UNIVERSITÀ DI TORINO

NoSQL databases for the Web

Prof. Fabio Ciravegna
Dipartimento di Informatica
Università di Torino
fabio.ciravegna@unito.it

UNIVERSITÀ DI TORINO

Scaling up to Web Size

- Large providers such as social media providers need to scale up their infrastructure
  - both physical (number of servers) and software
- Their infrastructure is composed of hundreds of thousands of physical servers
  - Failure of nodes is expected, rather than exceptional
  - They require to build in backup and failover.
  - The number of nodes in a cluster is not constant
- We will see the details in the lecture on Search Engines

Limits of SQL databases

- We need to define structure and schema of data first and then only we can process the data
- They are designed for the old mainframe world
  - They provides consistency and integrity of data
    - Useful in e.g. a banking system
    - But a significant performance overhead with large distributed data
- They require vertical scaling (i.e. increasing resources to the existing machine)

UNIVERSITÀ DI TORINO

ctd

- They are designed for a world where the data is to be mapped into a predefined structure
  - Most applications store their data in JSON format
    - which is flexible by design
  - Conversion of data has an enormous overhead in applications with high throughput
    - in one of my applications with 1 million users sending location data at high velocity
      - conversion from JSON to relational data was the single bottleneck that caused severe pain and required a large and highly parallel architecture
- Join operations are the deathbed of efficiency

No SQL databases

- Do not enforce a strict schema
- Provide
  - Native sharding for horizontal scaling
    - i.e. distribution of data on multiple machines along cloud computing paradigm of flexibility
      - If you need more resources, just add more nodes (as opposed to upgrade the server size)
      - if you need fewer resources, remove some nodes
  - Auto replication of data which in case of failure will return to the last consistent state
    - a requirement in large computing centres where failure is an expected situation

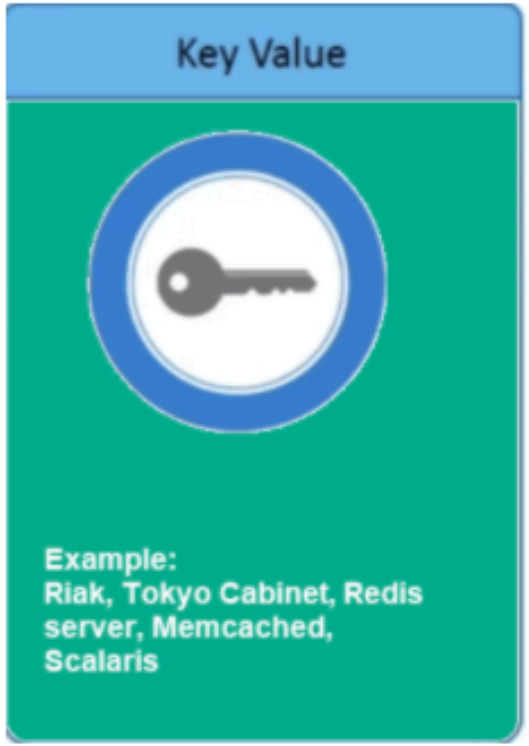
ctd

- Eventual Consistency
  - copies of data are stored on multiple machines to get high availability and scalability
  - Changes made to any data item on one machine has to be propagated to other replicas (and will eventually be propagated)
- BASE: Basically Available, Soft state, Eventual consistency
  - Basically, available means DB is available all the time
  - Soft state means even without an input; the system state may change
  - Eventual consistency means that the system will become consistent over time
- No single point of failure

Issues

- No standardisation for data into relations
  - Freedom but also a damning feature if overused
- Limited query capabilities
- No automatic consistency checking
  - e.g. when multiple transactions are performed simultaneously
- Eventual consistency is not appropriate for every application

Types



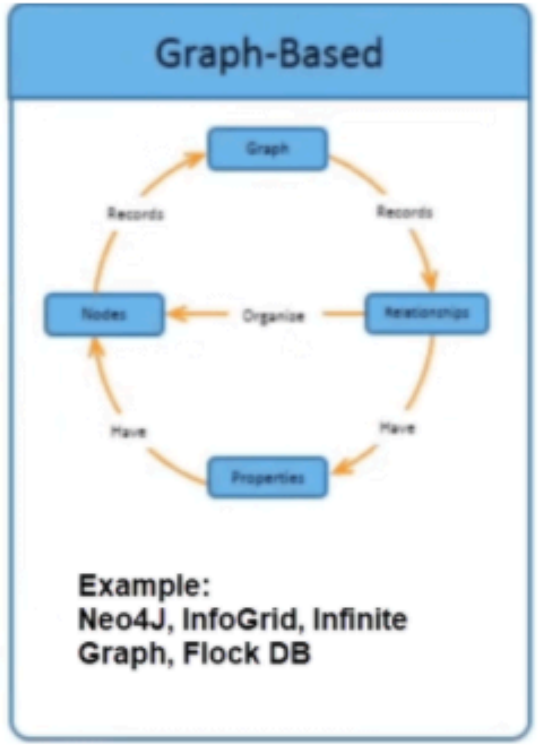


image from guru99

Types of NoSQL databases

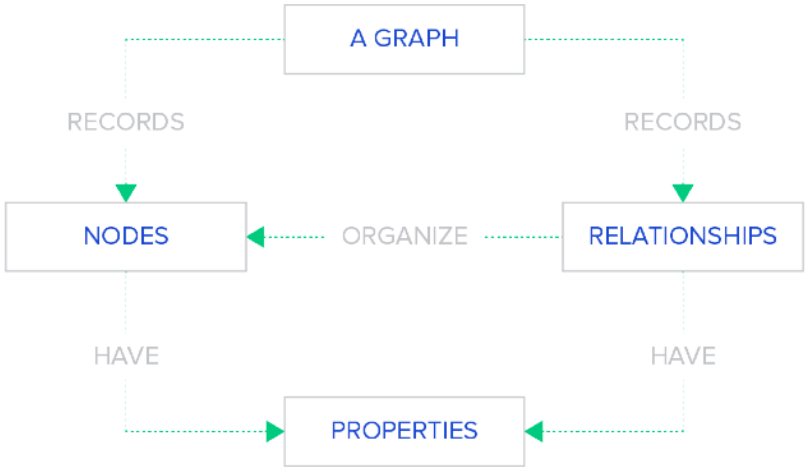
- Key value stores
  - Like hash tables

Key	Value
"Belfast"	{"University of Ulster, Belfast campus, York Street, Belfast, BT15 1ED"}
"Coleraine"	{"University of Ulster, Coleraine campus, Cromore Road, Co. Londonderry, BT52 1SA"}

- Document Oriented
  - as in key values but data is stored as a key value pair
    - but the value is a document (e.g. composed of multiple fields)
    - stored in JSON or XML formats
    - this has a number o advantages in terms of querying

Types (cdt)

- Column Databases
  - data is stored in columns, as opposed to rows in SQL DBs
  - fast read/write access to the data stored
  - enable effective compression as columns are typically largely uniform
  - focus on querying on one aspect of the data (e.g. price over time) rather than the entire record (price of company x between two periods)
  - created by Google for their core search engine storage system
- Graph Databases
  - a directed graph structure is used to represent the data
  - graphs are composed of edges and nodes
  - typically used in social networking applications.
  - Graph databases allow developers to
    - focus more on relations than on objects



UNIVERSITÀ DI TORINO

	Storage Type	Query Method	Interface	Programming Language	Open Source	Replication
Cassandra	Column Store	Thrift API	Thrift	Java	Yes	Async
MongoDB	Document Store	Mongo Query	TCP/IP	C++	Yes	Async
HyperTable	Column Store	HQL	Thrift	Java	Yes	Async
CouchDB	Document Store	MapReduce	REST	Erlang	Yes	Async
BigTable	Column Store	MapReduce	TCP/IP	C++	No	Async
HBase	Column Store	MapReduce	REST	Java	Yes	Async

UNIVERSITÀ DI TORINO

What you should know

- Understand the limitations of SQL databases
- Understand the large scale requirements for the web
- Understand the motivation behind NoSQL databases
- Understand the types of noSQL databases

UNIVERSITÀ DI TORINO

Questions?

© Prof. Fabio Cavegna, Università di Torino