

cassandra

Apache Cassandra

BACKUP/RESTORE BAZA PODATAKA

Đorđe Nikolić | Sistemi za upravljanje bazama podataka | maj 2022.

Sadržaj

Sadržaj	1
Uvod	2
Apache Cassandra	2
Arhitektura	2
Organizacija podataka	4
Backup i restore baza podataka.....	5
Apache Cassandra	5
Metode čuvanja stanja.....	5
Organizacija direktorijuma podataka.....	6
Priprema baze podataka za demonstriranje postojećih mehanizma	6
Slike baze podataka.....	7
Inkrementalno čuvanje	11
Povratak stanja baze podataka	13
Citirana dela	15

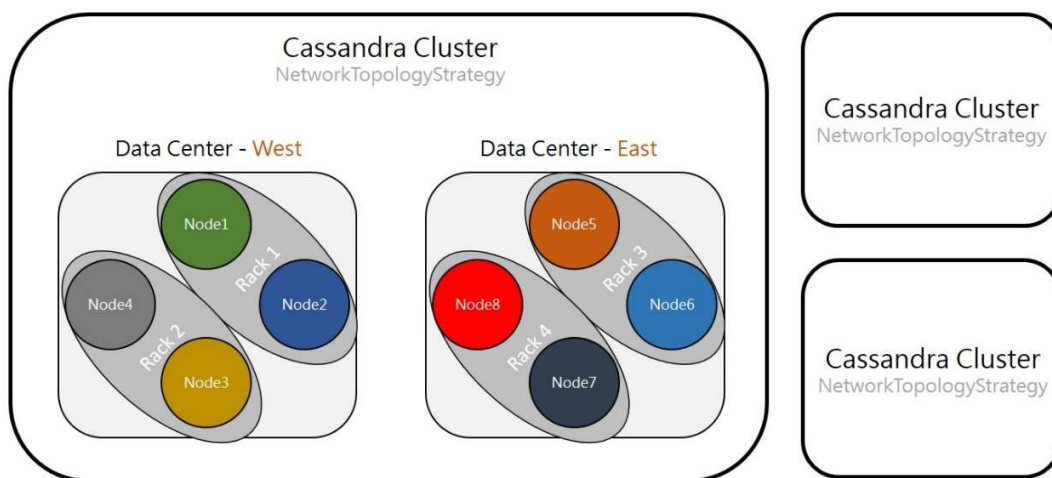
Uvod

U ovom radu će biti analiziran i predstavljen proces pravljenja rezervnih kopija baza podataka, kao i ponovno učitavanje tih kopija, u sistemu *Apache Cassandra*. U okviru prvog dela rada će biti predstavljen sam sistem, sa obzirom na to da nije u pitanju klasičan relacioni sistem baza podataka. U drugom delu rada, biće predstavljen proces kreiranja rezervnih kopija u ovom sistemu, kao i različite metode datog procesa. Za svaku od datih metoda će biti demonstrirani i primeri u trenutno najnovijoj verziji sistema (*Apache Cassandra 4.0.4*), uz pomoć evaluacione verzije *Hackolade* softvera, i CQLSH terminala.

Apache Cassandra

Apache Cassandra je NoSQL (nerelaciona) distribuirana baza podataka koda otvorenog tipa čiji je glavni fokus na odličnoj skalabilnosti, otpornosti na greške, i dostupnosti. Bazirana je na Amazon Dynamo, i Google Bigtable sistemima, i originalno je razvijana od strane Facebook-a, iako je od jula 2008. godine celokupni kod sistema potpuno dostupan javnosti.

ARHITEKTURA



Slika 1. Arhitektura sistema Apache Cassandra na visokom nivou [1]

Cassandra je organizovana po distribuiranim klasterima koji sadrže više homogenih čvorova (*nodes*), i po njima su podaci raspoređeni i organizovani tako da nikad ne postoji takozvana „jedna tačka propasti“ [2]. Čvorovi su u stalnoj međusobnoj komunikaciji i razmenjuju informacije o trenutnom stanju koristeći *peer-to-peer* komunikacioni protokol. Na svakom čvoru se održava sekvencijalni *commit log* koji beleži svaki upis podataka, koji se nakon toga indeksiraju i upisuju u strukturu u memoriji,

nazvanu *memtable* (jedna za svaku porodicu kolona). Svaki put kada se ova struktura popuni, podaci se upisuju na disk u formi *SSTable* fajlova. Svaki proces upisa u bazu je praćen automatskim particionisanjem i replikacijom na čvorovima u klasteru. Sistem periodično konsolidira podatke u *SSTable* fajlovima kroz specijalan proces kompakcije (engl. compaction) putem kog takođe briše zastarele podatke označene za brisanjem *tombstone* objektom. Kako bi se osiguralo stanje konzistentnosti svih podataka, sistem koristi različite *repair* mehanizme.

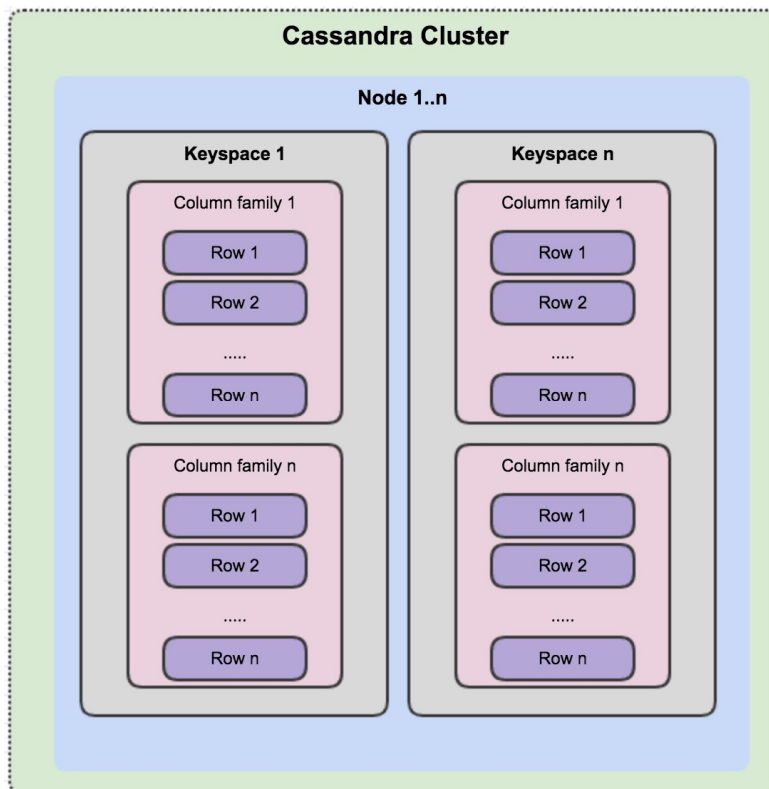
Cassandra je particionisana baza podataka koji svoje podatke čuva u redovima. Redovi su organizovani u tabele (porodice kolona) sa obaveznim primarnim ključem. Pristup bazi se vrši kroz CQL jezik, koji koristi sličnu sintaksu kao i SQL, jezik popularan među relacionim bazama podataka. Jedna od najvećih razlika između ova dva jezika, kao i odlika Apache Cassandra sistema koji ga najviše razlikuje od relacionih sistema baza podataka, je to što koncept spajanja tabela ne postoji. Takođe, nemoguće je pisati podupite. Podaci se čuvaju u denormalizovanom formatu, i prilikom dizajniranja baze podataka za neku aplikaciju, mora se pristupiti drugačije nego prilikom dizajniranja relacione baze podataka za istu namenu.

Zahtevi za čitanje ili upis podataka mogu biti poslani na bilo koji čvor u klasteru. Kada se klijent poveže na neki od čvorova, taj čvor postaje koordinator za interakciju sa tim klijentom. Preko koordinatora se vrši komunikacija između klijenta i čvorova koji zapravo sadrže potraživane podatke. On takođe određuje, na osnovu konfiguracije klastera, kom čvoru treba da prosledi zahtev na procesiranje.

Pregled najbitnijih struktura u sistemu:

- **Commit Log (dnevnik upisa):** Datoteka u kojoj se čuvaju zapisi o svim upisima u bazu podataka.
- **Memtable:** Struktura u memoriji koja čuva upisane redove pre nego što se sačuvaju na disk. Za jednu porodicu kolona, uglavnom postoji jedna ova struktura.
- **SSTable:** Datoteka u kojoj se čuva set nepromenljivih redova podataka, sortiranih po primarnom ključu. Kada se struktura *memtable* čuva na disk, kreira se novi *SSTable*.

ORGANIZACIJA PODATAKA



Slika 2. Organizacija podataka na nivou klastera u Apache Cassandra sistemu. [3]

Organizacija podataka na nivou klastera u ovom sistemu se vrši pomoću sledećih objekata [4]:

- **Keyspace:** Definiše kako se podaci repliciraju u jednoj grupaciji klastera (*datacenter*). Replikacija je broj kopija koje se čuvaju po klasteru. Keyspace sadrži tabele.
- **Tabela:** Definiše šemu za čuvanje kolekcije particija. Tabele se sastoje iz particija, koje sadrže redove podataka, koji sadrže kolone. Jedna od odlika tabele u ovom sistemu je to što se nove kolone mogu dodati vrlo lako.
- **Particija:** Definiše obavezni deo primarnog ključa koji svi redovi u sistemu poseduju kako bi se identifikovao čvor u kome se čuva dati red. Svi brzi upiti moraju da sadrže particioni ključ.
- **Red:** Sadrži kolekciju kolona identifikovanu putem unikatnog primarnog ključa koji se sastoji iz obaveznog particionog ključa i opcionih ključeva klasterovanja (određuju po kojim kolonama se podaci sortiraju i grupišu).

Cassandra omogućava performantno izvršenje upita na nivou pojedinačnih particija, dok su operacije nad više particija uglavnom nepodržane.

Backup i restore baza podataka

Backup baze podataka podrazumeva čuvanje operacionog stanja, arhitekture i podataka određene baze podataka. Omogućava kreiranje kopije baze, kako bi se stanje očuvalo u slučaju da primarna baza doživi neki prekid, ili bude izgubljena. Sačuvano stanje baze podataka takođe omogućava portabilnost, tj. transfer podataka na neku drugu lokaciju.

Sam proces čuvanja trenutnog stanja baze podataka, kao i eventualni oporavak iz tog sačuvanog stanja, zavisi od samog sistema za upravljanje bazama podataka.

APACHE CASSANDRA

Kao i svaki drugi sistem za upravljanje bazama podataka, Apache Cassandra ima svoj način čuvanja stanja baza podataka kao i njihov oporavak. Kao što je pomenuto ranije, podaci se na disku čuvaju u formi SSTable datoteka. U ovom sistemu, čuvanje stanja se izričito vrši kroz čuvanje kopija postojećih SSTable datoteka. Sa obzirom na to, svi podaci koji se nalaze u radnoj memoriji računara (u memtable strukturama) prilikom pokretanja procesa čuvanja, se prvo snimaju na disk u obliku SSTable datoteka, pre nastavka procesa čuvanja. [5]

Metode čuvanja stanja

Sistem podržava dve strategije čuvanja stanja baze:

- Slika (engl. snapshot)
- Inkrementalno čuvanje (engl. incremental backups)

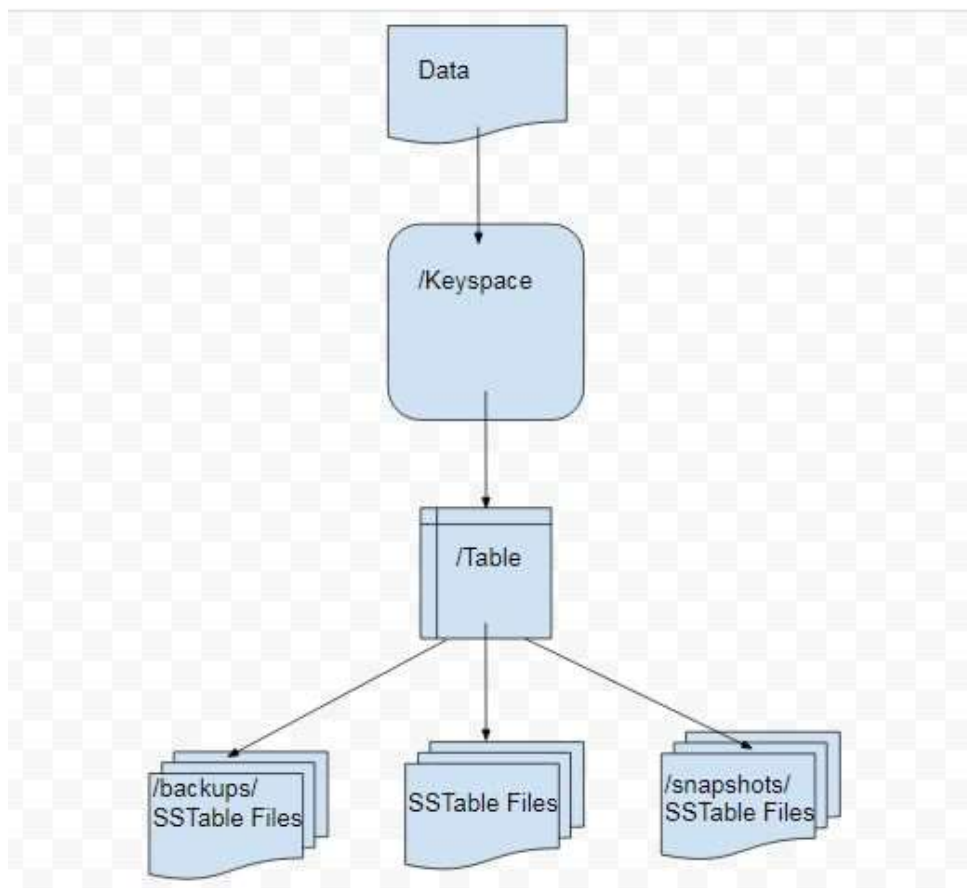
Slika predstavlja kopiju SSTable datoteka neke tabele, u određenom trenutku. Takođe se čuva i definicija samih tabela koja se može iskoristiti za njihovo ponovno kreiranje. Slike baze podataka mogu kreirati korisnici, ili kroz automatske procese. Kroz određena podešavanja u *cassandra.yaml* datoteci je moguće definisati da li se slike kreiraju pre svakog procesa kompakcije. Takođe je moguće podesiti da li se slike kreiraju automatski pre svakog procesa brisanja tabela ili celog keyspace-a. Ovo može uneti kašnjenje u samom brisanju, te je moguće definisati koliko će koordinator čekati da se čuvanje slike obavi.

Inkrementalno čuvanje predstavlja kreiranje kopije SSTable datoteka neke tabele u trenutku kada se memtable strukture sačuvaju na disk u obliku tih datoteka. Tipično, inkrementalno čuvanje se radi u kombinaciji sa kreiranjem slika kako bi se optimizovalo vreme potrebno za čuvanje baze podataka, kao i smanjih potreban prostor na disku. Po podrazumevanim podešavanjima, inkrementalno čuvanje nije omogućeno. Bitno je pomenuti da se ovim procesom takođe čuva stanje sistemski tabela.

Slika baze podataka, zajedno sa svim inkrementalnim čuvanjima i dnevnikom upisa od trenutka kreiranja slike, čine kompletan „backup“ baze podataka.

Organizacija direktorijuma podataka

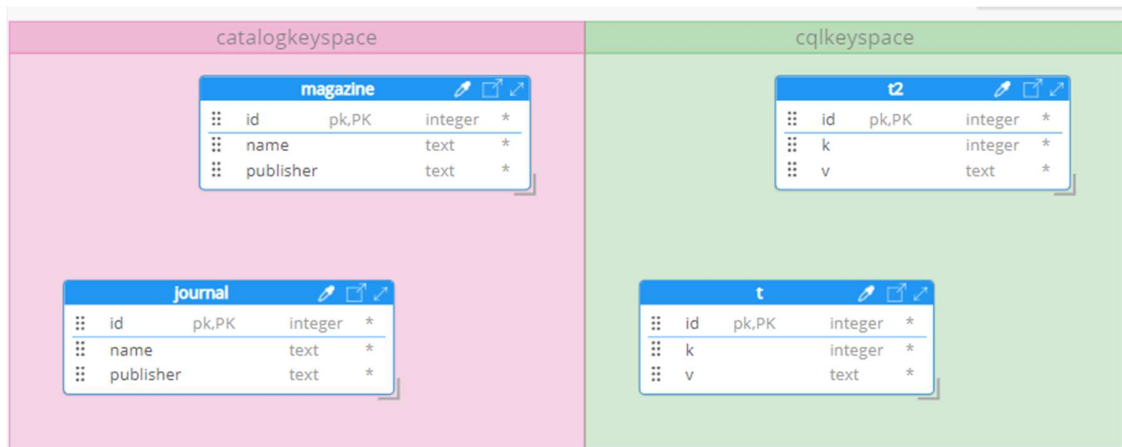
Struktura direktorijuma podataka u ovom sistemu se čuva hijerarhijski. Postoji poseban direktorijum za keyspace objekte, u okviru kog se čuvaju direktorijumi tabela. U okviru direktorijuma tabela se u posebnom direktorijumu čuvaju sami podaci (SSTable datoteka), dok postoje dva odvojena direktorijuma za slike tabele i inkrementalna čuvanja stanja tabele.



Slika 3. Struktura direktorijuma podataka u sistemu Apache Cassandra

Priprema baze podataka za demonstriranje postojećih mehanizma

Kako bi bilo moguće demonstrirati različite funkcionalnosti čuvanja baze podataka u sistemu Apache Cassandra, biće kreirana prosta baza podataka sa malom količinom podataka. Principi ostaju isti nezavisno od količine podataka u datoj bazi, kao i od kompleksnosti same baze. Kreirana baza će imati dva keyspace-a, sa po dve tabele, koje će imati po nekoliko redova podataka.



Slika 4. Šema baze podataka koja će biti korišćena u sledećim primerima

Slike baze podataka

Kreiraćemo sliku svih tabela u keyspace-u pod imenom „catalogkeyspace“ koristeći komandu *nodetool snapshot*:

```
# nodetool snapshot --tag catalog-ks catalogkeyspace
```

Requested creating snapshot(s) for [catalogkeyspace] with snapshot name [catalog-ks] and options {skipFlush=false}

Snapshot directory: catalog-ks

Zatim, možemo videti listu novokreiranih slika pomoću komande *nodetool snapshot*:

```
# nodetool listsnapshots
```

Snapshot Details:

Snapshot name	Keyspace name	Column family name	True size	Size on disk
---------------	---------------	--------------------	-----------	--------------

catalog-ks	catalogkeyspace	journal	0 bytes	5.89 KiB
------------	-----------------	---------	---------	----------

catalog-ks	catalogkeyspace	magazine	0 bytes	5.89 KiB
------------	-----------------	----------	---------	----------

Moguće je kreirati sliku samo jedne tabele u jednom keyspace-u (tabela „magazine“ iz keyspace-a „catalogkeyspace“):

```
# nodetool snapshot --tag magazine --table magazine catalogkeyspace
```

Requested creating snapshot(s) for [catalogkeyspace] with snapshot name [magazine] and options {skipFlush=false}

Snapshot directory: magazine

Takođe je moguće kreirati sliku više tabela iz jednog keyspace-a (tabele „t“ i „t2“ u „cqlkeyspace“ keyspace-u):

```
# nodetool snapshot --kt-list cqlkeyspace.t, cqlkeyspace.t2 --tag multi-table
```

Requested creating snapshot(s) for [cqlkeyspace.t, cqlkeyspace.t2] with snapshot name [multi-table] and options {skipFlush=false}

Snapshot directory: multi-table

Moguće je kreirati sliku istog para tabela tako da slika ima drugačiji naziv:

```
# nodetool snapshot --kt-list cqlkeyspace.t, cqlkeyspace.t2 --tag multi-table-2
```

Requested creating snapshot(s) for [cqlkeyspace.t, cqlkeyspace.t2] with snapshot name [multi-table-2] and options {skipFlush=false}

Snapshot directory: multi-table-2

Pomoću iste opcije koju smo koristili u prethodna dva primera, moguće je kreirati sliku tabela iz različitih keyspace-a:

```
# nodetool snapshot --kt-list catalogkeyspace.journal, cqlkeyspace.t --tag multi-ks
```

Requested creating snapshot(s) for [catalogkeyspace.journal, cqlkeyspace.t] with snapshot name [multi-ks] and options {skipFlush=false}

Snapshot directory: multi-ks

Izlistaćemo sve slike koje smo kreirali u prethodnim primerima:

```
# nodetool listsnapshots
```

Snapshot Details:

Snapshot name	Keyspace name	Column family name	True size	Size on disk
multi-table	cqlkeyspace	t	0 bytes	5.82 KiB
multi-table	cqlkeyspace	t2	0 bytes	5.85 KiB
multi-ks	cqlkeyspace	t	0 bytes	5.82 KiB
multi-ks	catalogkeyspace	journal	0 bytes	5.89 KiB
magazine	catalogkeyspace	magazine	0 bytes	5.89 KiB
multi-table-2	cqlkeyspace	t	0 bytes	5.82 KiB
multi-table-2	cqlkeyspace	t2	0 bytes	5.85 KiB
catalog-ks	catalogkeyspace	journal	0 bytes	5.89 KiB
catalog-ks	catalogkeyspace	magazine	0 bytes	5.89 KiB

Total TrueDiskSpaceUsed: 0 bytes

Moguće je pronaći direktorijume koji sadrže sve slike prostom pretragom:

```
# find -name snapshots
```

```
./var/lib/cassandra/data/cqlkeyspace/t-d021fe80e11e11ec8f3ec7dc38569b53/snapshots
```

```
./var/lib/cassandra/data/cqlkeyspace/t2-d61f80a0e11e11ec8f3ec7dc38569b53/snapshots
```

```
./var/lib/cassandra/data/catalogkeyspace/magazine-218932e0e12211ec8f3ec7dc38569b53/snapshots
```

```
./var/lib/cassandra/data/catalogkeyspace/journal-1d19a3c0e12211ec8f3ec7dc38569b53/snapshots
```

Kako bi pronašli sve direktorijume slika vezane za datu tabelu, moramo ući u jedan od izlistanih direktorijuma:

```
# cd ./var/lib/cassandra/data/catalogkeyspace/journal-1d19a3c0e12211ec8f3ec7dc38569b53/snapshots
# ls -l
total 8
drwxr-xr-x 2 cassandra cassandra 4096 May 31 21:24 catalog-ks
drwxr-xr-x 2 cassandra cassandra 4096 May 31 21:49 multi-ks
```

Kada uđemo u jedan od izlistanih direktorijuma slika, možemo videti sve kreirane kopije SSTable datoteka, kao i „schema.cql“ datoteku koja nosi informacije koje se kasnije mogu iskoristiti da bi se tabela rekreirala prilikom procesa restoracije iz slike:

```
# cd catalog-ks && ls -l
total 44
-rw-r--r-- 1 cassandra cassandra 31 May 31 21:24 manifest.json
-rw-r--r-- 3 cassandra cassandra 47 May 31 21:24 nb-1-big-CompressionInfo.db
-rw-r--r-- 3 cassandra cassandra 99 May 31 21:24 nb-1-big-Data.db
-rw-r--r-- 3 cassandra cassandra 9 May 31 21:24 nb-1-big-Digest.crc32
-rw-r--r-- 3 cassandra cassandra 16 May 31 21:24 nb-1-big-Filter.db
-rw-r--r-- 3 cassandra cassandra 16 May 31 21:24 nb-1-big-Index.db
-rw-r--r-- 3 cassandra cassandra 4768 May 31 21:24 nb-1-big-Statistics.db
-rw-r--r-- 3 cassandra cassandra 56 May 31 21:24 nb-1-big-Summary.db
-rw-r--r-- 3 cassandra cassandra 92 May 31 21:24 nb-1-big-TOC.txt
-rw-r--r-- 1 cassandra cassandra 893 May 31 21:24 schema.cql
```

Kako bi se obrisale slike, može se koristiti komanda *nodetool clearsnapshot*. Prilikom njenog korišćenja mora se specificirati ili ime konkretne slike, ili iskoristiti opcija *--all* zajedno sa imenom keyspace-a kako bi se obrisale sve slike vezane za njega:

```
# nodetool clearsnapshot -t magazine cqlkeyspace
```

Requested clearing snapshot(s) for [cqlkeyspace] with snapshot name [magazine]

```
# nodetool clearsnapshot --all cqlkeyspace
```

Requested clearing snapshot(s) for [cqlkeyspace] with [all snapshots]

Inkrementalno čuvanje

Proces inkrementalnog čuvanja se može omogućiti kroz određenu opciju u konfiguracionoj datoteci ili pozivom komande *nodetool enablebackup*. Nakon što se omogući inkrementalno čuvanje, ono se pokreće svaki put kada se neka memtable struktura snimi na disk. To se može ručno pokrenuti komandom *nodetool flush*:

```
# nodetool flush cqlkeyspace t
```

```
# nodetool flush cqlkeyspace t2
```

```
# nodetool flush catalogkeyspace journal magazine
```

Slično slikama, inkrementalna čuvanja se čuvaju u direktorijumima pod imenom „backups“ u okviru datih tabela. Mogu se naći prostom pretragom (ovakva pretraga bi izlistala i direktorijume vezane za sistemske tabele, jer se i njihova inkrementalna čuvanja izvršavaju, ali smo ih mi izbacili iz rezultata kako bi se fokusirali na „prave“ tabele):

```
# find -name backups
```

```
./var/lib/cassandra/data/cqlkeyspace/t-d021fe80e11e11ec8f3ec7dc38569b53/backups
```

```
./var/lib/cassandra/data/cqlkeyspace/t2-d61f80a0e11e11ec8f3ec7dc38569b53/backups
```

```
./var/lib/cassandra/data/catalogkeyspace/magazine-218932e0e12211ec8f3ec7dc38569b53/backups
```

```
./var/lib/cassandra/data/catalogkeyspace/journal-1d19a3c0e12211ec8f3ec7dc38569b53/backups
```

Kada bi sada proverili sadržaj nekog od ovih folder, videli bi da su prazni. Inkrementalna čuvanja se pokreću samo kada se sadržaj tabele promeni. Ako dodamo novi red u tabelu „journal“ i zatim pokrenemo inkrementalno čuvanje, videćemo sledeći sadržaj u „backups“ direktorijumu vezanom za datu tabelu:

```
cd ./var/lib/cassandra/data/catalogkeyspace/journal-1d19a3c0e12211ec8f3ec7dc38569b53/backups
ls -l
total 36
-rw-r--r-- 2 cassandra cassandra 47 May 31 22:34 nb-2-big-CompressionInfo.db
-rw-r--r-- 2 cassandra cassandra 77 May 31 22:34 nb-2-big-Data.db
-rw-r--r-- 2 cassandra cassandra 9 May 31 22:34 nb-2-big-Digest.crc32
-rw-r--r-- 2 cassandra cassandra 16 May 31 22:34 nb-2-big-Filter.db
-rw-r--r-- 2 cassandra cassandra 16 May 31 22:34 nb-2-big-Index.db
-rw-r--r-- 2 cassandra cassandra 4768 May 31 22:34 nb-2-big-Statistics.db
-rw-r--r-- 2 cassandra cassandra 56 May 31 22:34 nb-2-big-Summary.db
-rw-r--r-- 2 cassandra cassandra 92 May 31 22:34 nb-2-big-TOC.txt
```

Ako se doda još jedan red u tabelu i pogled sadržaj „backups“ direktorijuma, videćemo sledeće:

```
total 72
-rw-r--r-- 2 cassandra cassandra 47 May 31 22:34 nb-2-big-CompressionInfo.db
-rw-r--r-- 2 cassandra cassandra 77 May 31 22:34 nb-2-big-Data.db
-rw-r--r-- 2 cassandra cassandra 9 May 31 22:34 nb-2-big-Digest.crc32
-rw-r--r-- 2 cassandra cassandra 16 May 31 22:34 nb-2-big-Filter.db
-rw-r--r-- 2 cassandra cassandra 16 May 31 22:34 nb-2-big-Index.db
-rw-r--r-- 2 cassandra cassandra 4768 May 31 22:34 nb-2-big-Statistics.db
-rw-r--r-- 2 cassandra cassandra 56 May 31 22:34 nb-2-big-Summary.db
-rw-r--r-- 2 cassandra cassandra 92 May 31 22:34 nb-2-big-TOC.txt
-rw-r--r-- 2 cassandra cassandra 47 May 31 22:40 nb-3-big-CompressionInfo.db
-rw-r--r-- 2 cassandra cassandra 61 May 31 22:40 nb-3-big-Data.db
-rw-r--r-- 2 cassandra cassandra 10 May 31 22:40 nb-3-big-Digest.crc32
-rw-r--r-- 2 cassandra cassandra 16 May 31 22:40 nb-3-big-Filter.db
-rw-r--r-- 2 cassandra cassandra 8 May 31 22:40 nb-3-big-Index.db
-rw-r--r-- 2 cassandra cassandra 4764 May 31 22:40 nb-3-big-Statistics.db
-rw-r--r-- 2 cassandra cassandra 56 May 31 22:40 nb-3-big-Summary.db
-rw-r--r-- 2 cassandra cassandra 92 May 31 22:40 nb-3-big-TOC.txt
```

Sve datoteke koje su postojale malopre su se duplirale. Jedna od loših strana inkrementalnog čuvanja je to što se potencijalno može ostati bez prostora.

Povratak stanja baze podataka

Slika baze se sastoji iz praktično istih SSTable datoteka kao i inkrementalno čuvanje, gde slika takođe sadrži prethodno pomenuti „schema.sql“ datoteku. Shodno tome, kako bi se izvršio pravi povratak stanja u slučaju da je tabela obrisana, mora se koristiti prethodno kreirana slika.

Dve glavne komande za povratak stanja su *sstableloader* i *nodetool import*, ili kroz prosto kopiranje datoteka između direktorijuma. Zabeležićemo trenutno stanje tabele „journal“ kako bi kasnije proverili proces povratka njenog stanja:

id	name	publisher
1	Couchbase Magazine	Couchbase
0	Apache Cassandra Magazine	Apache Cassandra
2	Couchbase Magazine	Couchbase
3	Couchbase Magazine	Couchbase

Obrisaćemo sve datoteke iz direktorijuma vezanog za tabelu „journal“ (ostavićemo direktorijume „backups“ i „snapshots“ kako bi mogli da rekreiramo tabelu. Zatim, prekopiraćemo datoteke nađene u „snapshots/catalog-ks“ direktorijumu slike „catalog-ks“, koju smo malopre kreirali, u glavni direktorijum tabele. Iskoristićemo komandu *nodetool refresh* kako bi ponovo učitali tabelu.

```
# rm *
```

```
# ls
```

```
backups snapshots
```

```
# cp -r ./snapshots/catalog-ks/. .
```

```
# ls
```

```
backups    nb-1-big-CompressionInfo.db  nb-1-big-Digest.crc32  nb-1-big-Index.db    nb-1-big-  
Summary.db  schema.cql
```

```
manifest.json  nb-1-big-Data.db          nb-1-big-Filter.db    nb-1-big-Statistics.db  nb-1-big-TOC.txt  
snapshots
```

```
# nodetool refresh catalogkeyspace journal
```

Ako sada pogledamo tabelu, videćemo da u njoj fale nekoliko redova koje smo dodali nakon kreiranja slike „catalog-ks“:

id	name	publisher
1	Couchbase Magazine	Couchbase
0	Apache Cassandra Magazine	Apache Cassandra

Kako bi povratili ta dva reda, prekopiramo datoteke iz direktorijuma „backups“ u glavni direktorijum tabele:

```
# cp -r ./backups/. .
```

```
# nodetool refresh catalogkeyspace journal
```

Novi sadržaj tabele:

id	name	publisher
1	Couchbase Magazine	Couchbase
0	Apache Cassandra Magazine	Apache Cassandra
2	Couchbase Magazine	Couchbase
3	Couchbase Magazine	Couchbase

Citirana dela

- [Na mreži]. Available:
- 1] <https://jiankaiwang.gitbooks.io/itsys/content/cassandra/architecture.html>.

[Na mreži]. Available: [https://docs.datastax.com/en/cassandra-](https://docs.datastax.com/en/cassandra-oss/3.x/cassandra/architecture/archIntro.html)

 - 2] [oss/3.x/cassandra/architecture/archIntro.html](https://docs.datastax.com/en/cassandra-oss/3.x/cassandra/architecture/archIntro.html).

[Na mreži]. Available: [https://www.red-gate.com/simple-](https://www.red-gate.com/simple-talk/databases/nosql/apache-cassandra-data-modeling-and-query-best-practices/)

 - 3] [talk/databases/nosql/apache-cassandra-data-modeling-and-query-best-practices/](https://www.red-gate.com/simple-talk/databases/nosql/apache-cassandra-data-modeling-and-query-best-practices/).

[Na mreži]. Available:

 - 4] <https://cassandra.apache.org/doc/latest/cassandra/architecture/overview.html>.

[Na mreži]. Available:

 - 5] <https://cassandra.apache.org/doc/latest/cassandra/operating/backups.html>.