relational database is clearly the best option,
  - however, some NoSQL data stores CouchDB, Neo4J and AllegroGraph databases also support ACID transactions
- Some of the functionalities of SQL databases like functions, stored procedures, and procedural language may not be present in most of the databases.

6

6

Choices, choices …

- As the size of data increases, the capacity of certain database styles wane.
- Every database server ever designed was built to meet specific design criteria.
  - Those design criteria define the use cases where the database will fit well and the use cases where it will not.
  - For example, column-oriented database implementations are often built to scale across datacentres and support the largest "Big Data" sets, when Graph databases usually support the smaller data sets
- Other criteria to consider:
  - Durability, Availability, Consistency, Scalability and Security
  - Would MapReduce be enough, or ad hoc queries will be required?
  - Would an HHTP/REST interface be needed?
  - Would bulk data loaders be needed?

http://clipart-library.com/clipart/1788539.htm#

7

7

Storage Types

- There are various ways to classify NoSQL databases, but the most common classification is by data model (how the data is stored):

- Key-Value
- Column-oriented (column family)
- Document
- Graph

8

8

Key-value data model

- In the Key-value (KV) model data is represented as a collection of key-value pairs,
- A key-value store is like a dictionary.
  - A dictionary has a list of words and each word has one or more definitions
- The key in a key-value store is flexible and can be represented by many formats:
  - Logical path names to images or files
  - Artificially generated strings created from a hash of the value
  - REST web service calls
  - SQL queries

Key	Value
Name	Joe
Age	42
Occupation	Developer
WebPage	www.joe.co.uk

- Values, like keys, are also flexible and can be any BLOB of data, such as images, web pages, documents, or videos.
- Typical uses:
  - Dictionary, image store, document/file store, query cache, lookup tables

9

9

Advantages

- Key-value stores are optimized for querying against keys.
- No need for schema
- No need to specify a data type for the value, which means that you can store any data type that you want in the value – similar to document-based data store
- But:
  - Unlike a document store that can create a key when a new document is inserted, a key-value store requires the key to be specified
  - Unlike a document store where the value can be indexed and queried, for a key-value store, the value is opaque and as such, the key must be known to retrieve the value
- The most prominent use of working with a key-value store is for in-memory distributed data store or otherwise cache

10

10

Key-Value Use Cases

- Storing Web application session information
  - Web sessions have a unique id
  - Provides faster performance than a relational database
  - For example, a user's browsing history
- User profiles and preferences
  - User have unique ids
  - Data put into an object and stored with id as the key
- Shopping cart data
  - To be available across sessions, machines, browsers
  - Key – user id, value – content of the cart

11

11

Key-value databases

- Aerospike
- Berkeley DB
- Couchbase Server
- DynamoDB
- Kyoto Cabinet
- Oracle NoSQL Database
- Project Voldemort
- Redis
- Riak
- Others

12

12

Discussion Question

- What would be the disadvantages of the key-value data stores?

13

13

Part 2

14

14

Column-oriented (Column Family)

- The column-oriented (columnar) databases store data as columns as opposed to rows as in RDBMS.
  - Typically, an RDBMS table has a few columns, but it would hold many records.
- A column-oriented database
  - can contain any number of columns, which can store any type of data
  - imposes minimal need for upfront schema definition and can easily accommodate newer columns as the data evolves.

15

15

Example

Data

EmployeeID	FirstName	LastName	Age	Salary
SM1	Anuj	Sharma	45	10000000
MM2	Anand		34	5000000
T3	Vikas	Gupta	39	7500000
E4	Dinesh	Verma	32	2000000

In RDBMS, the data may be stored as:

```
SM1,Anuj,Sharma,45,10000000
MM2,Anand,,34,5000000
T3,Vikas,Gupta,39,7500000
E4,Dinesh,Verma,32,2000000
```

In column-oriented databases, the data will be stored internally as:

```
SM1,MM2,T3,E4
Anuj,Anand,Vikas,Dinesh
Sharma,,Gupta,Verma,
45,34,39,32
10000000,5000000,7500000,2000000
```

16

Advantages

- Most of the solutions allow adding columns over time without having to worry about filling in default values for the existing rows for the new columns.
  - This gives flexibility in model and entity design allowing one to account for new columns in future for unforeseen scenarios and new requirements.
- Provide great performance in computing maxima, minima, averages and sums, specifically on large datasets.
- While inserting new values partial data access is allowed without touching unrelated columns which makes it much faster in execution.
- Since columns will be of uniform type and mostly of the same length, there are possibilities of efficient storage in terms of size.

17

17

Column Family Use Cases

- Event logging
  - Application errors can be stored in Cassandra
  - Column family stores
    - Application name
    - Event/Error
    - Timestamp
  - Enables data to be processed for analysis using Hadoop
- Blogging/user-generated content
  - Enables fast searching on tags

18

18

Column-oriented databases

- Open Source:
  - Apache Cassandra
  - Apache HBase
  - ClickHouse
  - CrateDB
  - Druid
  - MapD
  - MonetDB
  - Others
- Platform as a Service
  - Amazon Redshift
  - Microsoft Azure SQL Data Warehouse
  - Google BigQuery
  - Oracle Autonomous Datawarehouse Cloud Service
  - Scylla (database) Cloud
  - Snowflake Computing
- Proprietary
  - IBM Db2
  - memSQL
  - Microsoft SQL Server 2012
  - Oracle Exadata
  - Teradata
  - Others

19

19

Document Store

- A document store (also known as document-oriented database) allows the inserting, retrieving, and manipulating of semi-structured data:
  - XML, JSON, BSON, or YAML,
- Data access is typically over HTTP protocol using RESTful API or over Apache Thrift protocol for cross-language interoperability.
- The documents themselves act as records, however, it is semi-structured as compared to rigid RDBMS:
  - Two records may have completely different set of fields or columns.
  - The database may not support a schema or validating a document against the schema at all.
  - indexes can be created and queried.

20

20

Example

Document 1

```
{
"EmployeeID": "SM1",
"FirstName" : "Anuj",
"LastName" : "Sharma",
"Age" : 45,
"Salary" : 10000000
}
```

Document 2

```
{
  "EmployeeID": "MM2",
  "FirstName" : "Anand",
  "Age" : 34,
  "Salary" : 5000000,
  "Address" :
  {
    "Line1" : "123, 4th Street",
    "City" : "Bangalore",
    "State" : "Karnataka"
  },
  "Projects" : [ "nosql", "rdbms"]
}
```

21

21

Advantages

- Content is schemaless, or at best loosely defined.
  - This is very useful in web-based applications where there is a need for storing different types of content that may evolve over time.
- Searching across multiple entity types is easier than in traditional RDBMS or even in column-oriented databases.
  - Everything inside a document is automatically indexed when a new document is added.
  - Even if a document structure is complex, a document store search API can remain simple and provides an easy way to select a document or subset of a document.

22

22

Document Data Store Use Cases

- Applications that are storing data in flat files
  - Document data stores provide structured search functionality
- Data is too complex to model in a relational database
- Application has high-volume “matching” of records
  - Trade clearing
  - Fraud detection
  - Transaction reconciliation

23

23

Document-oriented databases

- Amazon DocumentDB
- BSON
- Cosmos DB
- Apache CouchDB
- MongoDB
- OrientDB
- RethinkDB
- SciDB

24

24

Discussion Question

- What would be the drawbacks of using column-oriented databases?
- Give an example of the use case where you would use Document-based database.

25

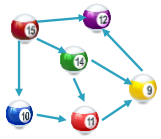
25

Part 3

26

26

Graph Data Stores



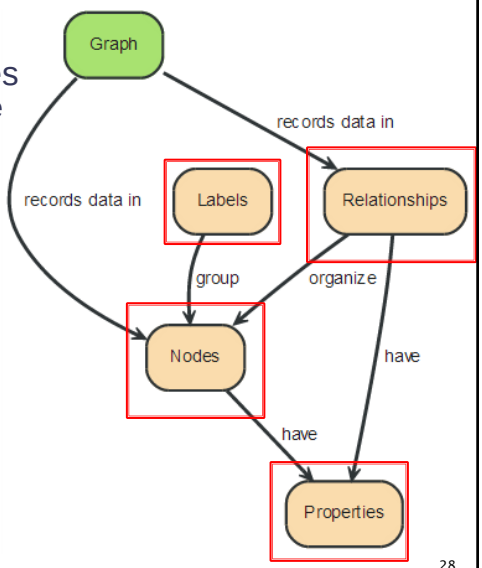
- Graph databases represent a special category of NoSQL databases where relationships are represented as graphs.
- Graph databases apply graph theory to the storage of information about the relationships between entities.
- There can be multiple links between two nodes in a graph—representing the multiple relationships that the two nodes share.
- The relationships represented may include social relationships between people, transport links between places, or network topologies between connected systems.

27

27

Graph architecture

- A Graph records data in Nodes which have Properties
  - A Node could start with a single Property and grow to a few million Properties
- Nodes are organized by Relationships which also have Properties
- Nodes are grouped by Labels into Sets
- A traversal navigates a Graph; it identifies Paths which order Nodes

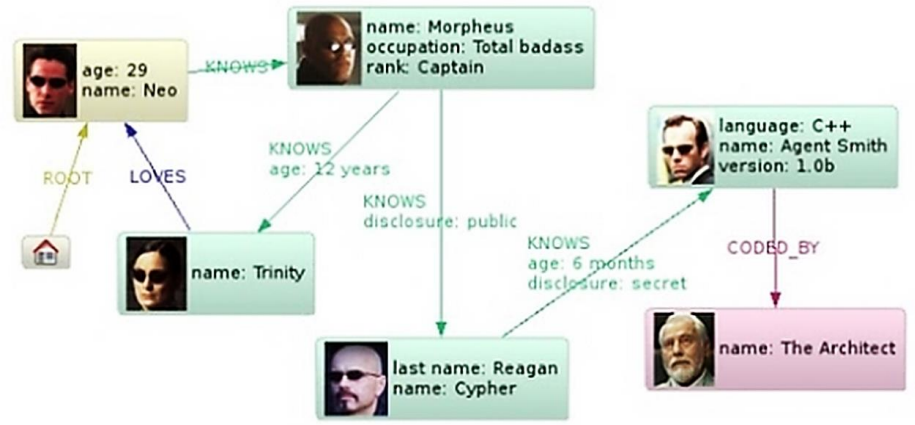


http://docs.neo4j.org/chunked/milestone/what-is-a-graphdb.html

28

28

Example



From an InfoQ article by Peter Neubauer
http://www.infoq.com/articles/graph-nosql-neo4j

29

29

Advantages

- Easy representation, retrieval and manipulation of relationships between the entities in the system.
- Graph databases can be considered as special purpose NoSQL databases optimized for relation-heavy data.
  - If there is no relationship among the entities, there is no use case for graph databases.
- It is not uncommon to store data in a document store and relationships in a graph database.

30

30

Graph Use Cases

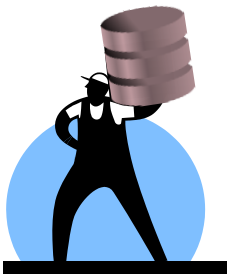
- Connected data
  - Represent employees and their skills
  - Projects worked on
- Dispatch services for retail delivery
  - Each delivery person and delivery is a node
  - Relationships can have distance property
- Recommendation engines
  - Any information application with users can make us of this
  - “Customer” nodes link to “hotels booked” nodes
  - Recommend similar hotels to customers based on previous purchase history

31

31

Graph Databases

- AgensGraph
- AllegroGraph
- ArangoDB
- Amazon Neptune
- Cayley
- FlockDB
- IBM Graph
- Neo4j
- Oracle Spatial and Graph
- OrientDB
- RedisGraph
- Titan
- Others



32

32

Polyglot Persistence

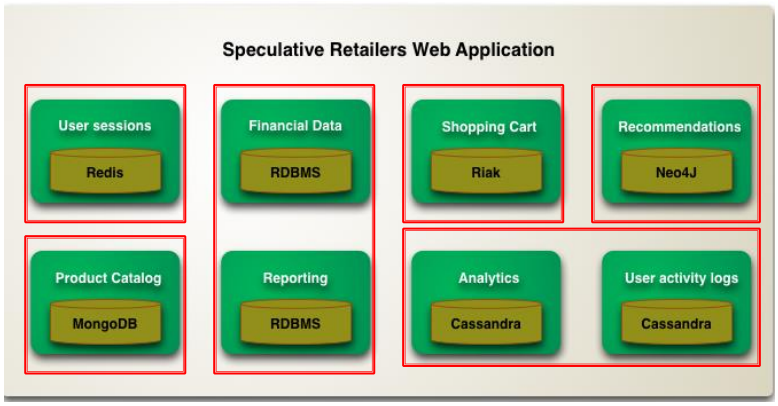
- **Polyglot Persistence:**
  - Using multiple data storage technologies, chosen based upon the way data is being used by individual applications
- Many modern enterprise systems use several databases together, leveraging their strengths to create an ecosystem that is more powerful, capable and robust than the sum of its parts.
- For example, an online retail Web application usually has the following storage requirements;
  - **Product catalog**: lots of reads, infrequent writes
  - **Financial data**: transaction requirements, reporting on orders, performance, etc.
  - **User sessions:** rapid access for read and write – no durability; user activity logs; high volume of writes
  - **Product recommendations**: links between products, people
  - **Analytics**: examine customer patterns

Sadalage, P., Fowler M, NoSQL Distilled, Addison-Wesley, 2012
https://sulmere.tumblr.com/PolyGlot

33

33

Polyglot Persistence – example



https://martinfowler.com/bliki/images/polyglotPersistence/polyglot.png

34

34

Discussion Questions

- Give an example of the use case where you would use Graph database.
- Can you provide another example of the use case of Polyglot Persistence with various RDBMS and NoSQL databases involved?

35

35

Essentials

- ✓ Discussed different NoSQL storage types
- ✓ Introduced key-value stores
- ✓ Discussed column-oriented stores
- ✓ Described document-oriented data stores
- ✓ Introduced graph databases
- ✓ Discussed Polyglot Persistence

36

36