

ZAPROGRAMUJ swoją przyszłość z GlobalLogic®



Konkurs
wygraj staż

1

Zgłoś swój udział

2

Wybierz projekt

3

Wygraj staż i nagrody



Rozpoczęcie konkursu 15 marca 2017
Targi Pracy, Wydział Mechaniczny
Uniwersytet Zielonogórski, ul. Szafrana 4.

www.globallogic.com | konkurs_zg@globallogic.com

Wprowadzenie do NoSql

Maksymilian Wiesiołek



Agenda

- O mnie,
- ACID a CAP,
- wstęp do NoSql
- PostgreSQL,
- Redis
- Hbase,
- MongoDB,
- Neo4j,

Agenda

- Coherence,
- Rozwiązania hybrydowe,
- Na co warto zwrócić uwagę,
- Zagrożenia,
- Testowanie,
- Co dalej?
- książki,
- Pytania odpowiedzi.

O mnie

Maksymilian Wiesiołek

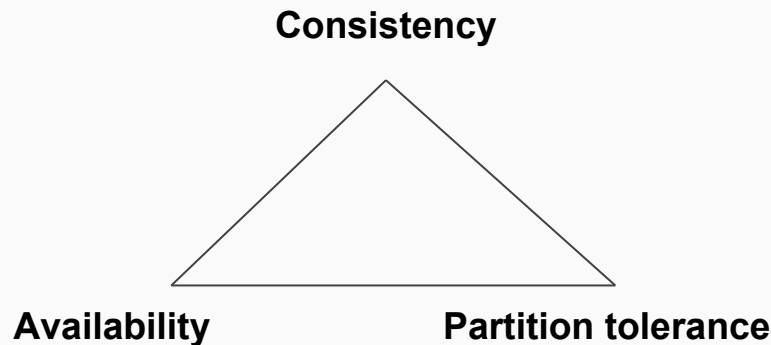
- programista z ponad 6 letnim doświadczeniem (BigData i aplikacje mobilne),
- Zainteresowania/hobby
 - BigData,
 - uczenie maszynowe (TensorFlow, Andrew Ng),
 - algorytmy i struktury danych (Robert Sedgewick),
 - światowy kryzys ekonomiczny (Inside Job; Liar's Poker, Michael Lewis),
 - HFT (Flash Boys, Michael Lewis),
 - magnetyzm (Walter Lewin),
 - squash.

ACID a CAP

ACID

- Atomicity (atomowość), wszystko albo nic,
- Consistency (spójność), zapewnienie spójności,
- Isolation (read uncommitted, read committed, repeatable read, serializable),
- Durability, trwałość i aktualność danych po zatwierdzeniu transakcji

CAP

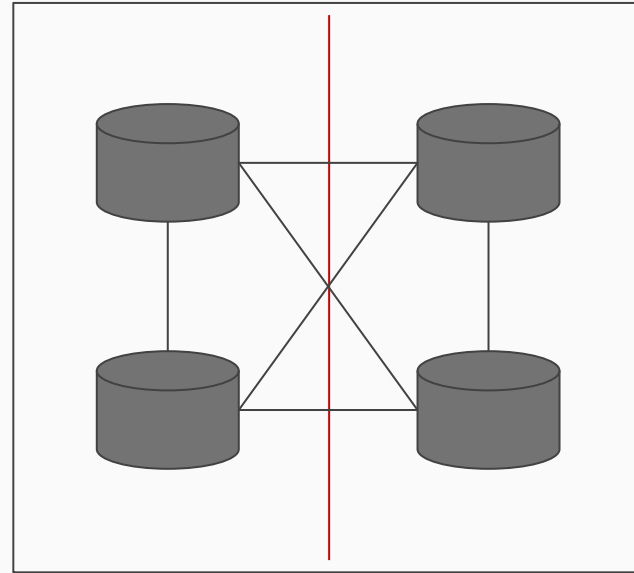


CAP

Partition tolerance + Availability (PA)

Partition tolerance + Consistency (PC)

Split brain

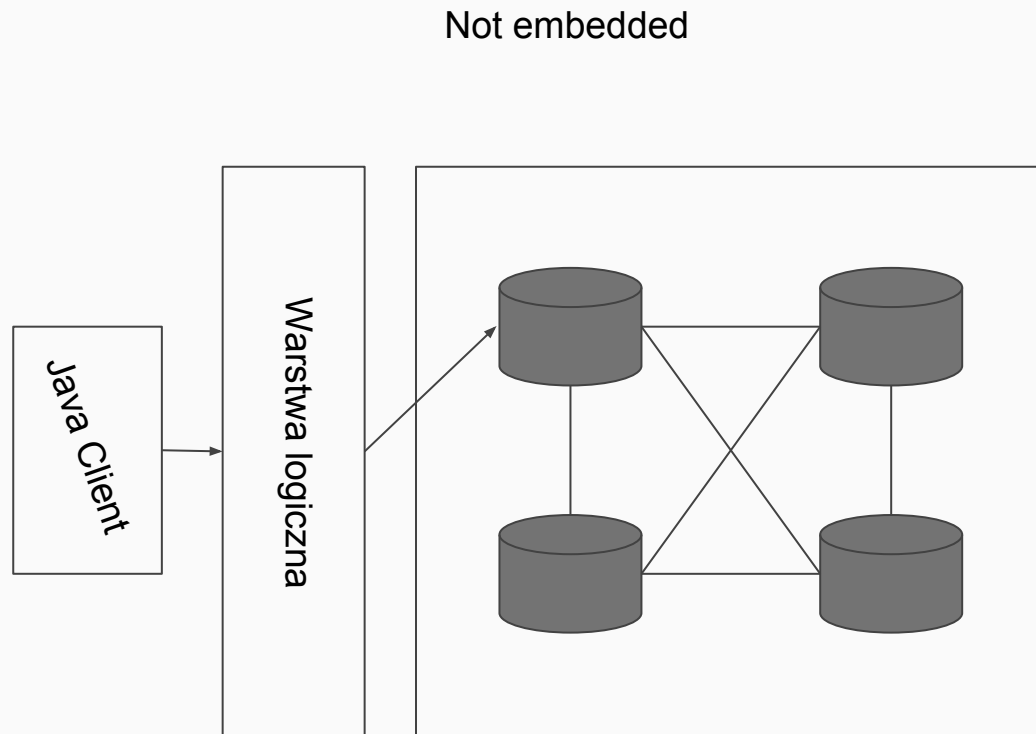


Wstęp do NoSql

- Architektura,
- Struktury danych,
- Scales out not up,
- fault tolerance,
- MapReduce/EntryProcessor/Aggregator,
- Garbage Collection.

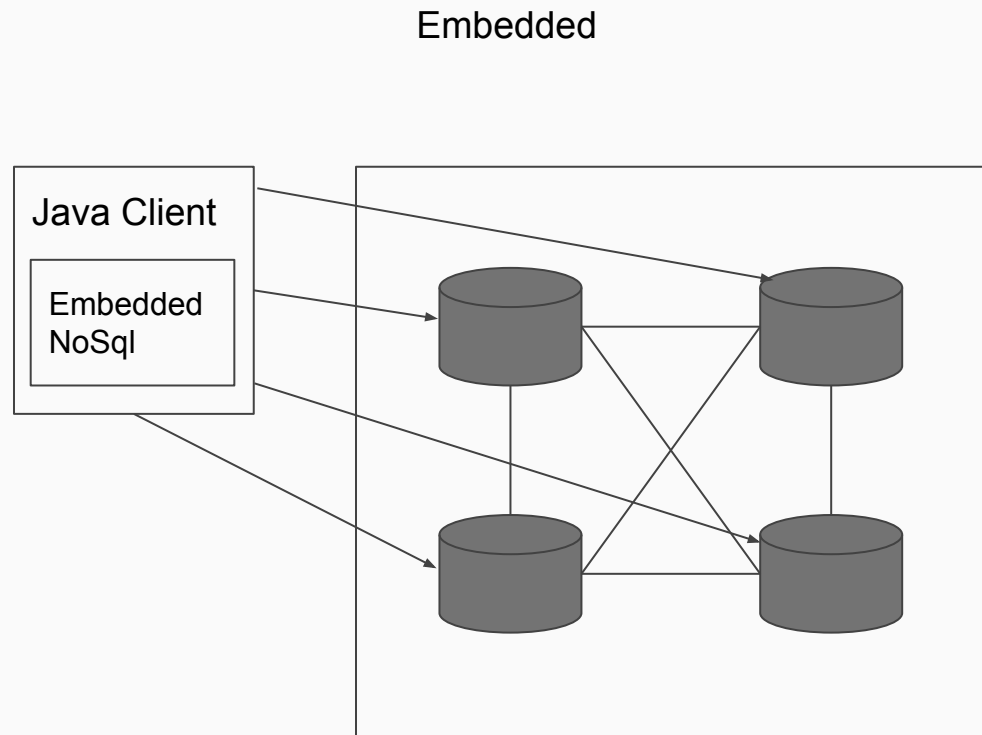
Wstęp do NoSql

Architektura



Wstęp do NoSql

Architektura



Wstęp do NoSql

Struktury danych

Wstęp do NoSql

Struktury danych

Kolumnowa

Struktury danych

Kolumnowa

- BigTable (Google),
- Cassandra,
- HBase (!)
 - atomowość na poziomie wiersza

	Keys (id)	family "user"	Family "product"
row (user)	1	first_name: "..." surname: "..." group: "..."	id: "..." name: "..." number: "..."
row (user)	2		

Wstęp do NoSql

Struktury danych

Grafy

Struktury danych

Grafy

- Neo4j

Wstęp do NoSql

Struktury danych

Dokument (Json)

Struktury danych

Document (Json)

- MongoDB
 - atomowość poprzez blokowanie całego dokumentu,

```
{
  users: [
    user: {
      id: 1,
      first_name: "Jan",
      surname: "Kowalski"
    },
    user: {
      id: 2,
      first_name: "John",
      surname: "Doe"
    }
  ]
}
```

Struktury danych

Document (Json)

- MongoDB

<https://github.com/mongodb/mongo-java-driver/blob/1d2e6faa80aeb5287a26d0348f18f4b51d566759/src/main/com/mongodb/ConnectionStatus.java#L213>

```
try {  
    .....  
} catch (Exception e) {  
    if (!((_ok) ? true : (Math.random() > 0.1))) {  
        return res;  
    }  
    .....  
}
```

- CouchDb

Wstęp do NoSql

Struktury danych

Key-Value

Struktury danych

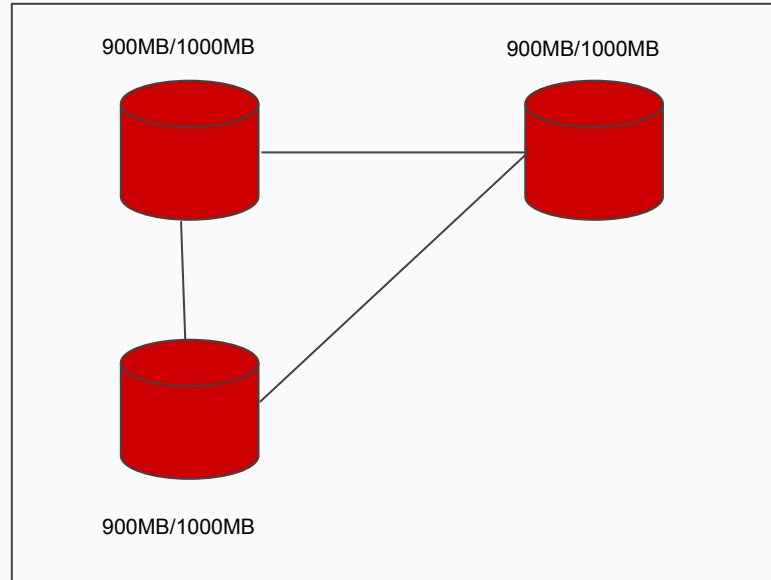
Key-Value

- Riak,
- Redis,
- Coherence (Hazelcast),

Wstęp do NoSql

Scales out not up

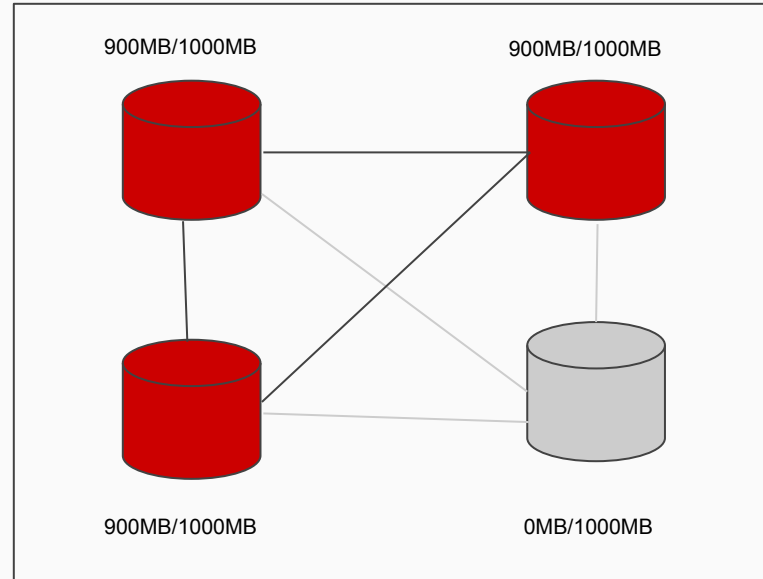
- Skalowalność,
- większa infrastruktura,



Wstęp do NoSql

Scales out not up

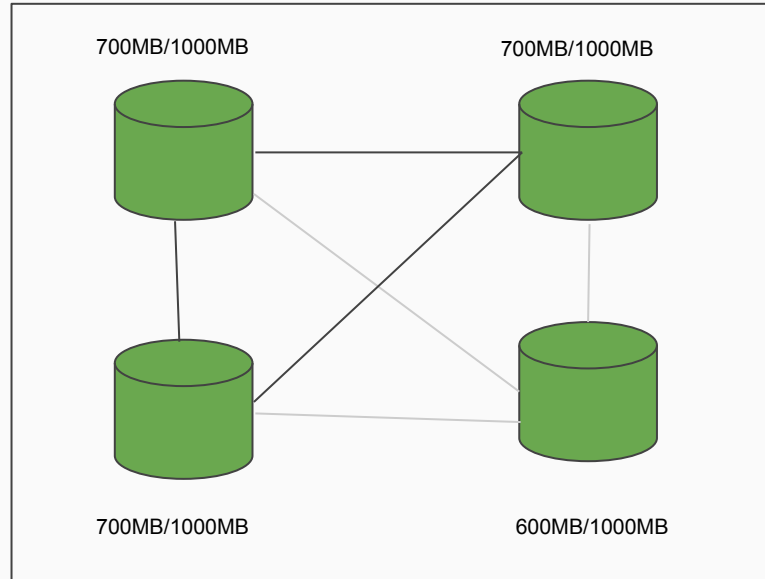
- Skalowalność,
- większa infrastruktura,



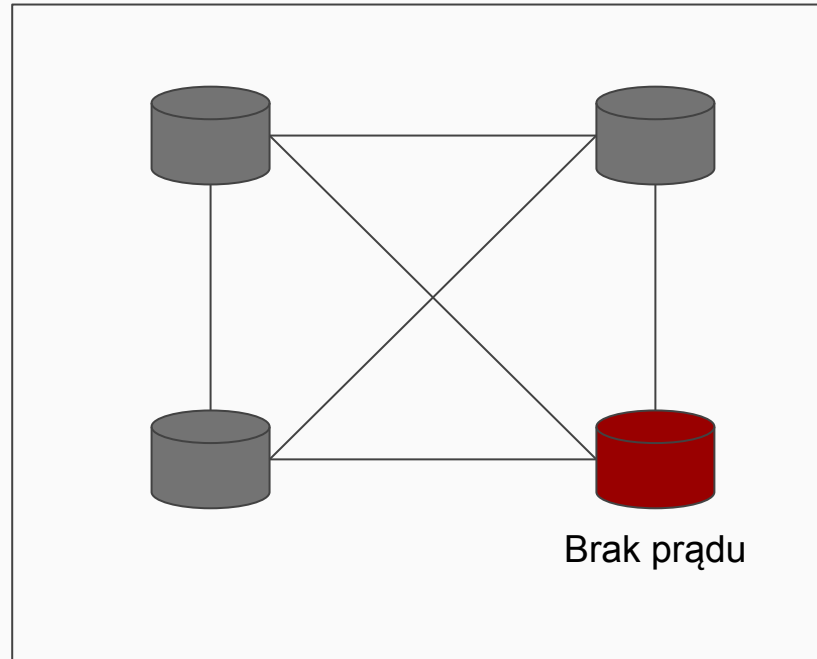
Wstęp do NoSql

Scales out not up

- Skalowalność,
- większa infrastruktura,

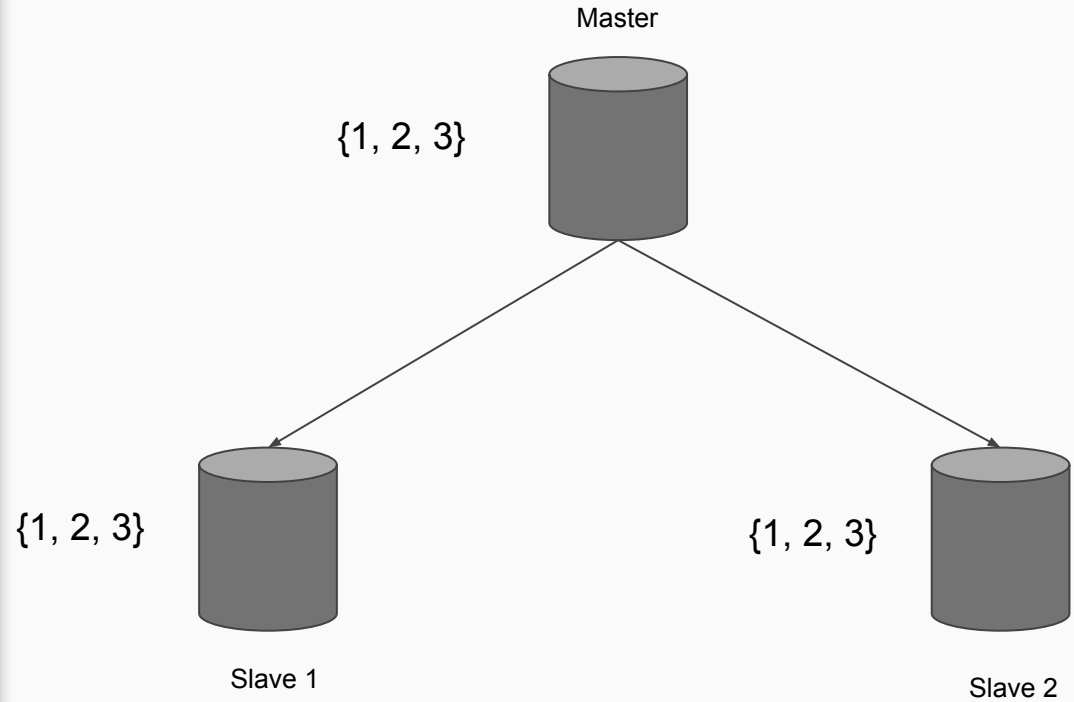


Fault tolerance



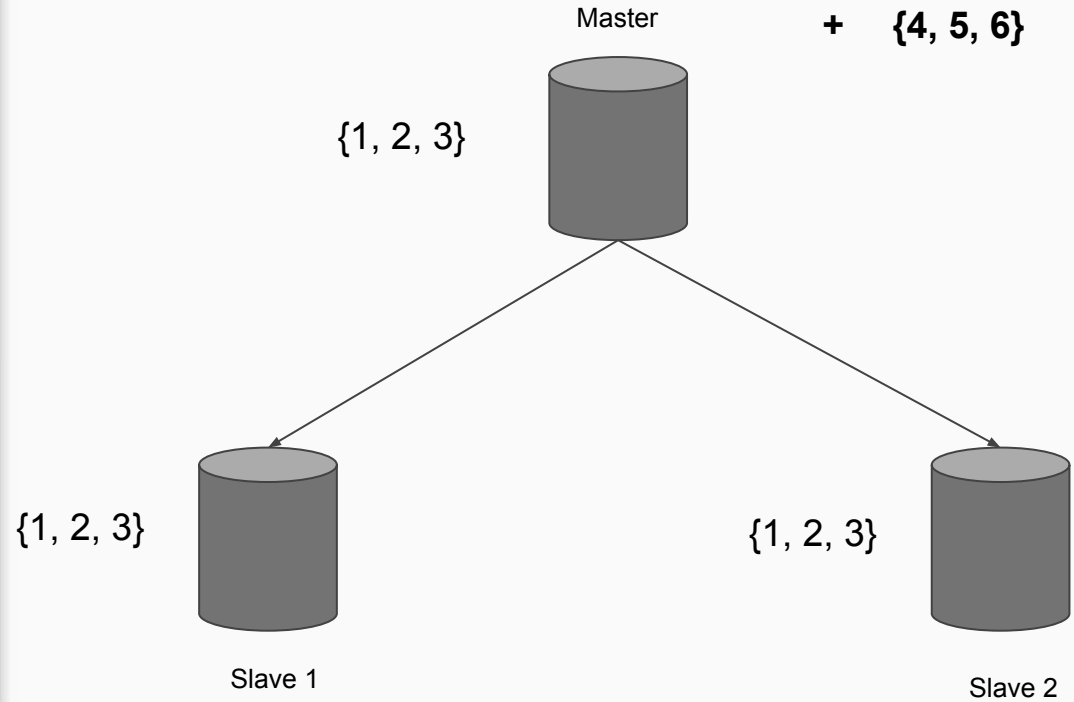
Wstęp do NoSql

Fault tolerance - replikacja
master/slave



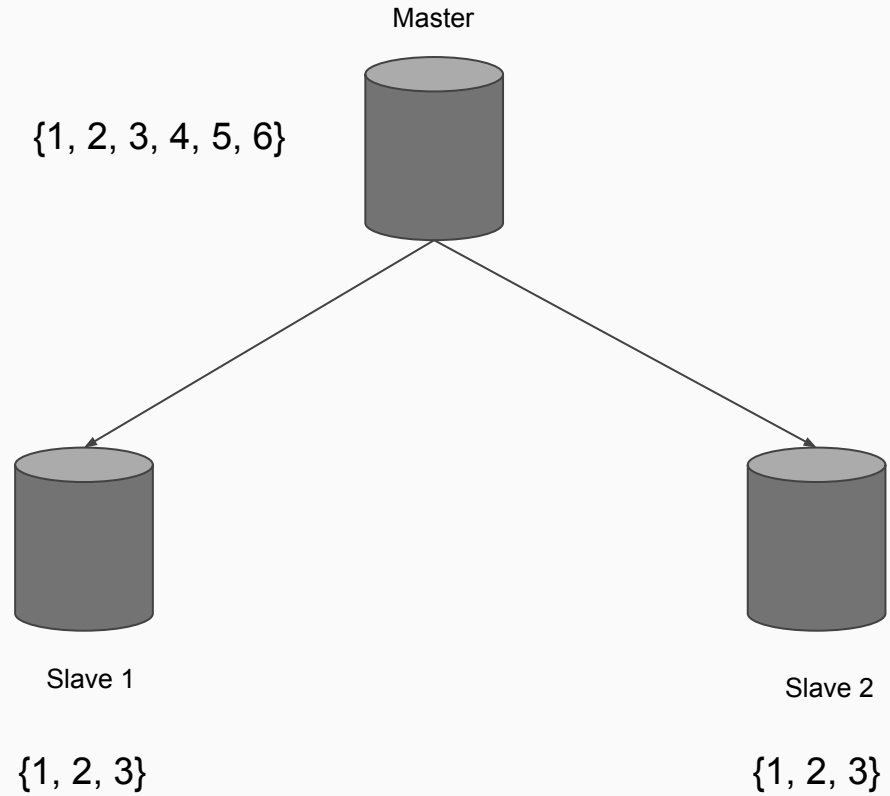
Wstęp do NoSql

Fault tolerance - replikacja
master/slave



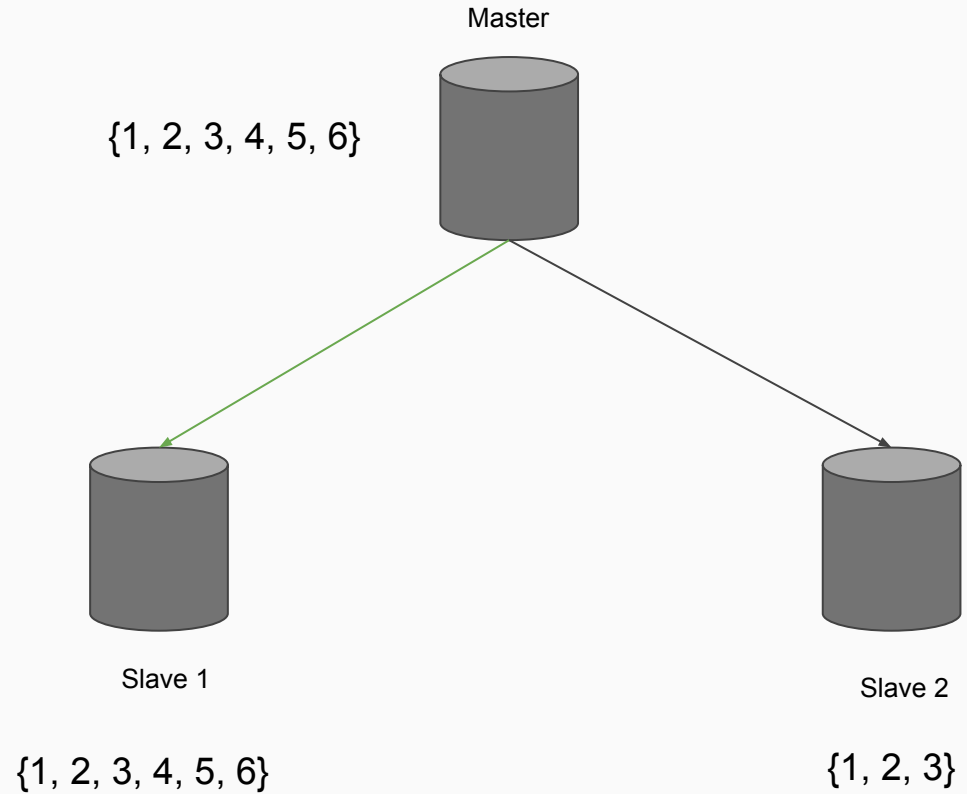
Wstęp do NoSql

Fault tolerance - replikacja
master/slave



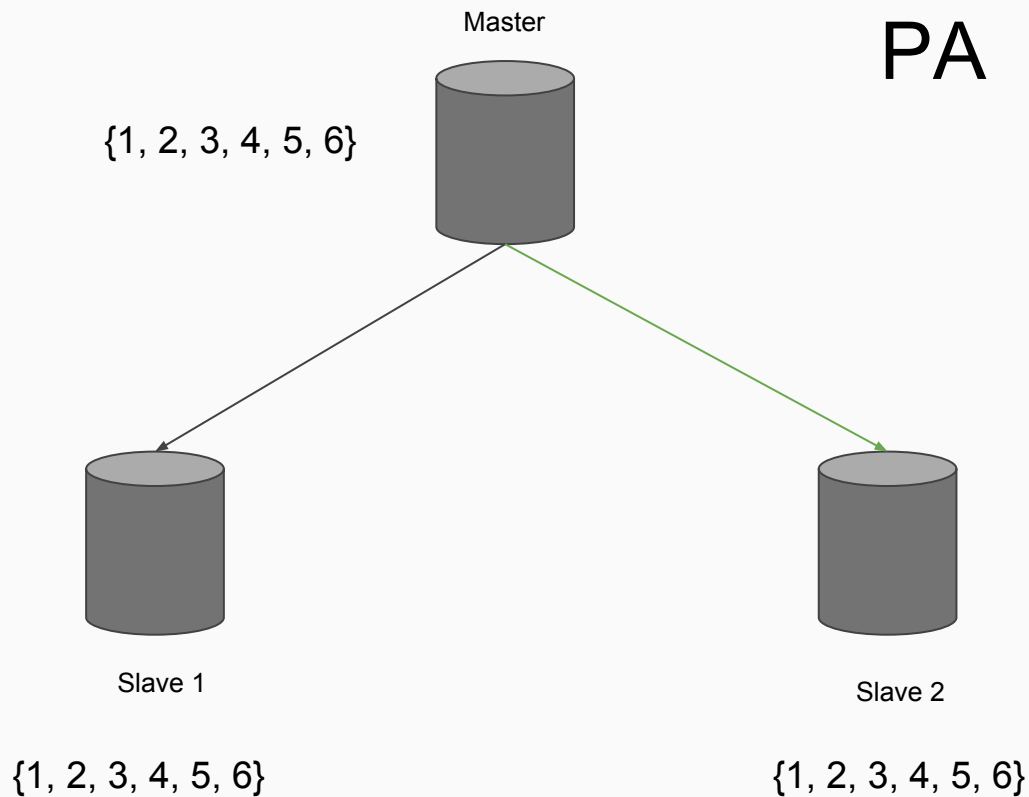
Wstęp do NoSql

Fault tolerance - replikacja
master/slave



Wstęp do NoSql

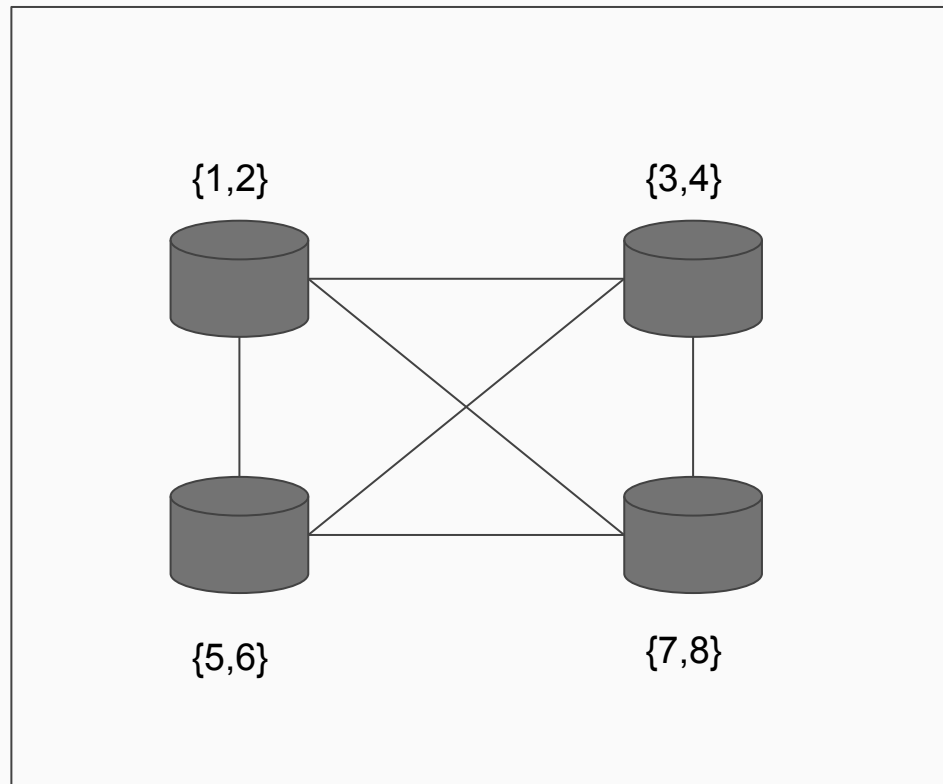
Fault tolerance - replikacja
master/slave



Wstęp do NoSql

Fault tolerance - horizontal partitioning
(partitioned cache)

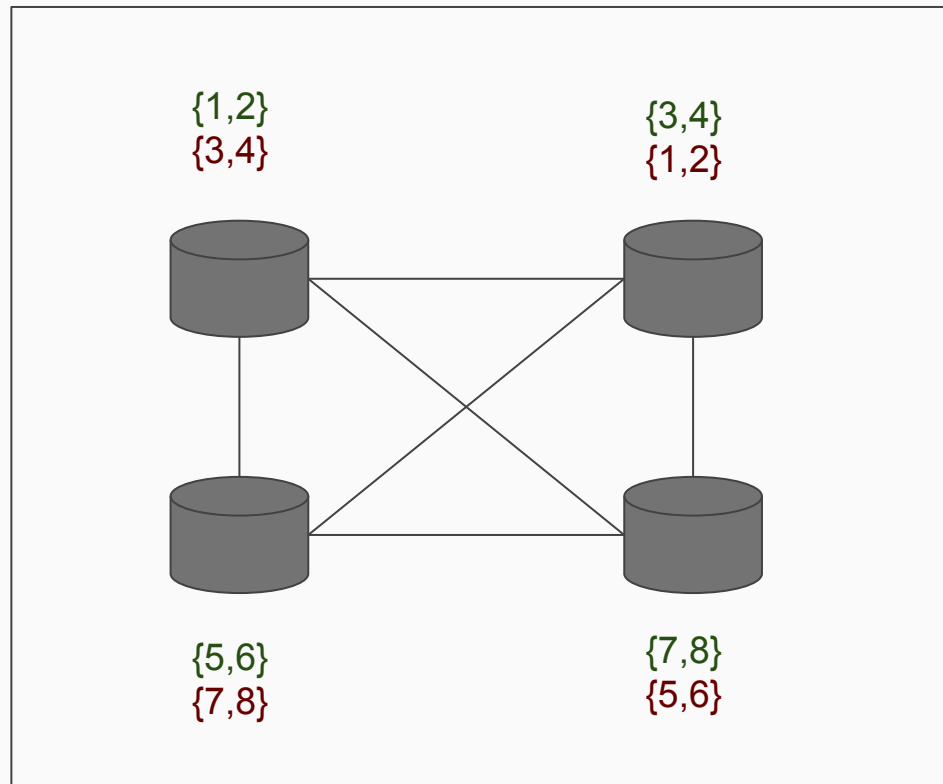
Zbiór danych: {1,2,3,4,5,6,7,8}



Wstęp do NoSql

Fault tolerance - horizontal partitioning
(partitioned cache)

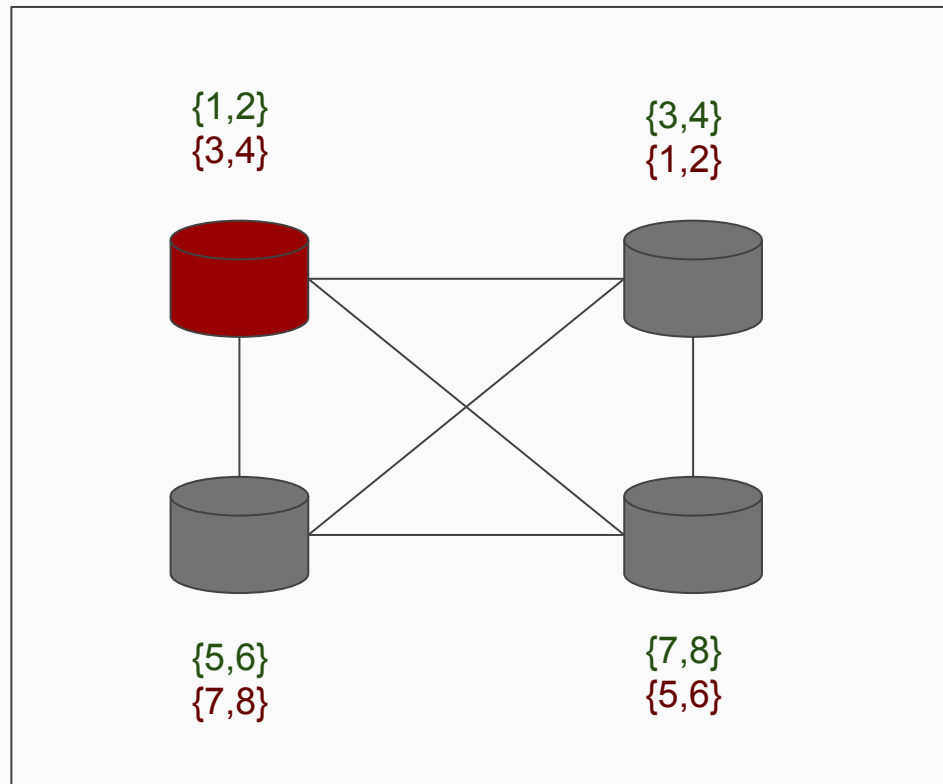
Zbiór danych: {1,2,3,4,5,6,7,8}



Wstęp do NoSql

Fault tolerance - horizontal partitioning
(partitioned cache)

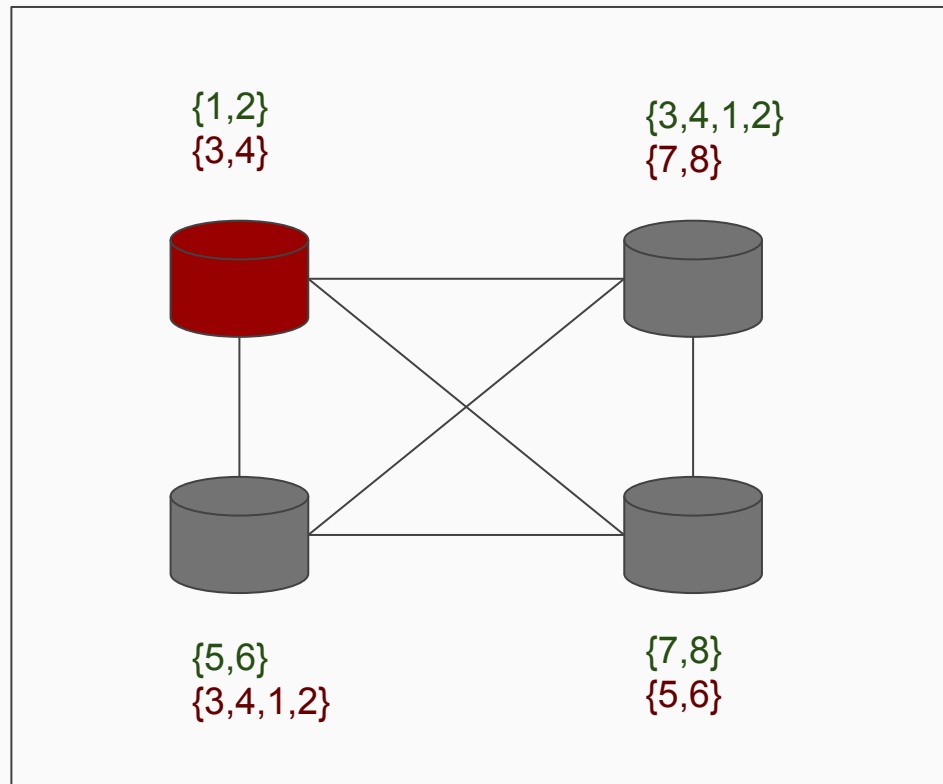
Zbiór danych: {1,2,3,4,5,6,7,8}



Wstęp do NoSql

Fault tolerance - horizontal partitioning
(partitioned cache)

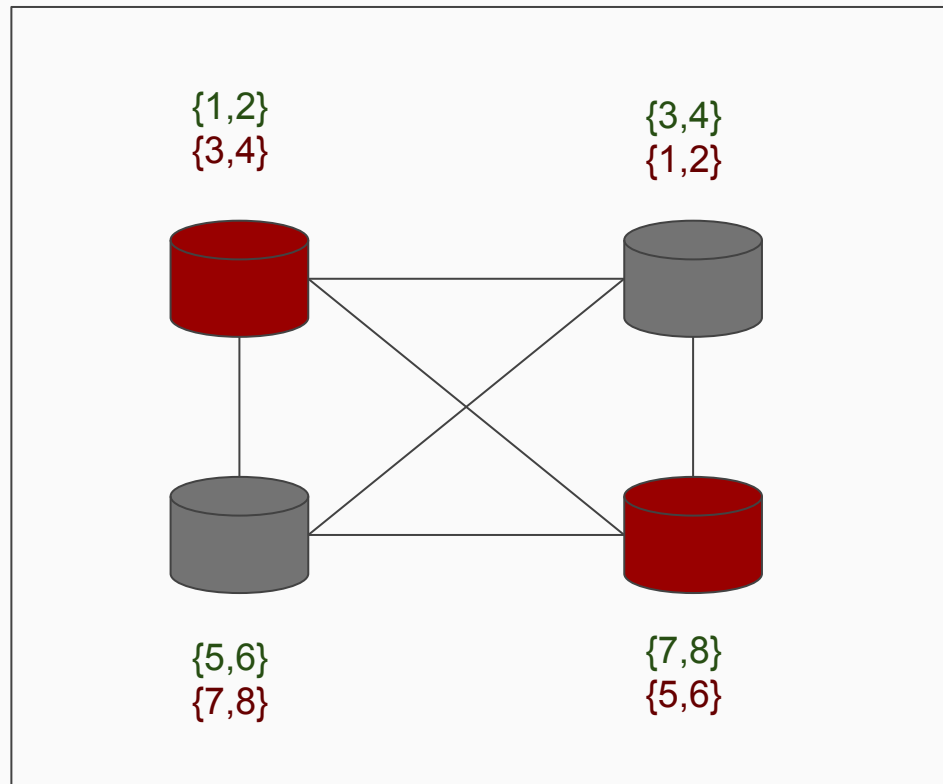
Zbiór danych: {1,2,3,4,5,6,7,8}



Wstęp do NoSql

Fault tolerance - horizontal partitioning
(partitioned cache)

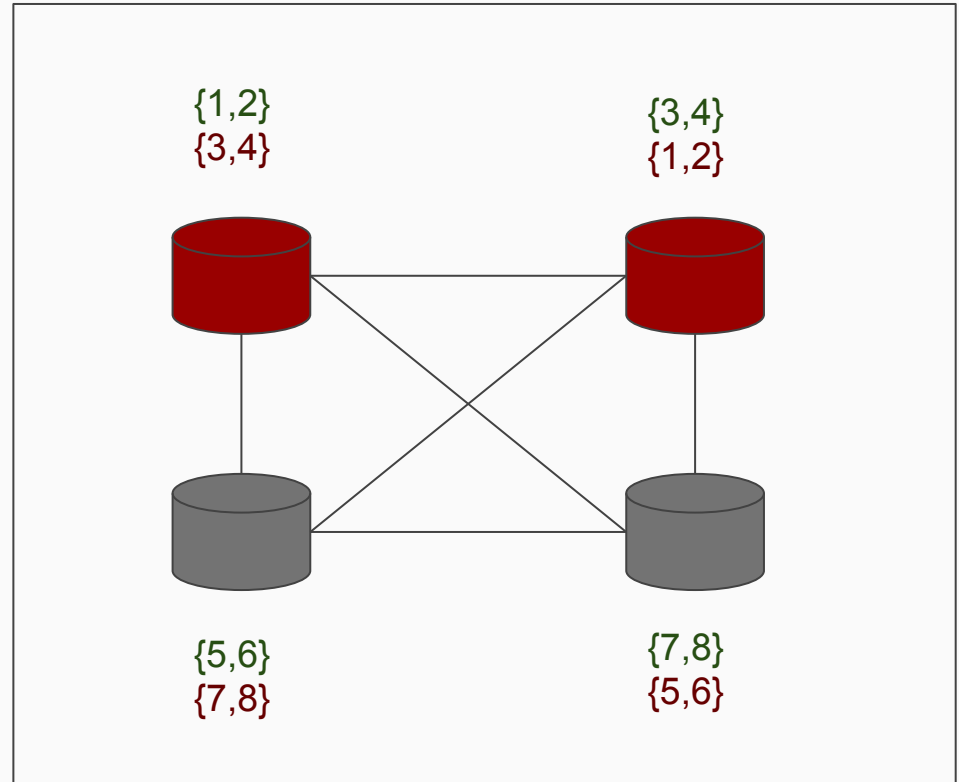
Zbiór danych: {1,2,3,4,5,6,7,8}



Zbiór danych: {1,2,3,4,5,6,7,8}

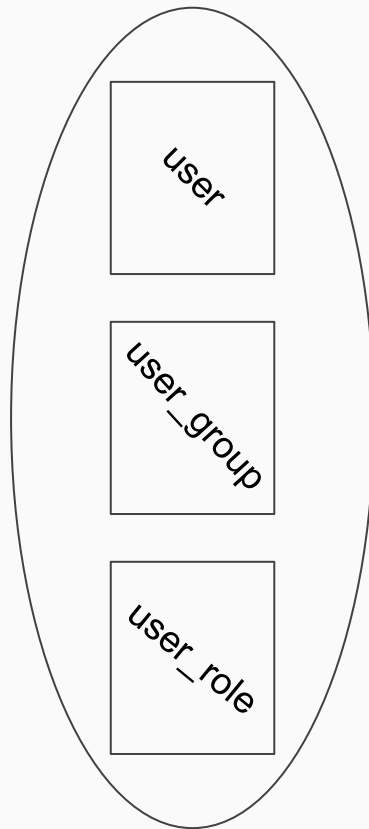
Wstęp do NoSql

Fault tolerance - horizontal partitioning
(partitioned cache)

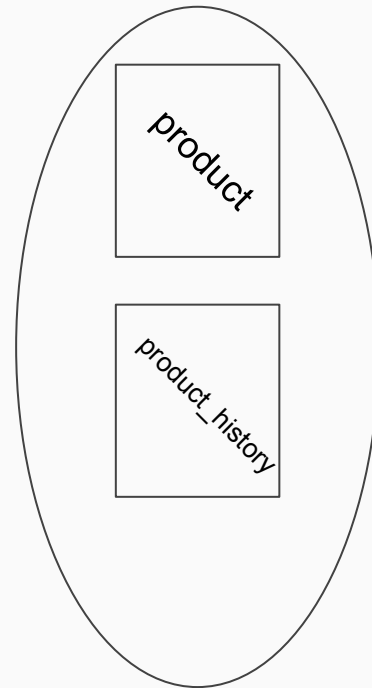


Wstęp do NoSql

Fault tolerance - vertical partitioning?



Serwer 1



Serwer 2

Wstęp do NoSql

MapReduce/EntryProcessor/Aggregator

- Przesłanie algorytmu do danych a nie danych do algorytmu,
- EntryProcessor/Aggregator - nazewnictwo z Coherence to samo co MapReduce

Wstęp do NoSql

MapReduce/EntryProcessor/Aggregator

Przykład:

Całkowity zbiór: {"a", "ab", "abc", "bb", "bac", "c", "z"}

Cel:

Obliczyć sumę wszystkich elementów z zakresu: a-b.

Serwer nr 1:

Zbiór: {"a", "ab", "abc"}

Serwer nr 2:

Zbiór: {"bb", "bac"}

Serwer nr 3:

Zbiór: {"c"}

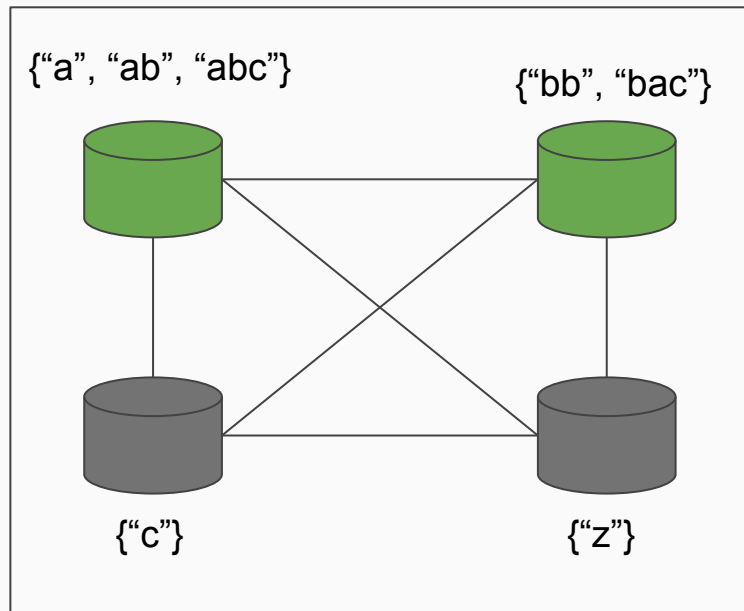
Serwer nr 4:

Zbiór: {"z"}

Wstęp do NoSql

MapReduce/EntryProcessor/Aggregator

Całkowity zbiór: {"a", "ab", "abc", "bb", "bac", "c", "z"}



Wstęp do NoSql

MapReduce/EntryProcessor/Aggregator

Operacja Map:

Server nr 1:

result_1=3

Server nr 2:

result_2=2

Operacja Reduce:

result = result_1 + result_2

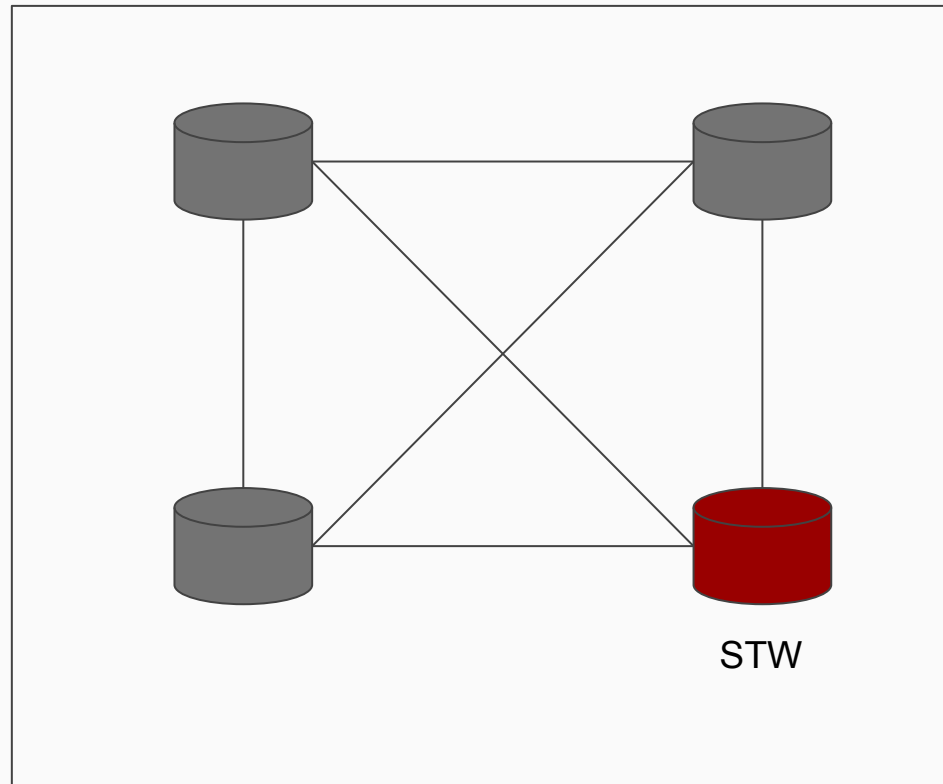
Wstęp do NoSql

Garbage Collection

- Możliwe zbudowanie systemu bez pauz GC (stop-the-world),

Wstęp do NoSql

Garbage Collection



Wstęp do NoSql

Podsumowanie

- Architektura
- Struktury danych,
- Scales out not up,
- fault tolerance,
- MapReduce/EntryProcessor/Aggregator,
- Garbage Collection.

PostgreSQL

- ACID,
- scales up
 - Rozproszone transakcje i joiny nie skalują się
- Zaawansowane indeksowanie,
- niestandardowe typy danych,
- Rozszerzenia
 - Zapytania geograficzne (wyszukiwanie po koordynatach),
- ponad 40 lat rozwoju (1970, Berkeley, “Interactive Graphics and Retrieval System”)

Redis

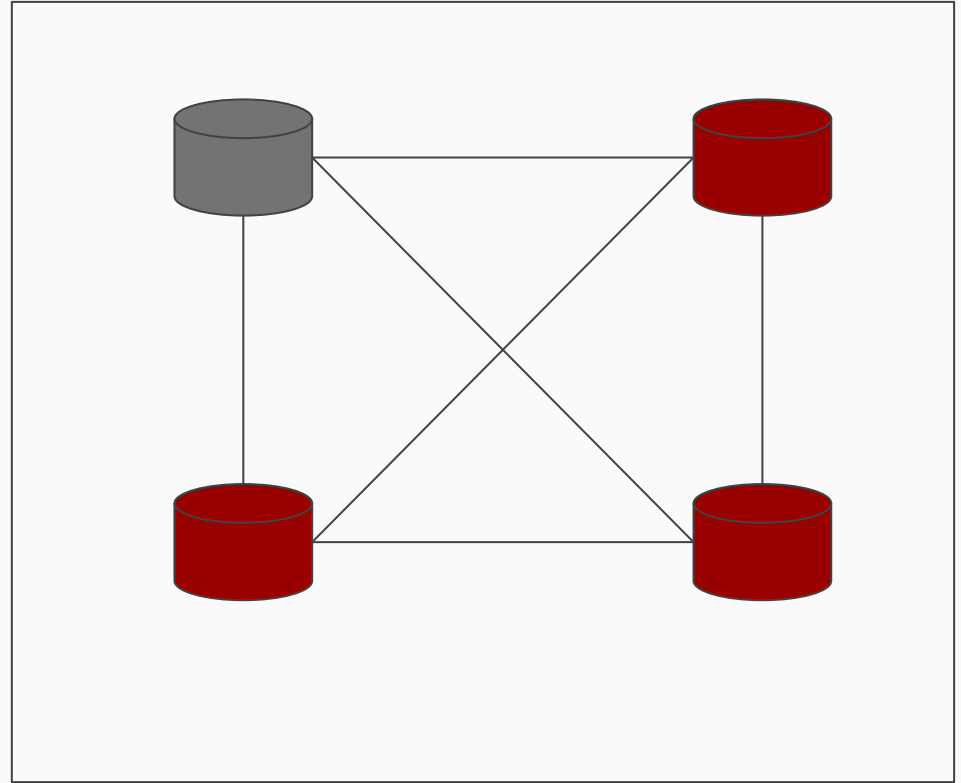
- magazyn danych (data store),
- brak MapReduce,
- key-value,
- bardzo szybki
- CAP: P, ewentualnie A, ewentualnie P
- snapshot
- Replikacja master/slave

HBase

- struktura kolumnowa,
- CAP: PC (wersjonowanie),

HBase

Availability



HBase

- atomowość na poziomie wiersza,
- brak indexowania,
- brak sortowania, prócz klucza (domyślnie)

MongoDb

- struktura - dokument,
- CAP: PC (blokowanie całego dokumentu)
- replikacja master/slave
- dobrze się sprawdza kiedy struktura danych jest nieznana i często się zmienia,
- CERN z tego korzystał, czy nadal?

Neo4j

- struktura graf
- CAP: PA
- skalowanie: replikacja master/slave
- zależność między danymi

Coherence

- HashMap,
- bardzo drogi,
- Hazelcast,
- dużo funkcjonalności
- CAP: PA (replikacja master/slave, horizontal partitioning)
- migracja obiektów

Rozwiązania hybrydowe

- Polyglot persistence
- Redis (import danych oraz cache) + MongoDB (trwałość) + Neo4j (relacja między danymi)

Sql a NoSql

Podsumowanie

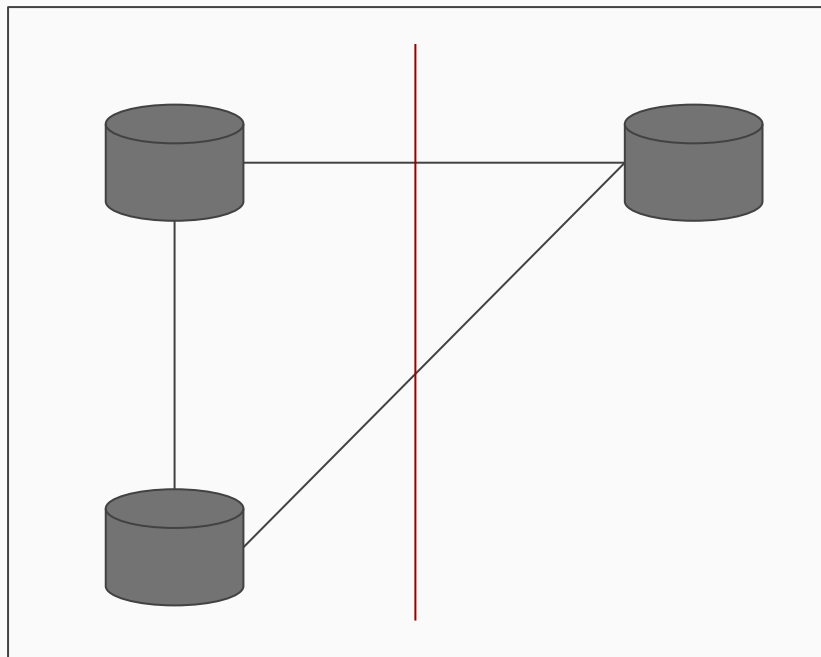
- PostgreSQL,
- Redis,
- Hbase,
- MongoDB,
- Neo4j,
- Coherence,
- Rozwiązania hybrydowe.

Na co warto zwrócić uwagę

- MapReduce/EntryProcessor, powinien się szybko wykonywać,
- Integracja z DI (Spring),
- Narzędzie do budowania projektu (Maven ,Gradle),
- Automatyczna instalacja instancji na wszystkich węzłach tworzących klaster (Ansible),
- Monitorowanie każdego węzła (JMX),
- Liczba węzłów powinna być liczbą nieparzystą (niektóre rozwiązania radzą sobie z parzystą liczbą)

Na co warto zwrócić uwagę

Nieparzysta ilość węzłów

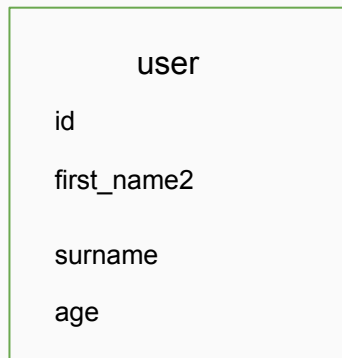
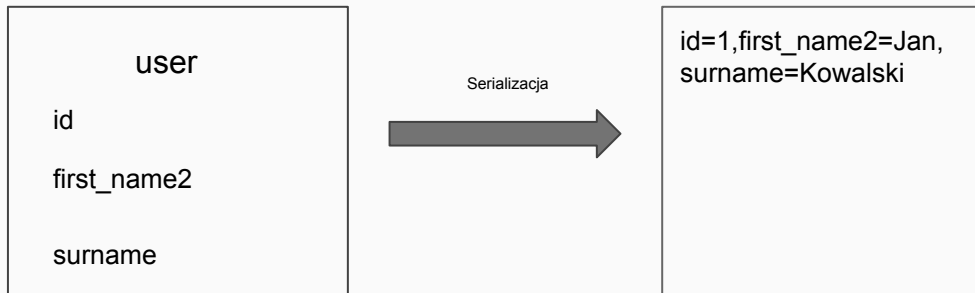


Zagrożenia

- Operacje bazodanowe w MapReduce/EntryProcessor/Aggregator,
- Transakcje wielopoziomowe (zapis do bazy + dodanie wiadomości do JMS),
- Przesył danych po sieci (między węzłami), wykonywanie algorytmu tam gdzie są dane,
- Wywołanie MapReduce/EntryProcessor w MapReduce/EntryProcessor może doprowadzić do zakleszczenia (deadlock),
- migracja obiektów.

Zagrożenia

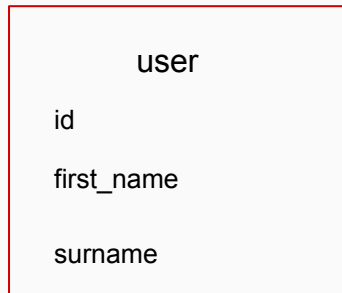
Migracja obiektów



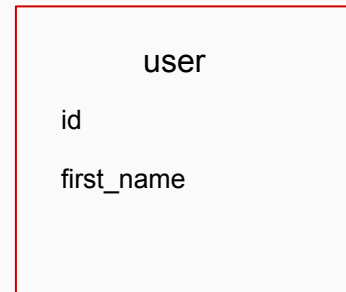
Zagrożenia

Migracja obiektów

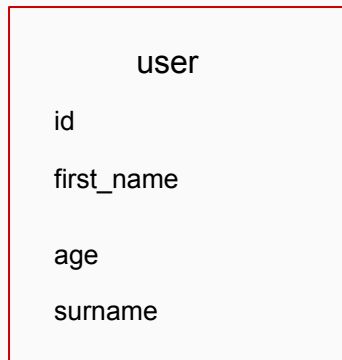
Zmiana nazwy pola



Usuwanie pola



Dodawanie pola w
środku



Testowanie

- chaos monkey
- Test integracyjne (little grid)

Co dalej?

- GridGain
- Splunk

Podsumowanie

- ACID a CAP,
- Wstęp do NoSql,
- Sql a NoSql,
- Na co warto zwrócić uwagę,
- Zagrożenia,
- Testowanie,
- Co dalej?

Książki

The
Pragmatic
Programmers

Seven Databases in Seven Weeks

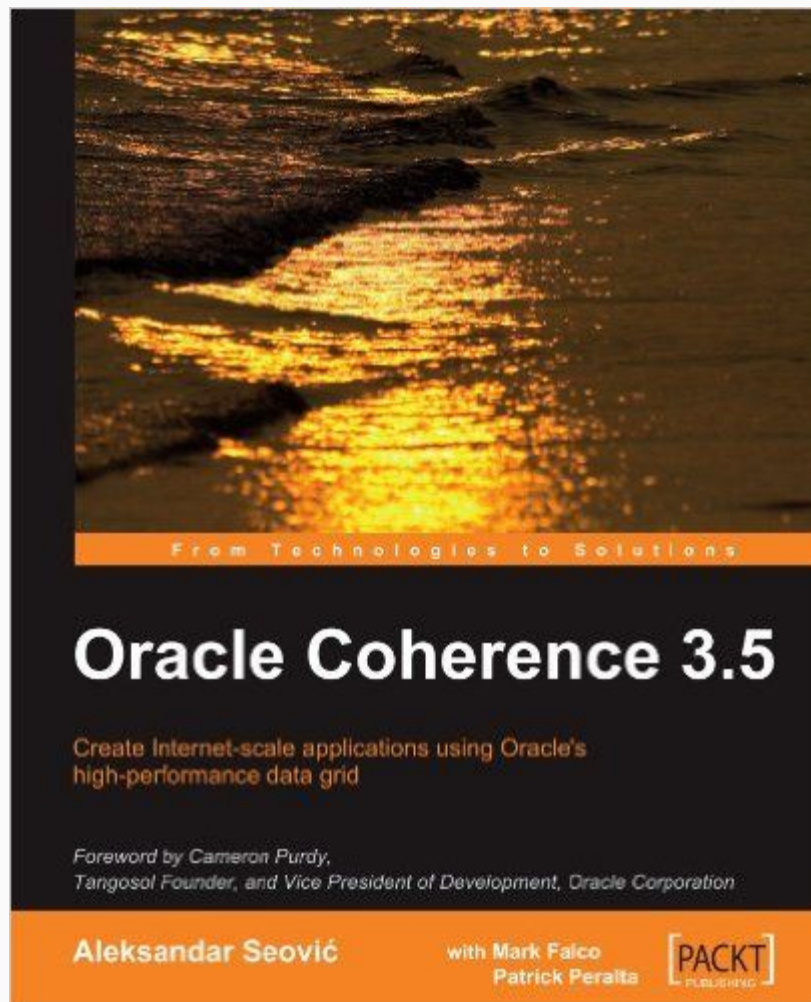
A Guide to Modern Databases
and the NoSQL Movement

Eric Redmond
and Jim R. Wilson

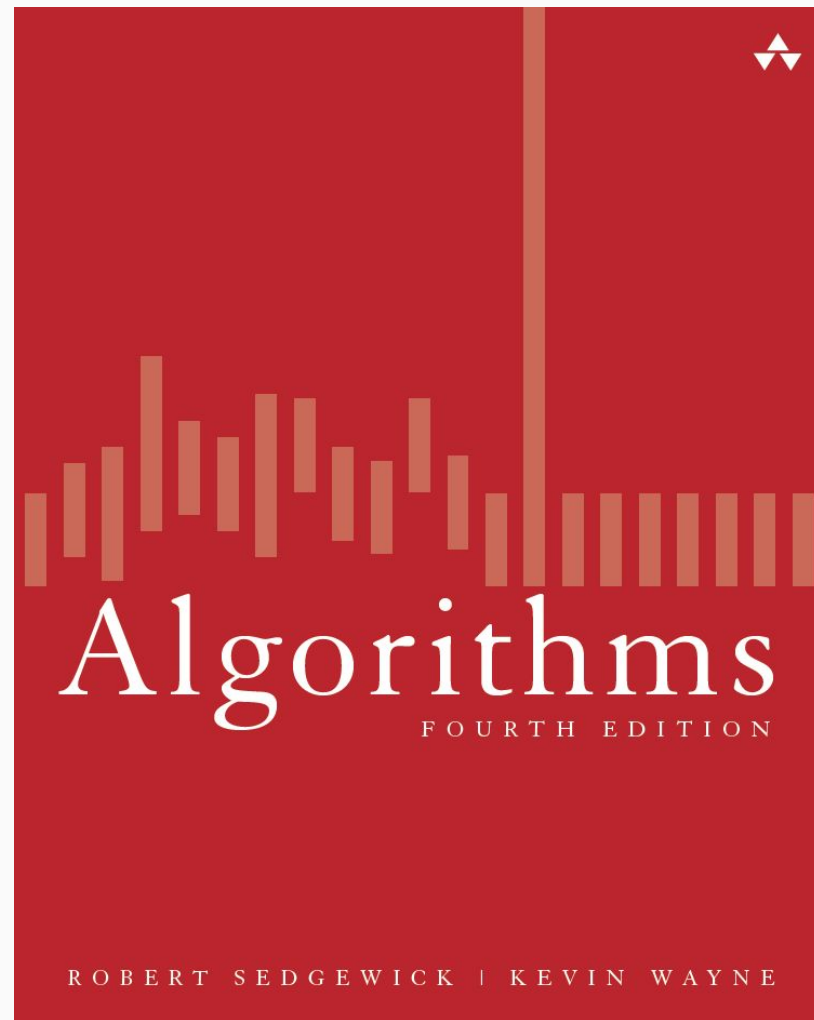
Series editor: *Bruce A. Tate*
Development editor: *Jacquelyn Carter*



Książki



Książki



Pytania i odpowiedzi

Dzięki

kontakt: maksymilian.wiesiolek@gmail.com

Uwagi, poprawki, udoskonalenia

Podziękowania dla Tomka Kleszczyńskiego, Pawła Żalejko oraz Kamila Krupy za poprawki w prezentacji