Bazy danych NoSQL
Część III

Maciej Zakrzewicz

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

Redis

- Serwer bazy danych NoSQL typu Key-Value Store
- Open source
- Przechowuje bazę danych w pamięci operacyjnej, zabezpieczając ją kopią na dysku
- Relatywnie bogaty zbiór typów danych
- Replikacja danych do dowolnej liczby serwerów
- Imponująca wydajność operacji zapisu i odczytu

Komponenty Redis

- Serwer Redis
  - redis-server
- Klient Redis
  - redis-cli

Parametry konfiguracyjne

- Plik redis.conf
- Odczyt i modyfikacja:
  - config set <parametr> <wartość>
  - config get <parametr>
- Wyświetlenie wszystkich parametrów konfiguracyjnych:
  - config get *

Typy danych

- **String** – łańcuch znakowy, maks. 512 MB
- **Hash** – kolekcja par pole-wartość, maks. 4 miliardy par
- **List** – lista łańcuchów znakowych, maks. 4 miliardy elementów
- **Set** – nieuporządkowany zbiór łańcuchów znakowych, maks. 4 miliardy elementów

Polecenia klienta Redis (String)

- ping – sprawdza połączenie z serwerem
- set – zapisanie klucza z wartością
- get – odczytanie klucza z wartością
- del – usunięcie klucza
- exists – sprawdzenie, czy klucz istnieje
- dump – pobranie serializowanej wersji wartości dla klucza
- expire, expireat, pexpire, pexpireat – ustawienie czasu wygaśnięcia klucza
- persist – wyłączenie wygasania klucza
- keys – znalezienie wszystkich kluczy pasujących do wzorca
- move – przeniesienie klucza do innej bazy danych
- rename – zmiana wartości klucza
- type – odczytanie typu wartości związanej z kluczem

Polecenia klienta Redis - przykład

```
127.0.0.1:6379> set '60-461' 'Poznan'
OK
127.0.0.1:6379> get '60-461'
"Poznan"
127.0.0.1:6379> exists '60-461'
(integer) 1
127.0.0.1:6379> exists '60-462'
(integer) 0
127.0.0.1:6379> type '60-461'
string
```

Polecenia klienta Redis (hash)

- hmset – zapisanie klucza z kolekcją par pole-wartość (hash)
- hmget – odczytanie wartości dla wskazanego pola związanego z kluczem (hash)
- hgetall – odczytanie wszystkich par pole-wartość dla podanego klucza (hash)
- hdel – usunięcie jednej lub wielu par związanych z kluczem (hash)
- hexists – sprawdzenie, czy klucz zawiera podane pole
- hkeys – pobranie wszystkich pól dla podanego klucza
- hvals – pobranie wszystkich wartości dla podanego klucza

Polecenia klienta Redis - przykład

```
127.0.0.1:6379> hmset 'Kowalski' imie 'Jan' departament 'Marketing'
OK.
127.0.0.1:6379> hgetall 'Kowalski'
1) "imie"
2) "Jan"
3) "departament"
4) "Marketing"
127.0.0.1:6379> hmget 'Kowalski' imie
1) "Jan"
127.0.0.1:6379> hkeys 'Kowalski'
1) "imie"
2) "departament"
127.0.0.1:6379> hvals 'Kowalski'
1) "Jan"
2) "Marketing"
```

Polecenia klienta Redis (list)

- lpush – dodanie elementu na początek listy
- rpush – dodanie elementu na końcu listy
- lpop – pobranie i usunięcie pierwszego elementu listy
- rpop – pobranie i usunięcie ostatniego elementu listy
- lindex – odczyt elementu z podanej pozycji listy
- lrange – odczyt zakresu elementów listy
- lrem – usunięcie elementów listy

Polecenia klienta Redis - przykład

```
127.0.0.1:6379> lpush zamowienia 'z1234'
(integer) 1
127.0.0.1:6379> lpush zamowienia 'z2345'
(integer) 2
127.0.0.1:6379> lpush zamowienia 'z3456'
(integer) 3
127.0.0.1:6379> rpop zamowienia
"z1234"
127.0.0.1:6379> rpop zamowienia
"z2345"
127.0.0.1:6379> rpop zamowienia
"z3456"
```

Polecenia klienta Redis (set)

- sadd – dodanie elementu do zbioru dla podanego klucza
- scard – pobranie liczby elementów zbioru dla podanego klucza
- sdiff – wyznaczenie różnicy zbiorów
- sinter – wyznaczenie części wspólnej zbiorów
- sunion – wyznaczenie sumy zbiorów
- smembers – pobranie wszystkich elementów zbioru dla podanego klucza
- spop – pobranie i usunięcie losowo wybranego elementu zbioru dla podanego klucza
- sismember – sprawdzenie, czy element należy do zbioru
- srem – usunięcie elementu(ów) zbioru dla podanego klucza

Polecenia klienta Redis - przykład

```
127.0.0.1:6379> sadd pracownicy_marketing 'Nowak' 'Kowalski' 'Zielinski'
(integer) 3
127.0.0.1:6379> smembers pracownicy_marketing
1) "Zielinski"
2) "Nowak"
3) "Kowalski"
127.0.0.1:6379> sismember pracownicy_marketing 'Kowalski'
(integer) 1
127.0.0.1:6379> sismember pracownicy_marketing 'Kowalska'
(integer) 0
127.0.0.1:6379> scard pracownicy_marketing
(integer) 3
127.0.0.1:6379> srem pracownicy_marketing 'Zielinski'
(integer) 1
```

Transakcje

- Gwarantują własności: atomowości i izolacji
- Rozpoczęcie transakcji:
  - `multi`
- Zakończenie transakcji:
  - `exec`
- Wycofanie transakcji:
  - `discard`

Zarządzanie pracą serwera

- Komenda info
- Kopia bezpieczeństwa:
  - save
  - uwaga: jeśli nie zostanie wykonana kopia bezpieczeństwa, to po zatrzymaniu serwera zostaną utracone wszystkie dane znajdujące się w bazie danych
- Odtwarzanie po awarii:
  - skopiowanie kopii bezpieczeństwa (dump.rdb) do katalogu roboczego
  - katalog roboczy: config get dir

Referencje

- http://redis.io/
- http://www.tutorialspoint.com/redis/index.htm