Bazy danych NoSQL
Część II

Maciej Zakrzewicz

Politechnika Poznańska, Instytut Informatyki, http://zakrzewicz.pl

Apache Cassandra

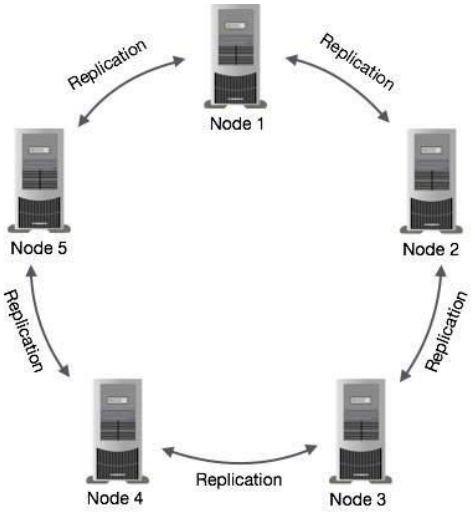
- Rozproszony, zdecentralizowany, kolumnowy (column-oriented) system bazy danych
- Inspirowany przez Google Bigtable, opracowany w Facebook (udostępniony w 2008)
- Wydajna obsługa bardzo dużych wolumenów danych strukturalnych
- Skalowalność, wysoka niezawodność, brak pojedynczego punktu awarii (ang. single point of failure)
- Stosowany przez m.in. Facebook, Twitter, Cisco, Rackspace, ebay, Twitter, Netflix

Architektura rozproszona

- Wszystkie węzły odgrywają taką samą rolę
- Każdy węzeł jest niezależny
- Każdy węzeł może przyjmować żądania zapisu i odczytu danych, niezależnie od fizycznej lokalizacji danych
- Po awarii węzła, jego zadania mogą realizować inne węzły

Replikacja

- Dane podlegają rozproszeniu i replikacji pomiędzy węzłami
- Po wykryciu, że węzeł posiada nieaktualne dane, Cassandra odnajdzie i zwróci najbardziej aktualny obraz danych, a następnie przystąpi do zaktualizowania danych nieaktualnych (read repair)
- Węzły są zorganizowane w ring
- Konfiguracja w `cassandra.yaml`



[tutorialspoint.com]

Twierdzenie Brewera - CAP

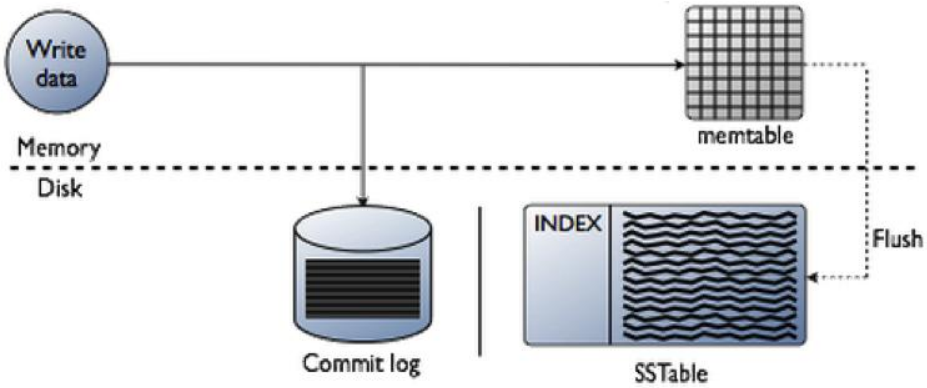
- W dużym systemie rozproszonym możliwe jest spełnienie tylko dwóch z trzech ważnych charakterystyk:
  - Consistency (spójność) – wszystkie węzły mają dostęp do jednakowych, aktualnych danych
  - Availability (dostępność) – każde żądanie doczeka się odpowiedzi
  - Partition Tolerance (odporność na podział) – system potrafi działać pomimo utraty niektórych węzłów

Konfiguracja poziomu spójności

- Własnośc Consistency (spójność) jest konfigurowalna:
  - ONE – zwraca dane z najbliższej repliki
  - QUORUM – zwraca najświeższy stan danych spośród większości węzłów (replik)
  - ALL – zwraca najświeższy stan danych spośród wszystkich węzłów (replik)

Składniki architektury

- Operacje zapisu wykonywane są w buforze *memtable* w pamięci operacyjnej, zabezpieczonym natychmiastowymi zapisami w dzienniku *commit log*
- Gdy rozmiar *memtable* przekrocza zadany poziom,jego dane zostają zapisane w pliku dyskowym *SSTable*
  - szybkie zapisy sekwencyjne
  - rozproszone, replikowane
- Węzeł: miejsce składowania danych
- Centrum danych: kolekcja powiązanych węzłów
- Klaster: jedno lub wiele centrów danych

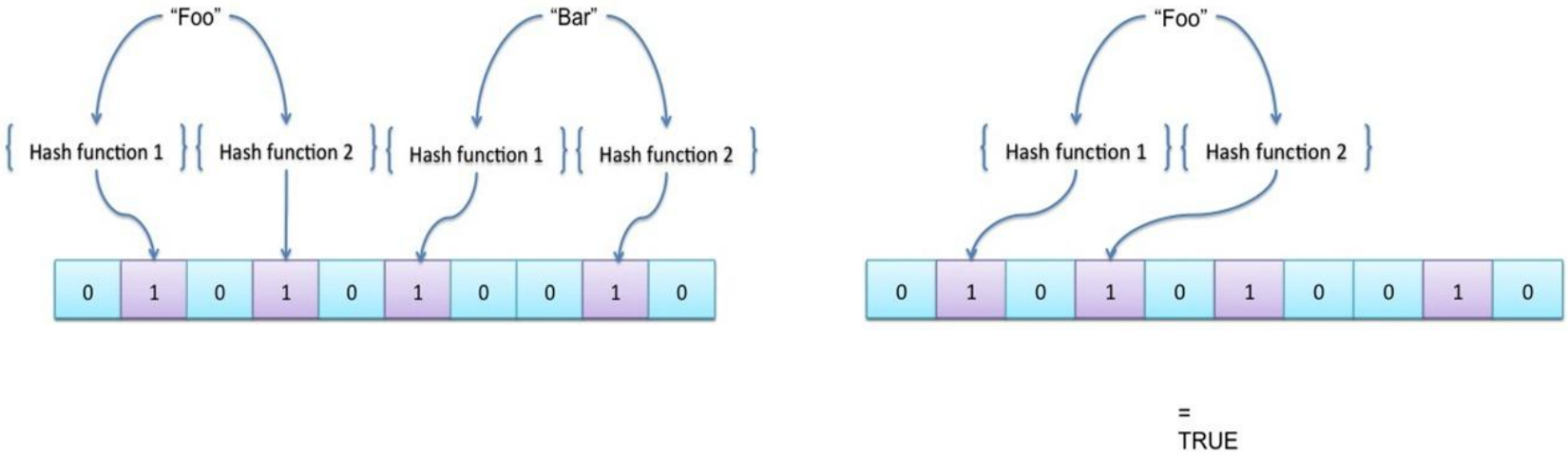


[datastax.com]

Filtr Blooma

- Wykorzystywany przez Cassandrę do szybkiego sprawdzenia, czy plik SSTable zawiera wiersz o podanej wartości klucza
- Filtr Blooma (B.H. Bloom, 1970) pozwala wstępnie stwierdzić, czy element należy do zbioru
  - możliwe fałszywe trafienia pozytywne (false positives)
  - niemożliwe pominięcia (false negatives)
- Ma postać wektora bitowego, w którym oznaczane są wartości funkcji haszowych przetwarzających wartości klucza
- Dokładność regulowana m.in. przez rozmiar wektora
  - `ALTER TABLE emp_data WITH bloom_filter_fp_chance = 0.1;`

Filtr Blooma – przykład działania



[kellabyte.com]

Partycjonowanie danych

- Partycjonowanie (i replikowanie) danych pomiędzy węzłami obsługiwane przez *Partitioner*
  - rodzaj funkcji haszowej
- Dostępne *Partitionery* (wybór w pliku konfiguracyjnym `cassandra.yaml`)
  - Murmur3Partitioner (domyślny): równomiernie rozprasza dane za pomocą funkcji haszowej MurmurHash
  - RandomPartitioner: równomiernie rozprasza dane za pomocą funkcji MD5
  - ByteOrderedPartitioner: rozprasza dane w sposób uporządkowany w oparciu o wartość klucza

Narzędzia Cassandra

- `cassandra` - serwer bazy danych Cassandra
- `cqlsh` - klient Cassandra umożliwiający interakcyjne wykonywanie poleceń w języku CSQL

Przestrzeń kluczy - Keyspace

- Kontener przechowujący dane
  - współczynnik replikacji – liczba węzłów przechowujących kopie tych samych danych
  - strategia lokalizacji replik – sposób rozlokowania replik w ringu, np. *simple strategy*, *old network topology strategy*, *network topology strategy*
- Przestrzeń kluczy zawiera listę *rodzin kolumn*
  - rodzina kolumn zawiera kolekcję *wierszy*
  - wiersz zawiera uporządkowane *kolumny*

Język CSQL

- **CREATE KEYSPACE** – tworzy przestrzeń kluczy
- **USE** – wybiera aktywną przestrzeń kluczy
- **ALTER KEYSPACE** – zmienia ustawienia przestrzeni kluczy
- **DROP KEYSPACE** – usuwa przestrzeń kluczy
- **CREATE COLUMNFAMILY | TABLE** – tworzy rodzinę kolumn w przestrzeni kluczy
- **ALTER COLUMNFAMILY | TABLE** – zmienia ustawienia grupy kolumn
- **DROP COLUMNFAMILY | TABLE** – usuwa grupę kolumn
- **TRUNCATE** – usuwa wszystkie dane z grupy kolumn
- **CREATE INDEX** – tworzy indeks na kolumnie grupy kolumn
- **DROP INDEX** – usuwa indeks

Język CSQL

- **INSERT** – wstawia wiersz do rodziny kolumn
- **UPDATE** – modyfikuje kolumnę w wierszu
- **DELETE** – usuwa wiersz z rodziny kolumn
- **BATCH** – wykonuje zestaw poleceń DML
- **SELECT** – odczytuje dane z rodziny kolumn
  - **WHERE** – dokonuje selekcji wierszy
  - **ORDER BY** – wskazuje kolejność odczytywania wierszy
  - do wyszukiwania w oparciu o kolekcje służy operator **CONTAINS**

Pozostałe polecenia CQLSH

- **CAPTURE** – umożliwia zapis wyniku zapytania do pliku
- **CONSISTENCY** – wyświetla/ustawia poziom spójności
- **COPY** – eksportuje zawartość tabeli do pliku
- **DESCRIBE KEYSPACES** – wyświetla istniejące przestrzenie kluczy
- **DESCRIBE TABLES** – wyświetla wykaz istniejących tabel w aktywnej przestrzeni kluczy
- **DESCRIBE TABLE** – wyświetla strukturę tabeli
- **DESCRIBE TYPES** – wyświetla wykaz typów zdefiniowanych przez użytkownika
- **DESCRIBE TYPE** – wyświetla strukturę typu zdefiniowanego przez użytkownika
- **SHOW** – wyświetla informacje o bieżącej sesji
- **SOURCE** – wykonuje polecenia z pliku

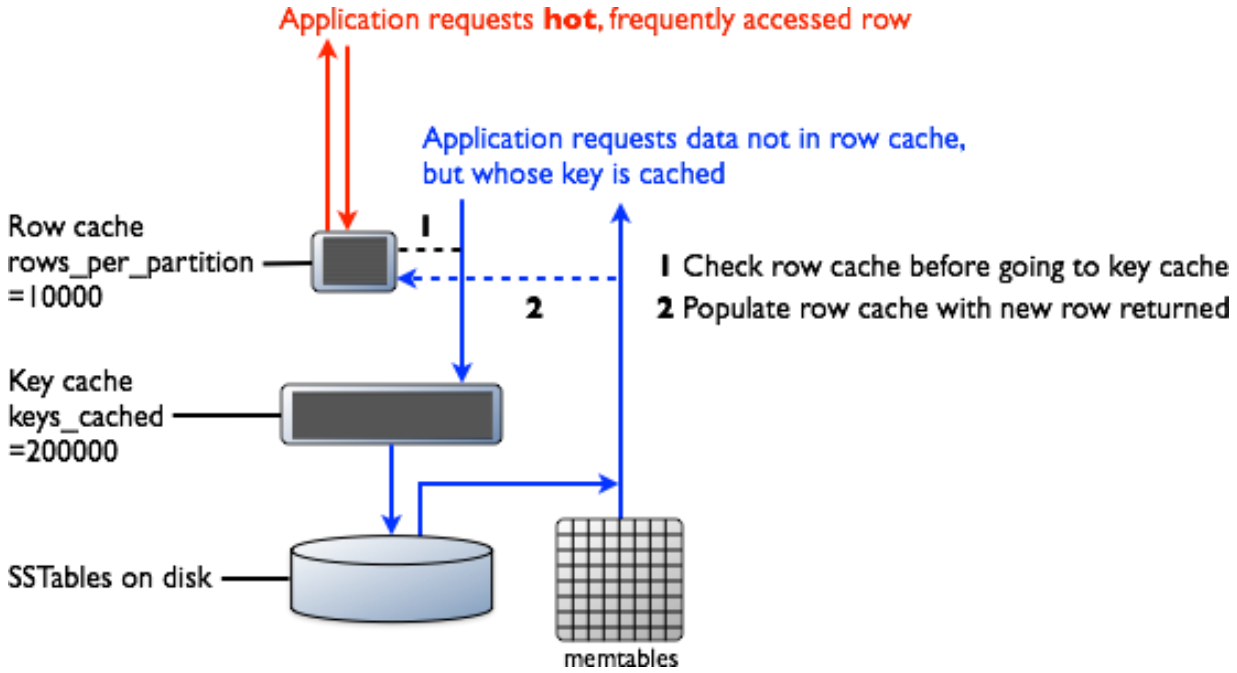
Tworzenie przestrzeni kluczy - język CQL

```
cqlsh> create keyspace Employees
with replication = {'class': 'SimpleStrategy',
                    'replication_factor' : 3};

cqlsh> describe keyspaces;

system_schema  system_auth  system  system_distributed
  system_traces  employees

cqlsh> use employees;
cqlsh:employees>
```

Rodzina kolumn – Column family

- Analogia tabeli
- Rodziny kolumn są definiowane, lecz ich kolumny mogą być dynamicznie dodawane w dowolnej chwili
- Posiada atrybuty:
  - keys_cached – liczba lokalizacji buforowania SSTable
  - rows_cached – liczba rekordów buforowanych w pamięci
  - preload_row_cache – czy bufor ma być wstępnie ładowany

Buforowanie SSTable



Kolumna

- Posiada trzy składowe:
  - nazwę
  - wartość
  - znacznik czasowy
- Superkolumna – specjalny rodzaj kolumny, zawiera kolekcję zagnieżdżonych kolumn

Typy danych

Data Type	Constants	Description
ascii	strings	Represents ASCII character string
bigint	bigint	Represents 64-bit signed long
blob	blobs	Represents arbitrary bytes
Boolean	booleans	Represents true or false
counter	integers	Represents counter column
decimal	integers, floats	Represents variable-precision decimal
double	integers	Represents 64-bit IEEE-754 floating point
float	integers, floats	Represents 32-bit IEEE-754 floating point
inet	strings	Represents an IP address, IPv4 or IPv6
int	integers	Represents 32-bit signed int
text	strings	Represents UTF8 encoded string
timestamp	integers, strings	Represents a timestamp
timeuuid	uuids	Represents type 1 UUID
uuid	uuids	Represents type 1 or type 4
		UUID
varchar	strings	Represents uTF8 encoded string
varint	integers	Represents arbitrary-precision integer

- Kolumny typu kolekcja:
  - set
    - np. set<text>
    - {'f@baggins.com', 'baggins@gmail.com'}
  - list
    - np. list<text>
    - ['rivendell', 'rohan']
  - map
    - np. map<timestamp,text>
    - { '2012-9-24' : 'enter mordor‘, '2012-10-2 12:00' : 'throw ring into mount doom' }

Tworzenie rodziny kolumn - język CQL

```
cqlsh:employees> create table emp_data(
                 id int,
                 firstname text,
                 lastname text,
                 primary key (id));

cqlsh:employees> insert into emp_data
                 (id, firstname,lastname)
                 values (1, 'John', 'Smith');
```

Zapisywanie/modyfikacja danych - CQL

```
cqlsh:employees> select * from emp_data;

 id | firstname | lastname
----+-----------+----------
  1 |      John |    Smith

cqlsh:employees> update emp_data set firstname='Susan', lastname='Brown'
                 where id=2;
cqlsh:employees> select * from emp_data;

 id | firstname | lastname
----+-----------+----------
  1 |      John |    Smith
  2 |     Susan |    Brown
```

update tworzy wiersz, jeśli nie istniał!

Operacje CAS (Compare and Set)

- Tzw. light transactions
  - ```
    insert into emp_data (id, firstname,lastname)
    values (1, 'John', 'Smith')
    if not exists;
    ```
  - ```
    update emp_data set lastname='Jones'
    where id=1
    if lastname='Smith';
    ```

Dodawanie/usuwanie kolumny – język CQL

```
cqlsh:employees> alter table emp_data
                 add salary float;

cqlsh:employees> alter table emp_data drop salary;
```

Tworzenie indeksu – język CQL

```
cqlsh:employees> create index f_ind
                 on emp_data(firstname);
```

(Indeks jest wymagany aby dokonywać selekcji w oparciu o kolumnę)

Wykonywanie kopii bezpieczeństwa

- Migawki (snapshots) plików SSTable
  - wszystkich przestrzeni kluczy
  - wybranej przestrzeni kluczy
  - wybranej rodziny kolumn
  - pełne lub przyrostowe
- Narzędzie `nodetool`:
  - `nodetool -h localhost -p7199 snapshot employees`

Odtwarzanie po awarii

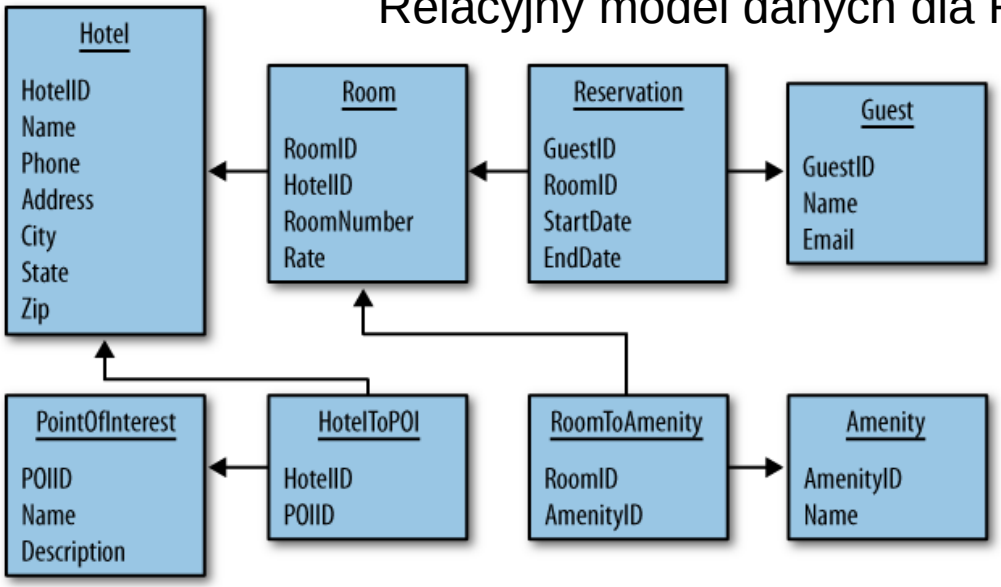
- Zatrzymaj węzeł
- Usuń wszystkie pliki z katalogu *commitlog*
- Usuń wszystkie pliki *.db w *data/keyspace_name/keyspace_name-*keyspace_name (lecz nie w katalogach */snapshots* i */backups*)
- Odnajdź najnowszy katalog migawki w *data/keyspace_name/table_name-UUID*/snapshots/*snapshot_name*
- Skopiuj jego zawartość do do katalogu *data/keyspace_name/table_name-UUID*
- Zrestartuj węzeł
- Uruchom `nodetool repair`

Monitorowanie serwera

- Współczynniki wydajnościowe:
  - `nodetool proxyhistograms`
  - `nodetool status`
  - `nodetool info`

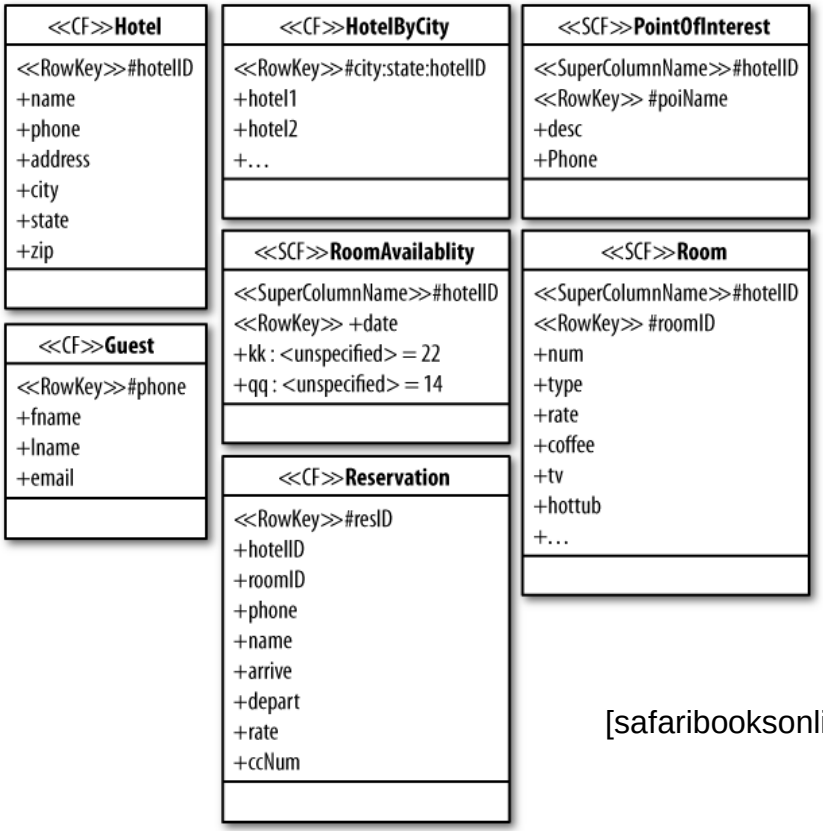
Modelowanie danych – Cassandra vs. RDBMS

Relacyjny model danych dla RDBMS



[safaribooksonline.com]

Modelowanie danych – Cassandra vs. RDBMS



Relacyjny model danych dla Cassandra

[safaribooksonline.com]

Casandra - dokumentacja

- http://docs.datastax.com
- http://www.tutorialspoint.com/cassandra/
- http://www.planetcassandra.org/
- https://www.safaribooksonline.com/library/view/cassandra-the-definitive/9781449399764/