

UNIVERZITET U BEOGRADU

MATEMATIČKI FAKULTET



PROJEKAT

Poređenje Cassandra i DB2 modela

ime i prezime	Dragan Đurđević , 1059/2012 Nikola Stanković , 1033/2012 Nikola Stanojević , 1064/2012
predmet	Distribuirane i objektne baze podataka
školska godina	2013/2014
nastavnik	dr Saša Malkov
datum	10.02.2013

Sadržaj

1	Opis problema	2
2	Kreiranje modela podataka	3
2.1	Cassandra model	3
2.2	Relacioni DB2 model	6
3	Upiti	8
4	Zauzeće memorije	13
5	Moguća poboljšanja	13
6	Zaključak	15
7	Reference	16

1 Opis problema

Nekoliko tabela opisuje (redundantno) podatke o proteinima u određenoj grupi. Na primer, tabele:

- DOBP.SITE_REGLEN_KINGDOM
- DOBP.SITE_REGLEN_PHYLUM
- DOBP.SITE_REGLEN_ORGANISM

opisuju koliko ima ukupno uređenih i neuređenih regiona određene dužine u određenoj grupi proteina (svi proteini u kraljevstvu, filumu ili jednom organizmu). U skladu sa time i imaju određene atribute. Prva tabela ima atribute: SUPERKINGDOM, RLEN, ORDCOUNT, DISORDCOUNT dok druga na to dodaje jos i PHYLUM, a treća dodaje i ORGANISM. Imaju redom po oko 4k, 25k i 300k redova.

Zadatak:

Napraviti Cassandra model podataka tabele:

SITE_PROTLEN_*

- SITE_PROTLEN_KINGDOM
- SITE_PROTLEN_PHYLUM
- SITE_PROTLEN_ORGANISM

SITE_REGLEN_*

- SITE_REGLEN_KINGDOM
- SITE_REGLEN_PHYLUM
- SITE_REGLEN_ORGANISM

SITE_REGPROC_*

- SITE_REGPROC_KINGDOM
- SITE_REGPROC_PHYLUM
- SITE_REGPROC_ORGANISM

tako da se mogu pretraživati po više kriterijuma. Napuniti test primerom podataka. Uporediti zauzeće prostora i efikasnost u odnosu na relacionu bazu DB2.

2 Kreiranje modela podataka

2.1 Cassandra model

Prvi korak u kreiranju Cassandra modela je kreiranje prostora ključeva i potrebnih tabela:

```
CREATE KEYSPACE dobp
WITH strategy_class = 'SimpleStrategy'
AND strategy_options:replication_factor = 1;

USE dobp;

CREATE TABLE SITE_PROTLEN_KINGDOM
(
SUPERKINGDOM TEXT ,
PLEN INT ,
PROTCOUNT INT ,
PRIMARY KEY (SUPERKINGDOM,PLEN)
);

CREATE TABLE SITE_PROTLEN_PHYLUM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
PLEN INT ,
PROTCOUNT INT ,
PRIMARY KEY (PHYLUM,PLEN)
);

CREATE TABLE SITE_PROTLEN_ORGANISM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
ORGANISM TEXT ,
PLEN INT ,
PROTCOUNT INT ,
PRIMARY KEY (ORGANISM,PLEN)
);

CREATE TABLE SITE_REGLEN_KINGDOM
(
SUPERKINGDOM TEXT ,
RLEN INT ,
ORDCOUNT INT ,
DISORDCOUNT INT ,
PRIMARY KEY (SUPERKINGDOM,RLEN)
);
```

```

CREATE TABLE SITE_REGLEN_PHYLUM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
RLEN INT ,
ORDCOUNT INT ,
DISORDCOUNT INT ,
PRIMARY KEY (PHYLUM ,RLEN)
);

CREATE TABLE SITE_REGLEN_ORGANISM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
ORGANISM TEXT ,
RLEN INT ,
ORDCOUNT INT ,
DISORDCOUNT INT ,
PRIMARY KEY (ORGANISM ,RLEN)
);

CREATE TABLE SITE_REGPROC_KINGDOM
(
SUPERKINGDOM TEXT ,
RLEN INT ,
ORDPROC DECIMAL ,
DISORDPROC DECIMAL ,
PRIMARY KEY (SUPERKINGDOM ,RLEN)
);

CREATE TABLE SITE_REGPROC_PHYLUM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
RLEN INT ,
ORDPROC DECIMAL ,
DISORDPROC DECIMAL ,
PRIMARY KEY (PHYLUM ,RLEN)
);

CREATE TABLE SITE_REGPROC_ORGANISM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
ORGANISM TEXT ,
RLEN INT ,
ORDPROC DECIMAL ,
DISORDPROC DECIMAL ,
PRIMARY KEY (ORGANISM ,RLEN)
);

```

Podaci se u tabelu ubacuju iz csv datoteke pomocu komande

```
COPY naziv_tabele (lista_atributa)
FROM 'lokacija_csv_datoteke';
```

na primer

```
COPY site_protlen_kingdom(superkingdom, plen, protcount)
FROM './site_protlen_kingdom.csv';
```

Da bi se omogućilo pretraživanje po atributima koji nisu ključni atributi, potrebno je napraviti indekse po datim atributima:

```
CREATE INDEX index_site_protlen_kingdom_pl
ON site_protlen_kingdom (plen);

CREATE INDEX index_site_protlen_kingdom_pc
ON site_protlen_kingdom (protcount);

CREATE INDEX index_site_protlen_phylum_pl
ON site_protlen_phylum (plen);

CREATE INDEX index_site_protlen_phylum_pc
ON site_protlen_phylum (protcount);

CREATE INDEX index_site_protlen_phylum_sk
ON site_protlen_phylum (superkingdom);

CREATE INDEX index_site_protlen_organism_pl
ON site_protlen_organism (plen);

CREATE INDEX index_site_protlen_organism_pc
ON site_protlen_organism (protcount);

CREATE INDEX index_site_protlen_organism_sk
ON site_protlen_organism (superkingdom);

CREATE INDEX index_site_protlen_organism_ph
ON site_protlen_organism (phylum);
```

2.2 Relazioni DB2 model

```
CREATE TABLE SITE_PROTLEN_KINGDOM
(
SUPERKINGDOM VARCHAR(10) NOT NULL,
PLEN INTEGER NOT NULL,
PROTCOUNT INTEGER NOT NULL,
PRIMARY KEY (SUPERKINGDOM,PLEN)
);

CREATE TABLE SITE_PROTLEN_PHYLUM
(
SUPERKINGDOM VARCHAR(10) NOT NULL,
PHYLUM VARCHAR(30) NOT NULL,
PLEN INTEGER NOT NULL,
PROTCOUNT INTEGER NOT NULL,
PRIMARY KEY (PHYLUM,PLEN)
);

CREATE TABLE SITE_PROTLEN_ORGANISM
(
SUPERKINGDOM VARCHAR(10) NOT NULL,
PHYLUM VARCHAR(30) NOT NULL,
ORGANISM VARCHAR(20) NOT NULL,
PLEN INTEGER NOT NULL,
PROTCOUNT INTEGER NOT NULL,
PRIMARY KEY (ORGANISM,PLEN)
);

CREATE TABLE SITE_REGLEN_KINGDOM
(
SUPERKINGDOM VARCHAR(10) NOT NULL,
RLEN INTEGER NOT NULL,
ORDCOUNT INTEGER NOT NULL,
DISORDCOUNT INTEGER NOT NULL,
PRIMARY KEY (SUPERKINGDOM,RLEN)
);

CREATE TABLE SITE_REGLEN_PHYLUM
(
SUPERKINGDOM VARCHAR(10) NOT NULL,
PHYLUM VARCHAR(30) NOT NULL,
RLEN INTEGER NOT NULL,
ORDCOUNT INTEGER NOT NULL,
DISORDCOUNT INTEGER NOT NULL,
PRIMARY KEY (PHYLUM,RLEN)
);
```

```

CREATE TABLE SITE_REGLEN_ORGANISM
(
SUPERKINGDOM VARCHAR(10) NOT NULL ,
PHYLUM VARCHAR(30) NOT NULL ,
ORGANISM VARCHAR(20) NOT NULL ,
RLEN INTEGER NOT NULL ,
ORDCOUNT INTEGER NOT NULL ,
DISORDCOUNT INTEGER NOT NULL ,
PRIMARY KEY (ORGANISM,RLEN)
);

CREATE TABLE SITE_REGPROC_KINGDOM
(
SUPERKINGDOM VARCHAR(10) NOT NULL ,
RLEN INTEGER NOT NULL ,
ORDPROC DECIMAL(31, 17) NOT NULL ,
DISORDPROC DECIMAL(31, 17) NOT NULL ,
PRIMARY KEY (SUPERKINGDOM,RLEN)
);

CREATE TABLE SITE_REGPROC_PHYLUM
(
SUPERKINGDOM VARCHAR(10) NOT NULL ,
PHYLUM VARCHAR(30) NOT NULL ,
RLEN INTEGER NOT NULL ,
ORDPROC DECIMAL(31, 17) NOT NULL ,
DISORDPROC DECIMAL(31, 17) NOT NULL ,
PRIMARY KEY (PHYLUM,RLEN)
);

CREATE TABLE SITE_REGPROC_ORGANISM
(
SUPERKINGDOM VARCHAR(10) NOT NULL ,
PHYLUM VARCHAR(30) NOT NULL ,
ORGANISM VARCHAR(20) NOT NULL ,
RLEN INTEGER NOT NULL ,
ORDPROC DECIMAL(31, 17) NOT NULL ,
DISORDPROC DECIMAL(31, 17) NOT NULL ,
PRIMARY KEY (ORGANISM,RLEN)
);

```

Koristeći IBM Data Studio opcijom *load* smo učitali podatke iz csv datoteka u tabele.

3 Upiti

U nastavku je dat spisak upita na kojima je testiran model. Prvi upit je SQL upit u IBM DB2 dok je drugi upit njegov CQL ekvivalent u Cassandra-i.

Upit 1:

```
SELECT * FROM site_protlen_kingdom
        WHERE superkingdom = 'Archaea';

SELECT * FROM dobp.site_protlen_kingdom
        WHERE superkingdom = 'Archaea';
```

Upit 2:

```
SELECT * FROM site_protlen_kingdom
        WHERE protcount = 0;

SELECT * FROM dobp.site_protlen_kingdom
        WHERE protcount = 0;
```

Upit 3:

```
SELECT * FROM site_protlen_phylum
        WHERE phylum = 'Firmicutes';

SELECT * FROM dobp.site_protlen_phylum
        WHERE phylum = 'Firmicutes';
```

Upit 4:

```
SELECT * FROM site_protlen_phylum
        WHERE plen = 128;

SELECT * FROM dobp.site_protlen_phylum
        WHERE plen = 128;
```

Upit 5:

```
SELECT * FROM site_protlen_phylum
        WHERE superkingdom = 'Bacteria'
        AND protcount > 28
        FETCH FIRST 500 ROWS ONLY;

SELECT * FROM dobp.site_protlen_phylum
        WHERE superkingdom = 'Bacteria'
        AND protcount > 28 LIMIT 500
        ALLOW FILTERING;
```

Upit 6:

```
SELECT * FROM site_protlen_organism
        WHERE organism = 'NC_007722.1';

SELECT * FROM dobp.site_protlen_organism
        WHERE organism = 'NC_007722.1';
```

Upit 7:

```
SELECT * FROM site_protlen_organism
        WHERE protcount = 4
        FETCH FIRST 1000 ROWS ONLY;

SELECT * FROM dobp.site_protlen_organism
        WHERE protcount = 4
        LIMIT 1000;
```

Upit 8:

```
SELECT * FROM site_protlen_organism
        WHERE plen IN (96, 256, 320, 384, 512, 768);

SELECT * FROM dobp.site_protlen_organism
        WHERE plen IN (96, 256, 320, 384, 512, 768)
        ALLOW FILTERING;
```

Upit 9:

```
SELECT count(*) FROM site_protlen_organism;

SELECT count(*) FROM dobp.site_protlen_organism;
```

Upit 10:

```
SELECT * FROM site_reglen_phylum
        WHERE superkingdom = 'Bacteria'
        AND ordcount > 8192;

SELECT * FROM dobp.site_reglen_phylum
        WHERE superkingdom = 'Bacteria '
        AND ordcount > 8192
        LIMIT 15000
        ALLOW FILTERING;
```

Upit 11:

```
SELECT * FROM site_reglen_phylum
        WHERE superkingdom = 'Bacteria'
        AND ordcount > 16384
        AND disordcount < 80000;

SELECT * FROM dobp.site_reglen_phylum
        WHERE superkingdom = 'Bacteria'
        AND ordcount > 16384
        AND disordcount < 80000
        LIMIT 10000
        ALLOW FILTERING;
```

Upit 12:

```
SELECT * FROM site_reglen_phylum
        WHERE superkingdom = 'Archaea'
        AND ordcount > 1024
        AND ordcount < 8192;

SELECT * FROM dobp.site_reglen_phylum
        WHERE superkingdom = 'Archaea'
        AND ordcount > 1024
        AND ordcount < 8192
        LIMIT 100000
        ALLOW FILTERING;
```

Upit 13:

```
SELECT * FROM site_reglen_kingdom
        WHERE superkingdom = 'Archaea'
        AND rlen IN (52,14,11,54,31);

SELECT * FROM dobp.site_reglen_kingdom
        WHERE superkingdom = 'Archaea'
        AND rlen IN (52,14,11,54,31);
```

Upit 14:

```
SELECT count(*) FROM site_reglen_kingdom
        WHERE superkingdom='Bacteria'
        AND rlen>16
        AND rlen<45;

SELECT count(*) FROM dobp.site_reglen_kingdom
        WHERE superkingdom='Bacteria'
        AND rlen>16
        AND rlen<45;
```

Upit 15:

```
SELECT DISTINCT phylum FROM site_reglen_phylum
                WHERE superkingdom = 'Bacteria';

SELECT DISTINCT phylum FROM dobp.site_reglen_phylum
                WHERE superkingdom = 'Bacteria';
```

Upit 16:

```
SELECT count(*) FROM site_reglen_organism
                WHERE rlen IN (16, 33, 55, 60);

SELECT count(*) FROM dobp.site_reglen_organism
                WHERE rlen IN (16, 33, 55, 60)
                LIMIT 2000
                ALLOW FILTERING;
```

Upit 17:

```
SELECT * FROM site_regproc_kingdom
                WHERE superkingdom = 'Bacteria'
                ORDER BY rlen DESC;

SELECT * FROM dobp.site_regproc_kingdom
                WHERE superkingdom = 'Bacteria '
                ORDER BY rlen DESC;
```

Upit 18:

```
SELECT phylum, rlen, ordproc, disordproc
       FROM site_regproc_phylum
       WHERE phylum IN('Crenarchaeota', 'Cyanobacteria')
       ORDER BY rlen;

SELECT phylum, rlen, ordproc, disordproc
       FROM dobp.site_regproc_phylum
       WHERE phylum IN('Crenarchaeota', 'Cyanobacteria')
       ORDER BY rlen;
```

Upit 19:

```
SELECT * FROM site_regproc_organism
                WHERE superkingdom='Bacteria'
                AND ordproc<2 AND ordproc>1;

SELECT * FROM dobp.site_regproc_organism
                WHERE superkingdom='Bacteria '
                AND ordproc<2 AND ordproc>1
                LIMIT 1500
                ALLOW FILTERING;
```

Upit 20:

```
SELECT * FROM site_regproc_organism
WHERE superkingdom='Bacteria'
AND ordproc<2.048
AND ordproc>1.512;

SELECT * FROM dobp.site_regproc_organism
WHERE superkingdom='Bacteria '
AND ordproc<2.048
AND ordproc>1.512
LIMIT 1500
ALLOW FILTERING;
```

Vremena izvršavanja datih upita prikazani su u tabeli 1.1 i data su u milisekundama.

	IBM DB2	Cassandra
upit 1	265	52
upit 2	25	36
upit 3	82	57
upit 4	62	24
upit 5	6	6477
upit 6	81	24
upit 7	48	5257
upit 8	236	780
upit 9	521	7165
upit 10	60	559
upit 11	5	598
upit 12	5	96
upit 13	20	4
upit 14	3	8
upit 15	4	4
upit 16	63	115
upit 17	57	7
upit 18	32	8
upit 19	21	7850
upit 20	24	6367

Tabela 1.1: Vremena izvršavanja upita.

4 Zauzeće memorije

DB2:

Za dobijanje informacije o zauzeću memorije DB2 baze podataka koristili smo poziv procedure:

```
|| CALL GET_DBSIZE_INFO(?,?,?,-1);
```

Cassandra:

Za dobijanje informacije o zauzeću memorije Cassandra baze podataka koristili smo program za ispis statistike:

```
|| nodetool -host localhost cfstats
```

Dobijeno je da je veličina db2 baze podataka $\approx 160\text{MB}$, dok je veličina Cassandra baze podataka $\approx 37\text{MB}$

5 Moguća poboljšanja

Kako je navedeno da je česta pretraga za tabele *site_protlen_**, *site_reglen_** i *site_regproc_** po fiksnom uslovu za atribute *superkingdom*, *phylum* i *organism* moguće je u nekim tabelama dodati ove atribute u ključ za partitionisanje. Tako da u tabelama **_phylum* u ključ za partitionisanje dodajemo i atribut *superkingdom* dok u tabelama **_organism* u ključu za partitionisanje dodajemo i atribut *phylum* i atribut *superkingdom*. Novokreirana tabela *site_regproc_phylum* ima sledeću formu:

```
CREATE TABLE SITE_REGPROC_PHYLUM
(
  SUPERKINGDOM TEXT ,
  PHYLUM TEXT ,
  RLEN INT ,
  ORDPROC DECIMAL ,
  DISORDPROC DECIMAL ,
  PRIMARY KEY ((SUPERKINGDOM , PHYLUM) , RLEN)
);
```

dok tabela *site_regproc_organism* izgleda

```

CREATE TABLE SITE_REGPROC_ORGANISM
(
SUPERKINGDOM TEXT ,
PHYLUM TEXT ,
ORGANISM TEXT ,
RLEN INT ,
ORDPROC DECIMAL ,
DISORDPROC DECIMAL ,
PRIMARY KEY ((SUPERKINGDOM , PHYLUM , ORGANISM) , RLEN)
);

```

Nakon ovog eksperimenta možemo uočiti nekoliko pravila. Prvo, i glavno, u slučaju da u novonapravljenim tabelama vršimo restrikciju samo po atributu *superkingdom* dobijaju se bolji rezultati. Vreme izvršavanja se smanjuje za 10% do 20%. Naravno, ovo pravilo važi i ako vršimo restrikciju i po atributu *phylum* u tabelama **_organism*. U slučaju složenijih uslova prilikom restrikcije, kao na primer u upitima 19 i 20 iz tabele 1.1, iznenađujuće dobijaju se dosta lošija rešenja nego prilikom merenja u početnom modelu.

6 Zaključak

Cassandra se pokazala kao bolje rešenje kada se vrši restrikcija po vrednostima ključa za partitionisanje. To posebno dolazi do izražaja za skupove podataka sa relativno malim brojem slogova. Ipak, kada je potrebno obaviti složeniju restrikciju tu se DB2 pokazao znatno boljim. Posebno treba naglasiti da pri postavljanju restrikcije nad atributima tipa *decimal* performanse značajno slabe u odnosu na DB2 (što se najbolje vidi pri rezultatima izvršavanja upita broj 19 i 20 u tabeli 1.1).

Kako je korišćen CQL za postavljanje upita uočeno je da je taj jezik znatno siromašniji od standardnog SQL-a. Tako na primer, CQL ne podržava spajanja, podupite i agregatne funkcije, izuzev *count*, koja ima izuzetno ograničene mogućnosti. Dakle, moguće je postavljanje izuzetno jednostavnih upita dok DB2 sa svojom implementacijom SQL-a pruža mnogo veće mogućnosti za obradu podataka.

Konačno, može se zaključiti da je korišćenje DB2 na priloženom skupu podataka, koristeći opisane modele, za realizovanje prikazanih upita znatno povoljniji.

7 Reference

1. DataStax Cassandra Documentation.

<http://www.datastax.com/docs>

2. DB2 Information Center.

<http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp>