

NoSQL

Introduzione

Obiettivi

- Introdurre i concetti base del mondo NoSQL
- Scrivere semplici applicazioni con mongoDB
- Scrivere semplici applicazioni con CouchBase

Da SQL a NoSQL

- Problemi del database relazionale
 - Schema fisso
 - Scalabilità con grandi numeri di utenti
 - Partizionabilità geografica
- Nuovi requisiti
 - Elevatissimo numero di utenti
 - Cloud
 - Big Data

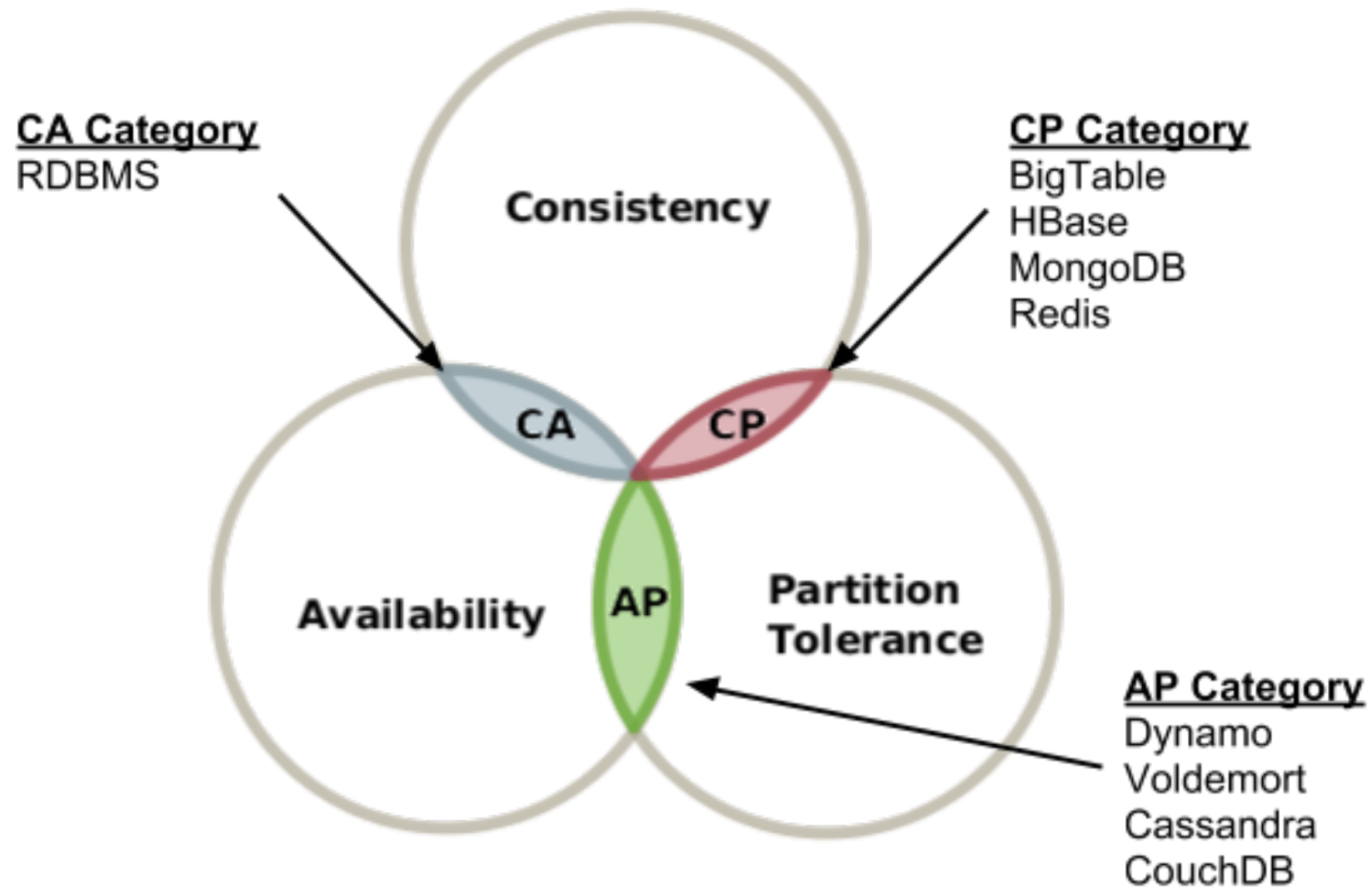
I primi NoSQL

- Bigtable (Google, 2006)
- Dynamo (Amazon, 2007)
- Facebook (Cassandra, 2008)
- LinkedIn (Voldemort, 2009)

Teorema CAP

- ***Consistency / Availability / Partition tolerance***
 - “Solo due tra C,A,P sono possibili”
- ***Consistency***
 - ogni lettura dà il dato più recente o un errore
- ***Availability***
 - tutte le letture non danno mai errore, ma non c'è garanzia del dato più recente
- ***Partition tolerance***
 - il sistema opera anche in caso di disconnessione tra nodi

Teorema CAP



Tipi di database NoSQL

- Chiave - valore
 - Memcached
- Struttura dati
 - Redis
- Documento
 - Couchbase, MongoDB
- Colonna
 - Cassandra
- Grafo
 - Neo4J, OrientDB

Chiave / valore

- Sono una grande mappa (key, value)
 - get(key)
 - set(key, value)
 - delete(key)
- Usati anche come *cache* del database relazionale

Documento

- Ogni record è un documento
- Non c'è schema fisso
 - si possono aggiungere campi solo a certi documenti
- Ogni documento individuato da una chiave unica

Colonna

- Sono organizzati a colonna invece che per riga
- A riga
 - 1 / Smith / Joe
 - 2 / Doe / Jane
 - 3 / Jones / Mary
- A colonna
 - 1 / 2 / 3
 - Smith / Doe / Jones
 - Joe / Jane / Mary

Grafo

- Rappresentano una rete di nodi ed archi
- Coppie (key, value) associate sia a nodi che ad archi
- Sono un documentale con in più i grafi

Database a documento

- Un record è un documento
- Ogni documento è associato ad una chiave unica
- Non c'è schema fisso
 - Ogni documento ha una struttura diversa
- Non ci sono limiti di complessità
- Formato JSON, XML, ...
- Il contenuto è indicizzato
- Sono possibili *query*
- *Sharding* automatico e repliche

mongoDB

- Database a documento JSON
- Open source (Apache License 2.0 + Server Side Public License)
- Versioni Community e Enterprise
- Caratteristiche principali
 - Indicizzazione
 - Replica
 - Load balancing (con *sharding*)
 - *Map-Reduce*
 - Esecuzione di JavaScript
- Transazioni aggiunte solo di recente (4.0, 6/2018)

Couchbase

- Database a documento o a chiave / valore
- Open source (Apache License 2.0)
- Versioni Community e Enterprise
- *Non è CouchDB*
- Caratteristiche principali
 - Indicizzazione
 - Replica
 - Ha una cache nativa
 - *Sharding* automatico

Esercizi con mongoDB

- Lanciare il server dockerizzato
 - `docker run --name mongoddb -d -p 27017:27017 mongo:4.1.5`
- Verificare l'indirizzo IP del server
 - `docker network inspect bridge`
 - Supponiamo sia **172.17.0.2**
- Lanciare il client dockerizzato
 - `docker run -it --name mongoddb-client -v $HOME:/home/developer mongo:4.1.5 /bin/bash`
- Dentro il container, lanciare il client
 - `mongo 172.17.0.2`

Esercizi con mongoDB

- Popolare dei dati
 - `mongo 172.17.0.2 < populate`
- Lanciare `MongoDemo.java`

Esercizi con Couchbase

- Lanciare il server dockerizzato
 - `docker run --name couchbase -d -p 8091-8094:8091-8094 -p 11210:11210 couchbase:community-6.0.0`
- Connettere Firefox to <http://localhost:8091>
- “Setup new cluster”
 - Cluster Name: **default**
 - Create Admin Username: **Administrator**
 - Create Password: **adminadmin**

Esercizi con CouchBase

- Proseguire con i wizard lasciando i default
- Andare sulla pagina dei sample buckets e aggiungere **travel-sample**

Esercizi con CouchBase

- “Add user”
 - name: **demo**
 - description: **demo**
 - password: **demodemo**
- roles
 - travel-sample: application access

Esercizi con CouchBase

- Inserire le credenziali in application.properties
- mvn install
- mvn springboot:run

Riepilogo

- Domande & risposte