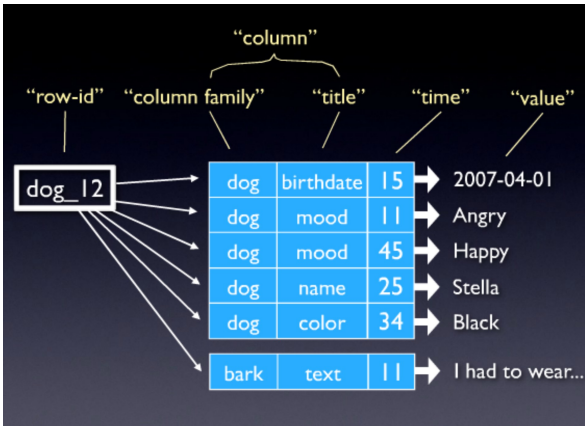
3. Wide column stores / extensible records

23 November 2021 18:16

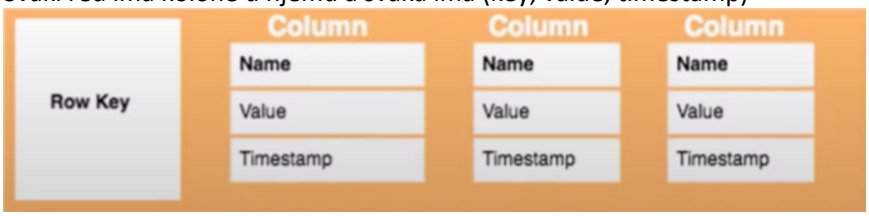
Kad imamo podatke koji spadaju u mali broj kategorija (7 dana u nedelji, 12 meseci u godini itd.)

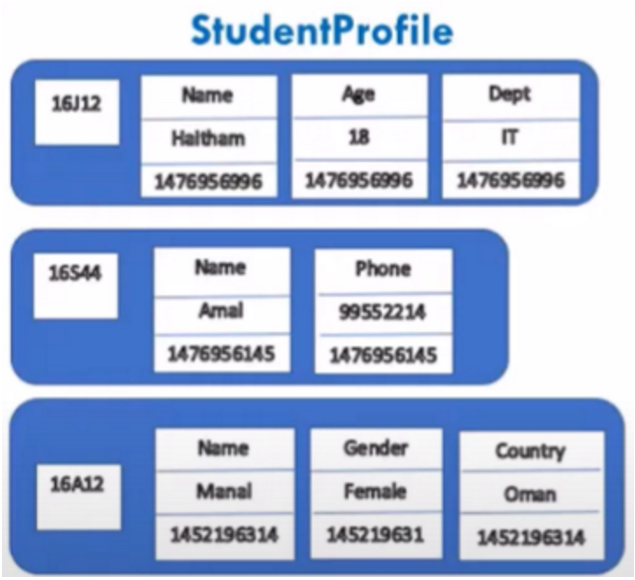
Pamti se po kolinama - drugacija organizacija



Kod relacionih imamo sve kolone cak I ako imamo null, ovde nemamo null jer ne pamtimo
Ako ne znamo koje kolone imamo mozemo samo da napravimo
Sve informacije mogu da budu razlicite samo kljuc je isti
Ako neki kljuc ima vise informacija istog tipa samo dodamo kolonu za svaku info
5 br telefona za jedno ime ima 5 razlicitih kolona
Smanjeno spajanje sa drugom tabelom jer moze sve da se cuva ovde (moze da se desi)

BigTable
Google 2006
Rasuta, distribirana, perzistentna multi-dimenzionalan sortirana mapa
Mapa - hesh
Multi- dimenzionalna -
(row, column, timestamp) dimenzije
Svaki red ima kolone u njemu a svaka ima (key, value, timestamp)



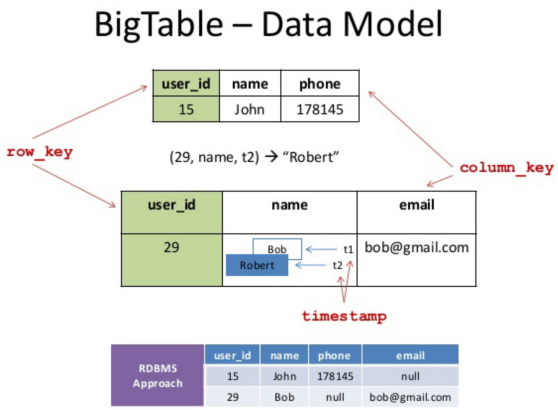


Timestamp - najnovija ili predhodna verzija podatka

Distribuirana - sortirana po kljucu, ako jedan kljuc ne moze sve podatke da sacuva radi se particionisanje po kljucu

Rasuta - sve kolone nisu pune, nema null, ne mora svi elementi da postoje

Cuva blobove



Vrste

- Sortitani leksikografski po kljucu vrste (sortirani po imenu kao u recniku)
- Vrste koje su blizu jedna do druge su obicno na istom serveru ili malom br servera
- Pristup kolonama je atomican - operacije pisanja I citanja se odvijaju bez prekida

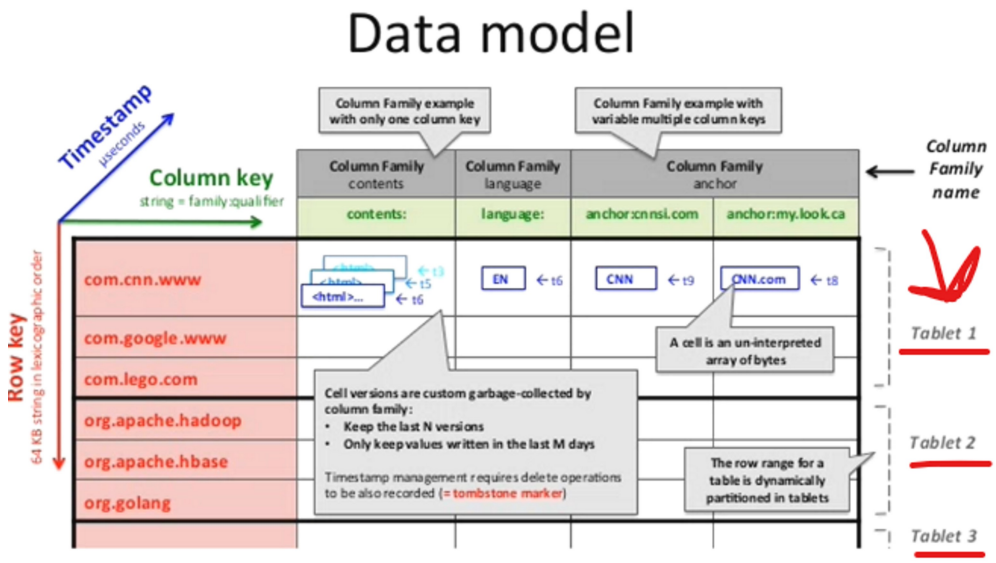
Kolone

- Familije kolona - cuva informacije koje su povezane zajedno
  - Cela familija se cuva I kompresuje zajedno

Timestamp

- Svaka celija moze imati vise vrednosti
- Moze se rucno dodeliti I promeniti

Tableti - particije redova u modelu



Range scan operacije su efikasne

Kad trazimo opseg informacija od do

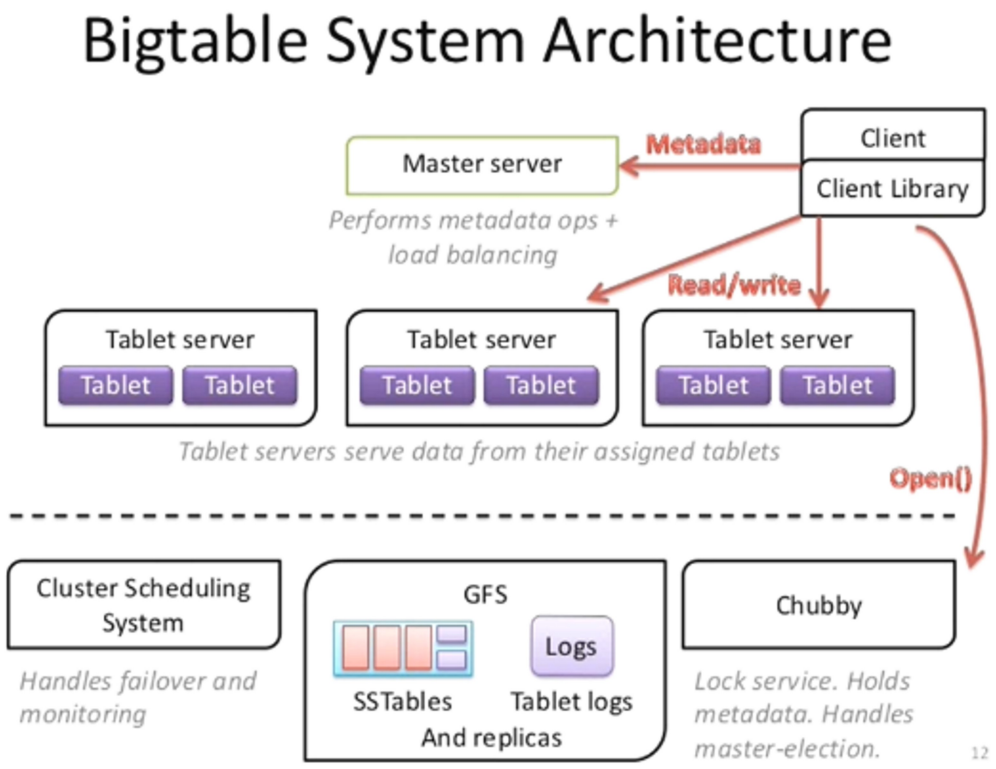
Podaci koji se retko koriste su kompresovani radi ustede memorije

Verzije

Automatski garbage collection

Cuva se zadnjih N verzija

Cuvaju se samo verzije novije od datuma nekog



Master server - cuva metadata o svemu

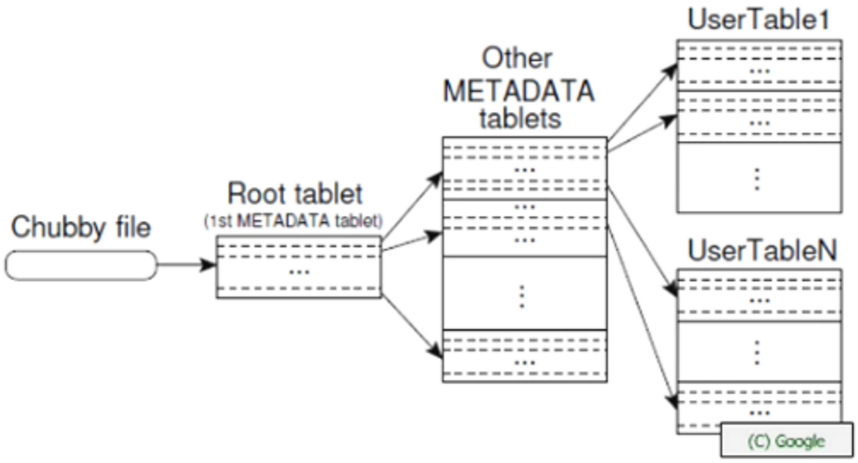
Gde je koj tablet, info o garbage collection itd

Chubby - omogucava kontrolu pristupa deljivim resursima u distribuiranom sistemu

Lock service

U njemu se cuva lokacija root tableta I svi mu se obracaju za pocetak pretrage

Posle se u root nalazi drugi metadata, pa sledeci dok ne nadjemo podatak

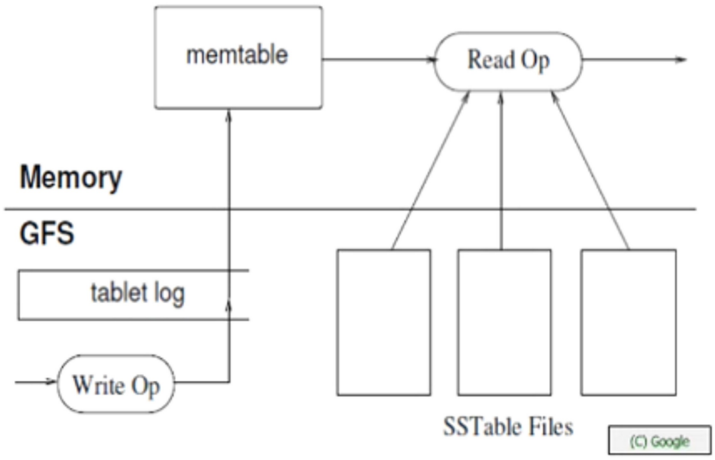


Tablet operacije

Read, Write, Split

Split - kad tablet postane mnogo veliki deli se na nove tablete

Pristup tabletu



Tablet log

SSTable fajlovi - cuva podatke

tabele sortiranih stringova (stringovi su kljuc, vrednost)

Memtable - kesh

Read Op

Prvo se trazi u kesh pa u storige ako nije u kesh

Za pretrazivanje se koristi bloom filter

Vraca verovatnocu da je podatak tamo gde ga trazimo I gde je javljeno pozitivno nalazenje al ne mora da znaci da jeste

Write Op

podatak upise u log

Podatak se upise u kesh (memtable)

Ovo radimo radi velike brzine

Problemi kod sinhronizacije zbog brzine mogu da se dese

Periodicno se podaci iz kesha prebacuju u SSTable (storage)

Ako se desi greska u log ima dovoljno info da se ponovi upis

Casandra

Nema single point of failure

Osobine:

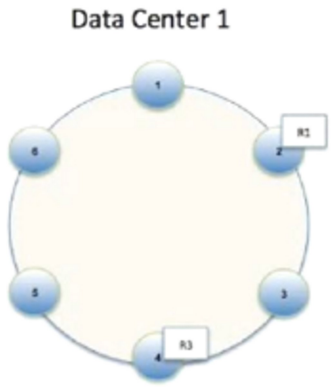
- High preformance
- Distributed & decentralized - svaki podatak ima na vise cvorova kao replika
- Elastic scalability - lako se dodaju novi cvorovi u sistem
- Fault tolerance
- Tunable consistency
- Column orented
- CQL query interface - casandra query language, podseca na relacione upite

Operacije:

- Write
  - Salje se zahtev cvoru I on postaje kordinator cvor
  - On salje svim ostalim cvorovima zahtev za pisanje
  - Bitno je da vecina vrati pozitivan odgovor, netrebaju svi
    - Ako je jedan nedostupan primenjuje se tehnika hinted handoff
- Read
  - Zahtev se salje geografski najblizem cvoru se salje I on kontaktira sve cvorove koji imaju podatak
  - Cita se prvo iz memtable pa iz SSTable
  - Read repair ako ima konflikt resava se

- Vise forsira write nego read jer kod read ima resavanje komflikata

Za razliku od relacionih sam kljuc moze da bude podatak ovde



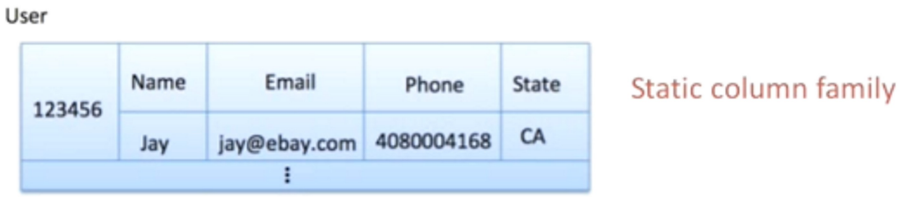
Keyspace

- Atributi:
- Faktor replikacije
  - Podaci se cuvaju u 3 + noda
- Faktor distribucije replikacija
  - Nodovi obicno nisu na istoj geografskoj lokaciji
  - Ako su 2 kopije podatka na istom geografskom mestu pazi se na rack (vitrina servera)
    - Ista kopija podatka da nije u istom rack-u

Particionisanje podataka se radi na osnovu hesh vrednosti kljuca
Lako trazenje podataka

Svaki podatak je cetvorodimenzionalna hash mapa
[Keyspace][Column family][Ke][Column]

Tipovi vrsta:
Wide rows - veliki br kolona mali br vrsta
Skinny rows - mali br kolona veliki br vrsta
Tipovi column families:
Standardne - kolone I super kolone
Super column family - samo super kolone



Prvi bi koristio sekundarni kljuc da spoji 2 tabele
Drugi ima sve na jednom mestu a podatak dole ne mora da je ime proizvoda vec moze da je html, json …

Prvo je super column
Drugo ima sve informacije u jednoj tabeli
Izbegava se super kolona stavljanjem kategorije ispred imena I komparator ce informacije jer pocinju isto da sortira jedno do drugo (ovime izbegavamo super kolone)

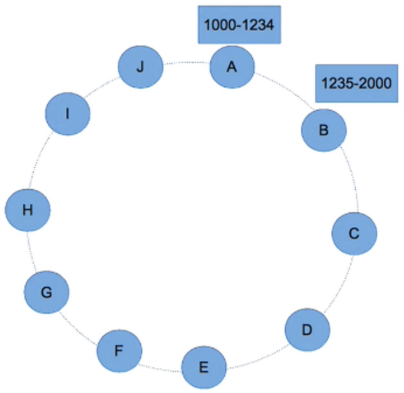
Kljucevi:
Universal Unique ID
Time stamp

Primary key = Partition key, Clustering key

Partition key - cvor/particija gde ce podatak se smestiti

Clustering key - definise kolone po kojima ce se podaci sortirati

Hesh-uju se



ABC... - nodes

1000-1234 - partitioning keys ma tom nodu

Primena

Katalozi proizvoda

Playliste

Podaci senzora

Vremenske serije podataka

Detekcija prevara

Kesiranje podataka

Sve moze da se upakuje u jednu vrstu (listu playliste)

Lose koriscenje

Stroga konzistentnost - jer ima vise nodova pa moze doci do problema

ACID transakcije - atomicnost, konzistentnost, izolacija, trajnost

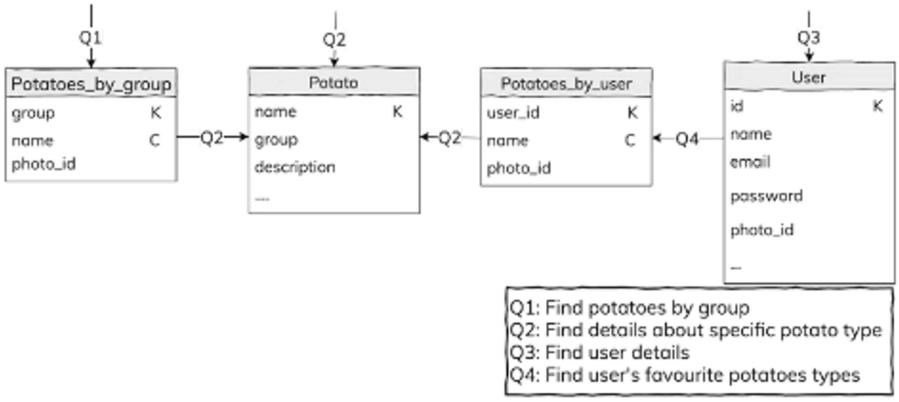
Funkcije agregacije - min, max, sum, avg, count

Pretraga bez primarnog kljuca - moze al treba da pretrazi sve podatke svih cvorova I presporo je

Veliki br citanja podataka - vise voli pisanje nego citanje

Puno brisanje I azuriranje podataka - voli pisanje I kozistentnost podataka radi pretrage

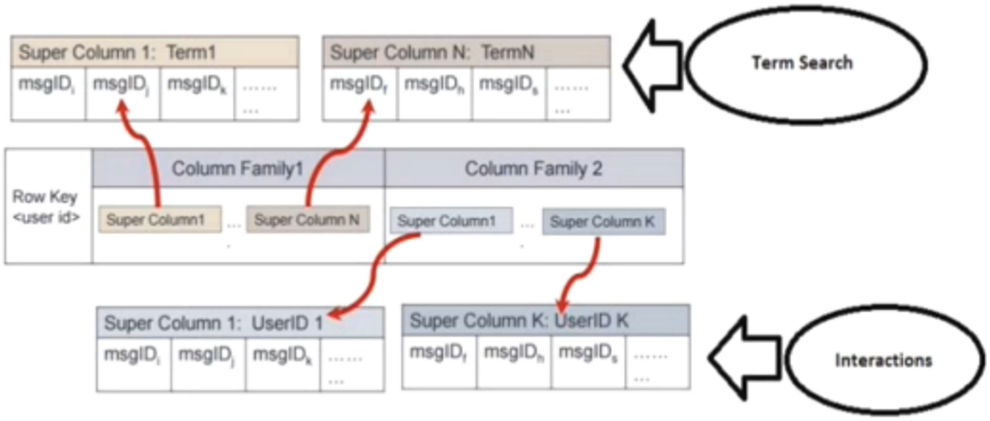
- Chebotko diagram



Svaka pretraga vraca novu tabelu

Tabele nemaju konekcije izmedju njih, za svaki spoj se pravi tabela nova koja simulira spoj

Ima redundantne podatke ne izbegavaju se



Column family 1 je za pretragu termina

Termini se koriste kao kljucevi za pretragu a ispod msgID je drugi termin povezan sa prvobitnim terminom

Column family 2 je za pretragu chatova

Koristnik je kljuc a posle msgID je ID drugog korisnika I poruka koje su razmenili