

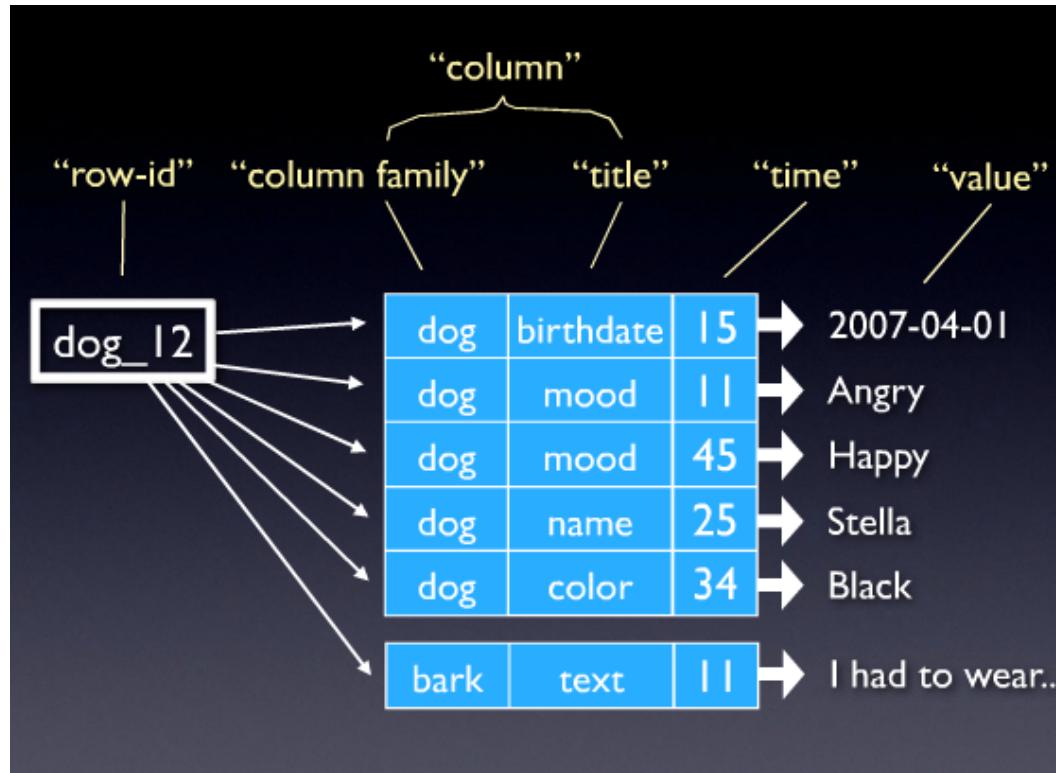
Računarstvo i informatika

*Katedra za računarstvo
Elektronski fakultet u Nišu*

Napredne baze podataka Wide column stores

Zimski semestar 2020/2021

Column stores



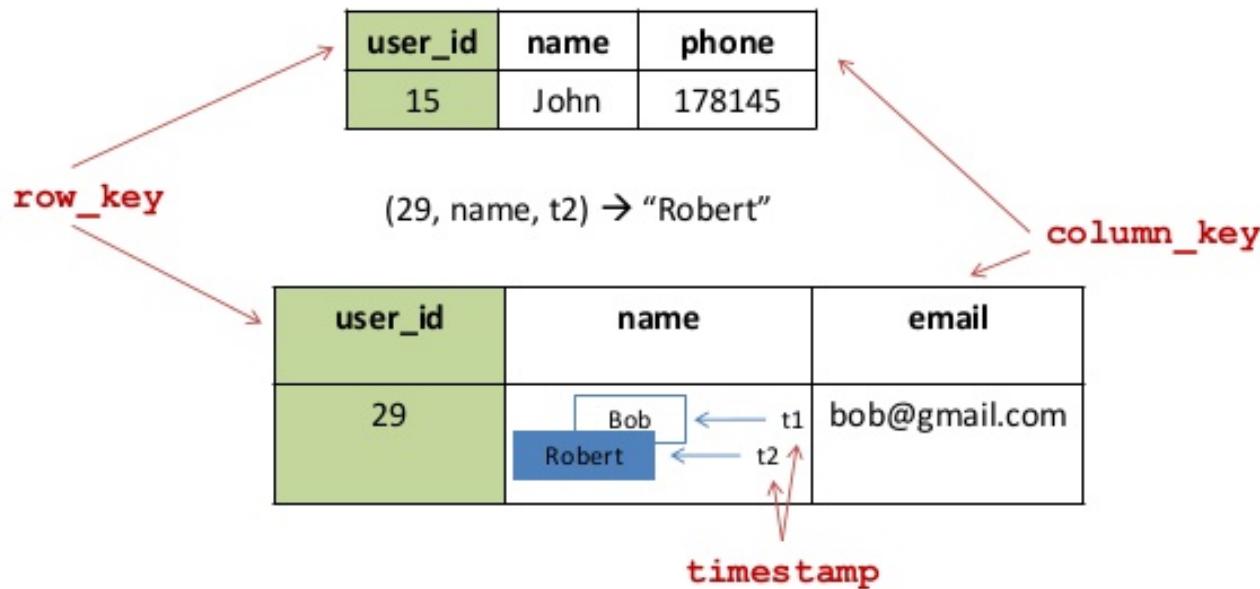
BigTable

- Google, 2006. godine

- Rasuta, distribuirana, perzistentna multi-dimenzionalan sortirana mapa
- $(\text{row}, \text{column}, \text{timestamp})$ dimenziije
- Vrednost je nestruktuirani niz bajtova
- Osnovne karakteristike:
 - Hibridno row/column skladište
 - Jedna master kopija (stand-by replika)
 - Verzije

BigTable model podataka

BigTable – Data Model

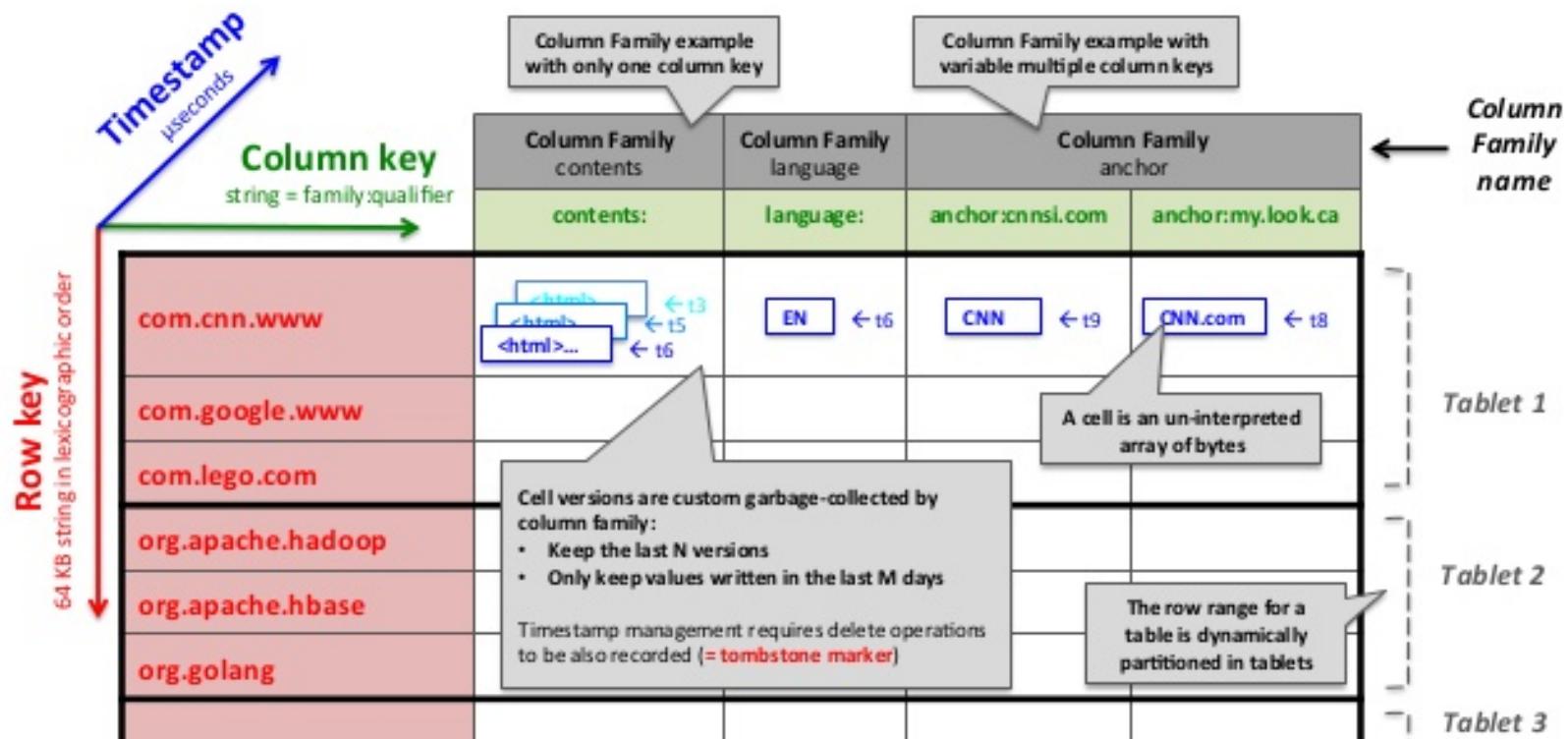


RDBMS Approach	user_id	name	phone	email
	15	John	178145	null
	29	Bob	null	bob@gmail.com



BigTable model podataka

Data model





BigTable model podataka

- **Vrste**
 - Ključevi vrsta su proizvoljni stringovi (max. veličina 64KB)
 - Podaci su leksikografski sortirani po ključu vrste
 - Leksikografski bliske vrste se obično nalaze na istom server ili na malom broju server
 - Pristup kolonama u vrsti je atomičan
- **Kolone, Familije kolona**
 - Ključevi kolona su proizvoljni stringovi
 - Neograničeni broj kolona
 - Familiji kolona se pridružuje informacija o tipu podataka
 - Podaci u familiji kolona se čuvaju i kompresuju zajedno
 - Kontrola pristupa je implementirana na nivou familija kolona



BigTable model podataka

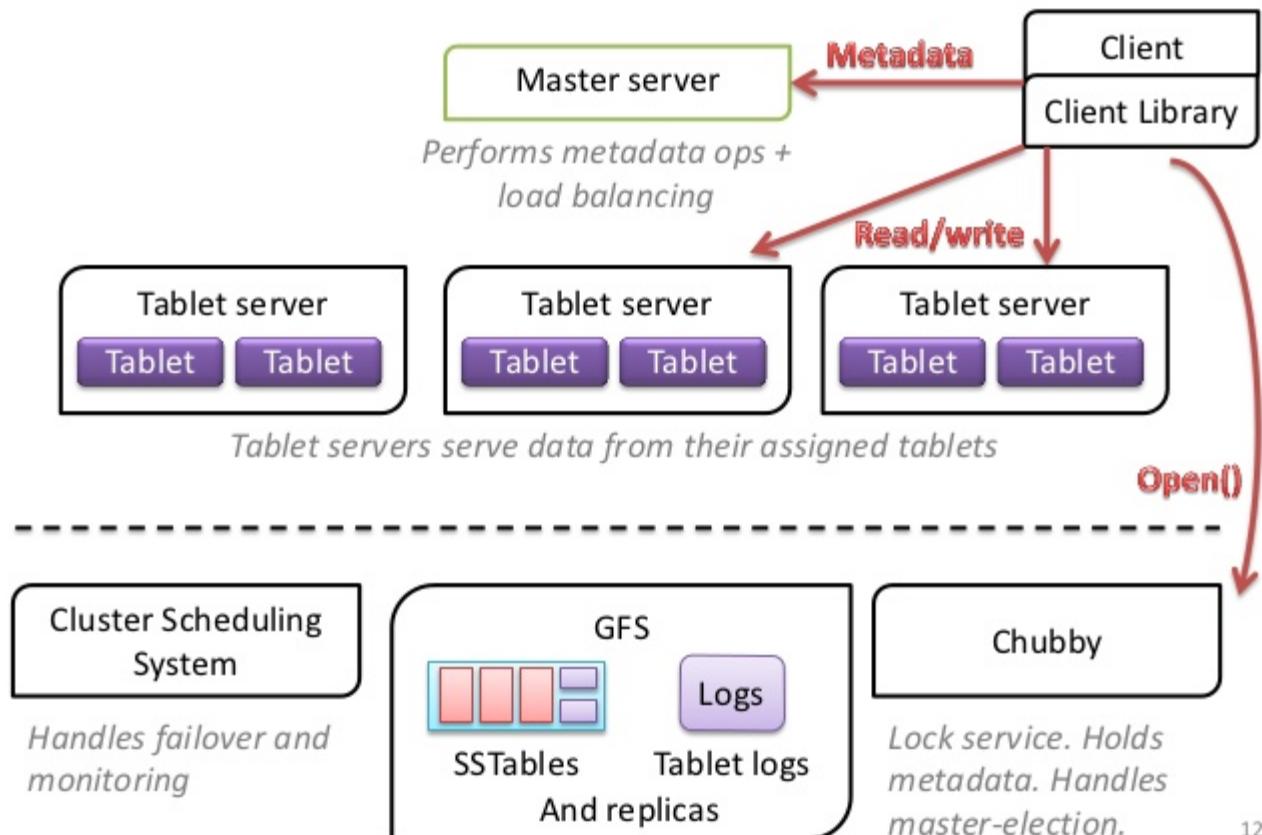
- **Timestamp**
 - Svaka ćelija može da ima više vrednosti
 - Timestamp se može ručno promeniti/dodeliti
- **Tableti**
 - Opseg vrsta je dinamički particionisan u tablete (sekvence vrsta)
 - Range scan operacije su jako efikasne
 - Ključevi vrsta se biraju tako da povećaju lokalnost operacija pristupa podacima (bliski podaci treba da budu u istom tabletu)
 - Velična tableta približno 100 - 200GB, nakon toga se podaci kompresuju

BigTable model podataka

- Verzije
 - Automatski garbage collection
 - Čuva se N poslednjih verzija
 - Čuvaju se samo verzije novije od zadatog timestamp-a

BigTable arhitektura

Bigtable System Architecture



BigTable arhitektura

- Arhitektura
 - Podaci se skladište u Google file system (GFS)
 - Chubby (distribuirani lock servis, Paxos)
 - Map Reduce
 - 1 Master server
 - Veliki broj tablet servera

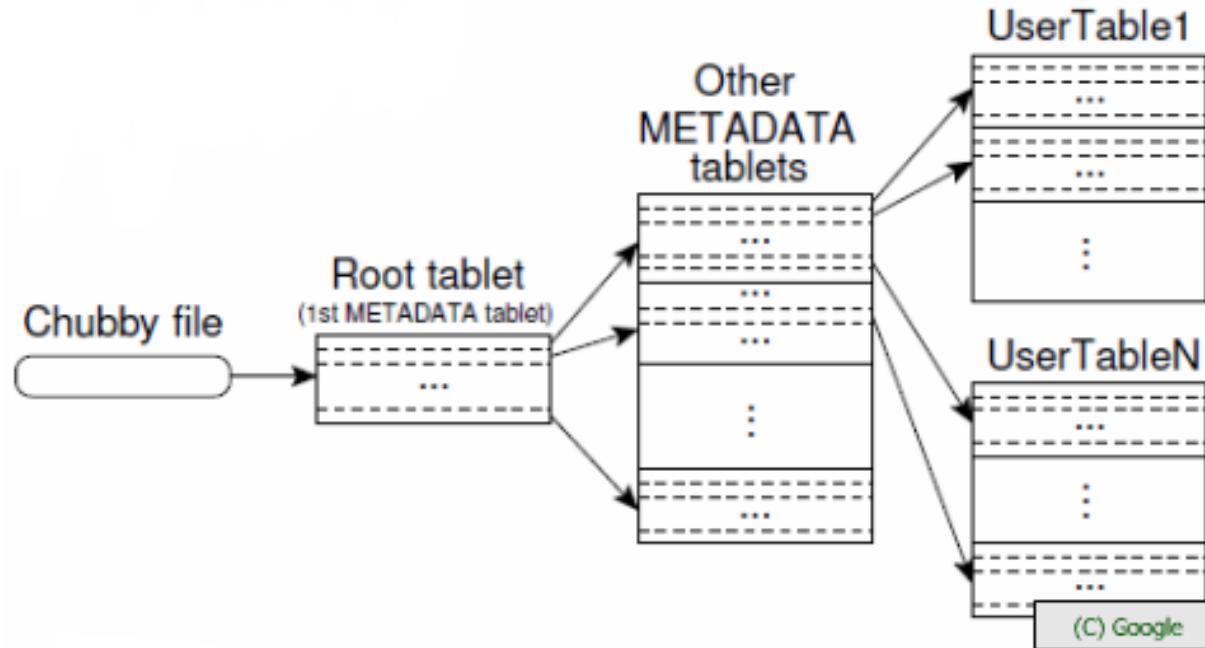
BigTable arhitektura

- Master server
 - Operacije sa metapodacima
 - Balansira opterećenje tablet servera
 - Garbage collection
 - Upravljanje šemom
 - Klijent ne pristupa master serveru već direktno tablet serverima
 - Master server ne upravlja lokacijom tableta
- Tablet server
 - Veliki broj tablet servera
 - Upravljuju Read/Write/Split operacijama nad tabletim

BigTable workflow

- Lokacija tableta
 - B⁺ stablo
 - Root (Chubby file) – lokacija Root Metadata tableta
 - Prvi nivo (Root Metadata tablet) – lokacija svih metadata tableta
 - Drugi nivo (Metadata tablet) – lokacija skupa tableta sa podacima
 - Klijenti keširaju lokacije podataka
 - Root Metadata tablet se često nalazi na zasebnom serveru kako ne bi predstavljaо usko grlo

BigTable workflow





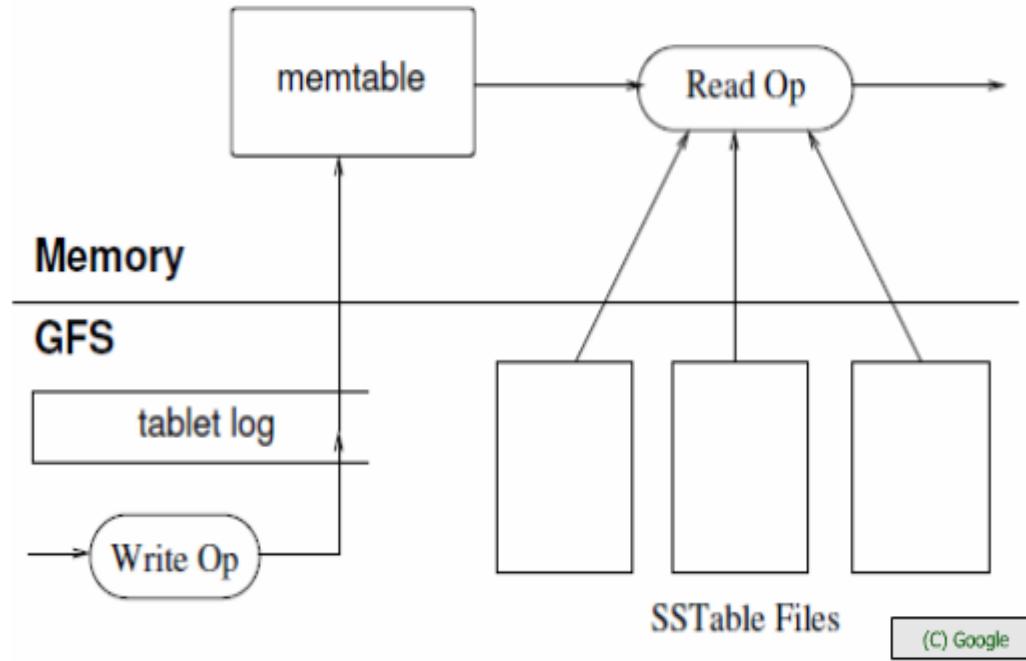
BigTable workflow

- Pristup tabletu
 - SSTable (sorted strings table)
 - perzistenta, uređena, nepromenljiva mapa ključevi – vrednost
 - Ključevi i vrednosti su stringovi
 - Sadrži sekvencu blokova i indeks bloka
 - Kompresija se vrši na nivou bloka
 - Dvofazna kompresija, 10:1
 - Ažuriranje
 - Ažurira se redo-log
 - Ažurira se in-memory buffer (memtable)
 - Ukoliko je memtable buffer pun, kreira se nova SSTable i memtable se prazni

BigTable workflow

- Čitanje
 - Izvršava se nad memtable i svim SSTable
 - Pristup memtable je izuzetno efikasan (podaci su unapred sortirani)
 - Koriste se Bloom filteri – mape koje definišu da li određena SSTable sadrži odgovarajuću ćeliju (vrsta, kolona) - drastično smanjen broj pristupa fajl sistemu

BigTable workflow



Cassandra

- Alexandra ili Cassandra
- Kralj Priam i kraljica Hecuba (Troja)
- Proročica koju su bogovi prokleti da joj ljudi ne veruju.
- Fortuneteller of Doom – proročanstva bila tačna ali joj niko nije verovao



Cassandra

- Open source distribuirani sistem za upravljanje bazom podataka
- Podržava upravljanje velikom količinom distribuiranih podataka bez postojanja **single point of failure**.
- Inicijalno razvijen od strane Facebook-a (Facebook Inbox search)
- Bazirana na Google BigTable i Amazon Dynamo rešenjima
- Open source projekat od 2008 godine
- Apache projekat od 2009 godine
- Danas važi za jedno od najpopularnijih NoSQL rešenja (skalabilnost, dostupnost, performanse).

Cassandra

Amazon Dynamo
(architecture)



Google BigTable
(data model)



- DHT
- Eventual consistency
- Tunable trade-offs, consistency

- Values are structured and indexed
- Column families and columns

Cassandra

The Evolution of Cassandra



2005

Data Model

- Wide rows, sparse arrays
- High performance through very fast write throughput.



2006

Infrastructure

- Peer-Peer Gossip
- Key-Value Pairs
- Tunable Consistency

facebook



Instagram

- Originally for Inbox Search
- But now used for Instagram



Apache



Cassandra



2008: Open-Source Release / 2013: Enterprise & Community Editions

Cassandra

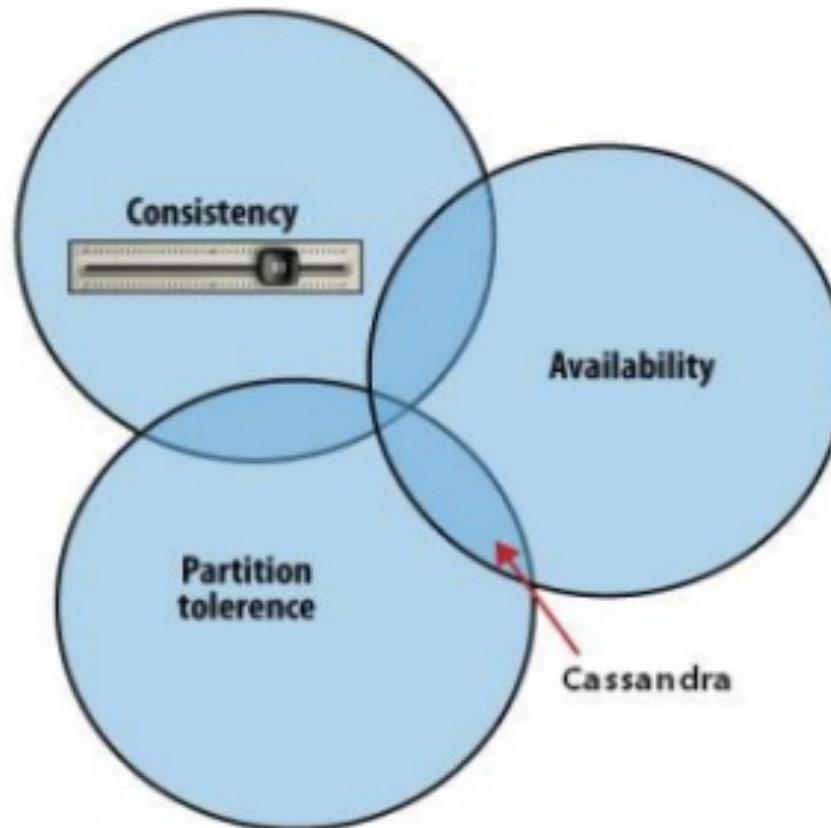


Cassandra

- Distribuirana i decentralizovana
 - Distribuiranost: izvršava se na većem broju čvorova
 - Decentralizovana: P2P arhitektura (gossip protokol), ne postoji signle point of failure.
- Elastična skalabilnost
 - Horizontalna skalabilnost
 - Jednostavno dodavanje novih čvorova
- Visoka dostupnost
 - Veliki broj nodova u klasteru, linearna skalabilnost
- Otpornost na otkaze
 - Automatsko prepoznavanje otkaza
 - Drugi čvorovi preuzimaju funkciju čvora koji je otkazao

Cassandra

- Cassandra ispunjava A i P iz CAP teoreme
- C može da se podešava (Eventual consistency)

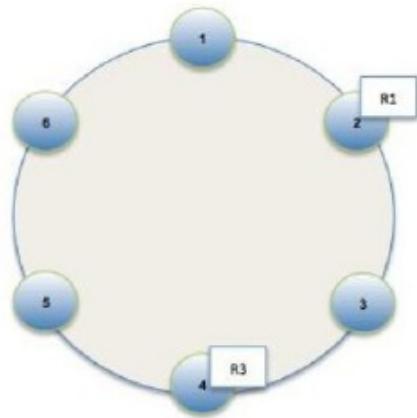




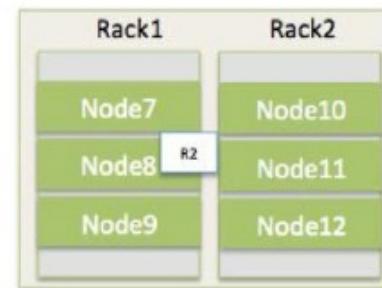
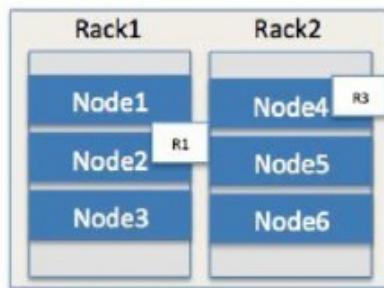
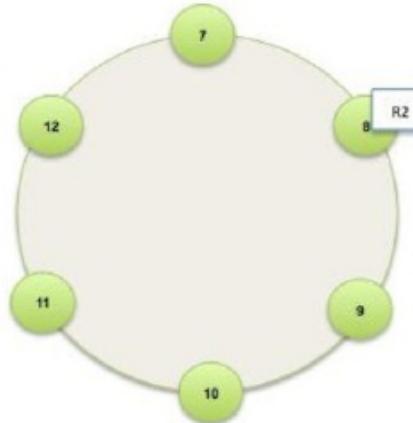
0

Cassandra

Data Center 1



Data Center 2



Replica for a column family row key

Cassandra

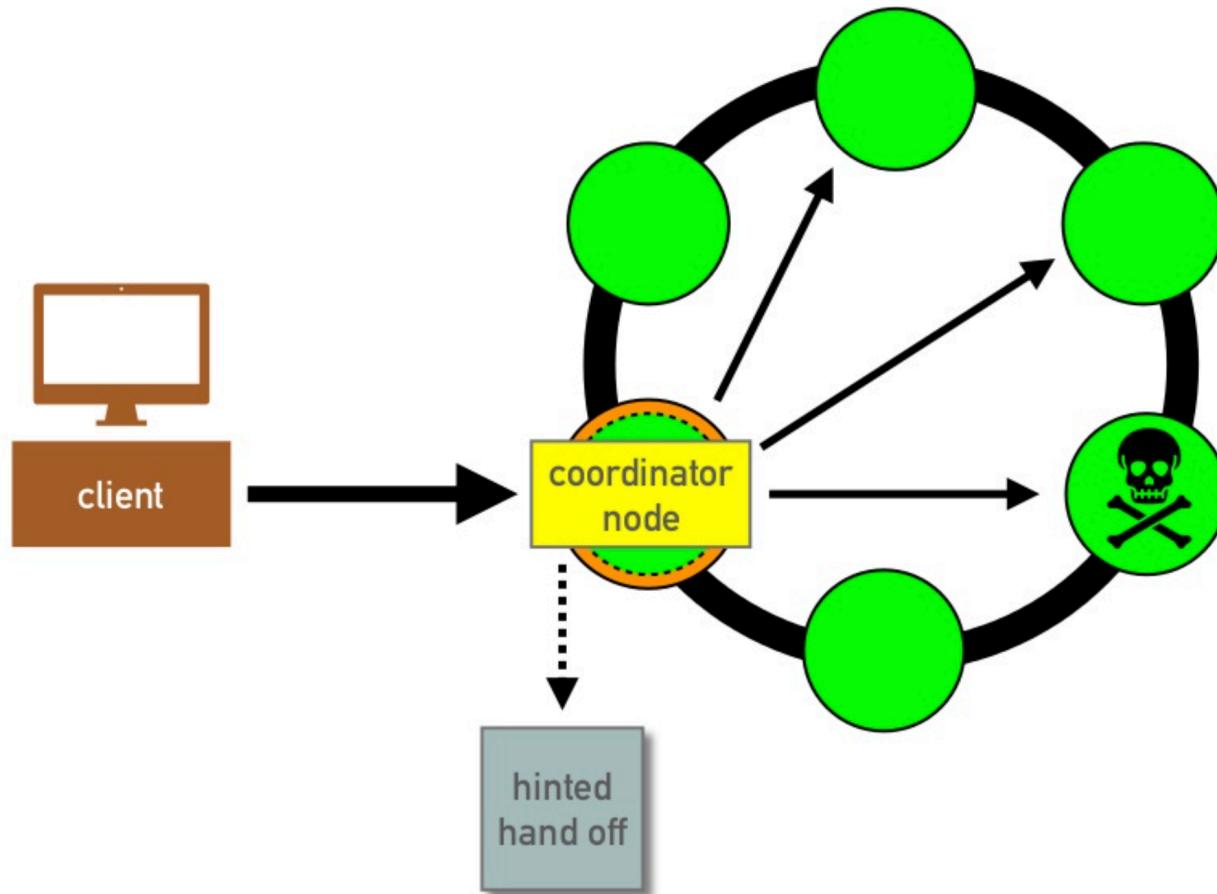
- Pripadnost klasteru
 - Gossip protokol – svaki čvor komunicira sa 1-3 susedna čvora o stanju klastera (razmenjuje informacije)
 - Promene u klasteru (dodavanje/uklanjanje čvorova, otkazi) se brzo propagiraju
 - Tehnike zasnovane na verovatnoći za otkrivanje otkaza
- Dinamičko particionisanje
 - Konzistentan hash mehanizam
 - Prsten čvorova
 - Čvorovi mogu da menjaju poziciju u prstenu zbog balansiranja opterećenja

Cassandra - operacije

- Write (gotovo == BigTable)
 - Klijenti šalju zahteve random čvoru, čvor koji primi zahtev određuje čvor odgovoran za podatke
 - Podaci su replicirani u N čvorova
 - Podaci se prvo dodaju u commit log, zatim u memtable i na kraju u SSTable
- Reads (gotovo == BigTable)
 - Zahtev se šalje random čvoru, čvor koji primi zahtev ga prosleđuje ka N čvorova koji poseduju podatke
 - Podaci se prvo čitaju iz memtable, nakon toga iz SSTable, Koriste se Bloom filteri.
- Komapkcije (== BigTable)

Cassandra - operacije

- WRITE operacija (na nivou kompletног klastera)

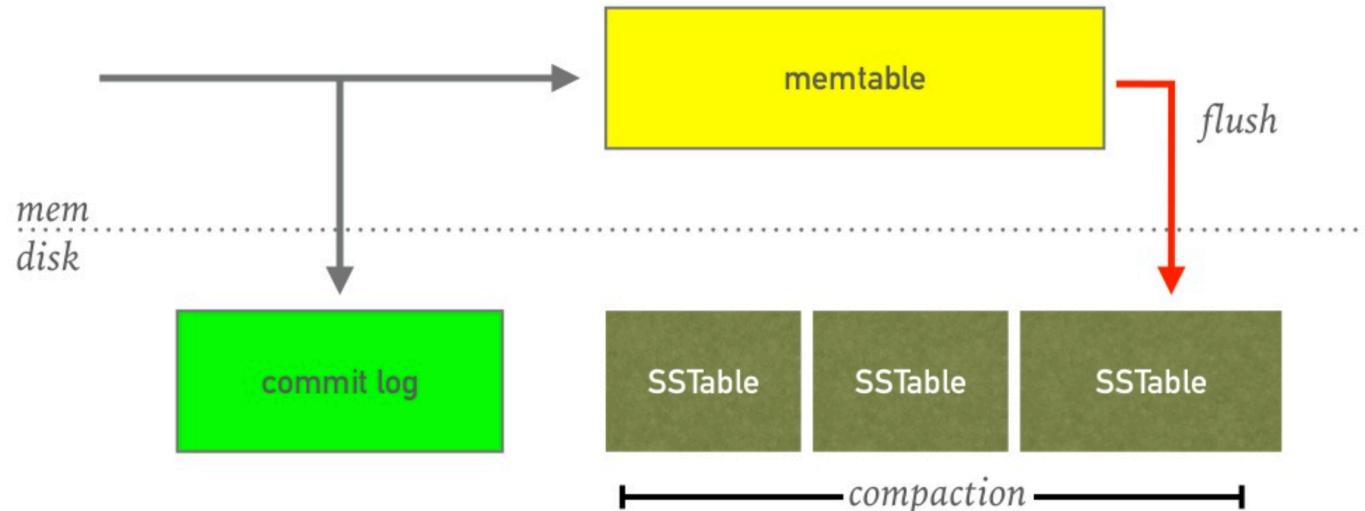


Cassandra - operacije

- WRITE operacija - podržani nivo konzistentnosti
 - ONE – Samo jedna replika mora da potvrdi
 - TWO, THREE - Dve/tri replike moraju da potvrde
 - QUORUM – Većina replika, koja predstavlja $(N/2+1)$ repliku, mora da potvrdi
 - ALL – Sve replike moraju da potvrde
 - LOCAL_QUORUM - većina replika u lokalnom data centru moraju da potvrde (u onom gde se nalazi koordinator)
 - EACH_QUORUM - većina replika u svakom datacentru mora da potvrdi
 - LOCAL_ONE – Samo jedna replika iz lokalnog datacentra mora da potvrdi
 - ANY – Bilo koja replika može da potvrdi

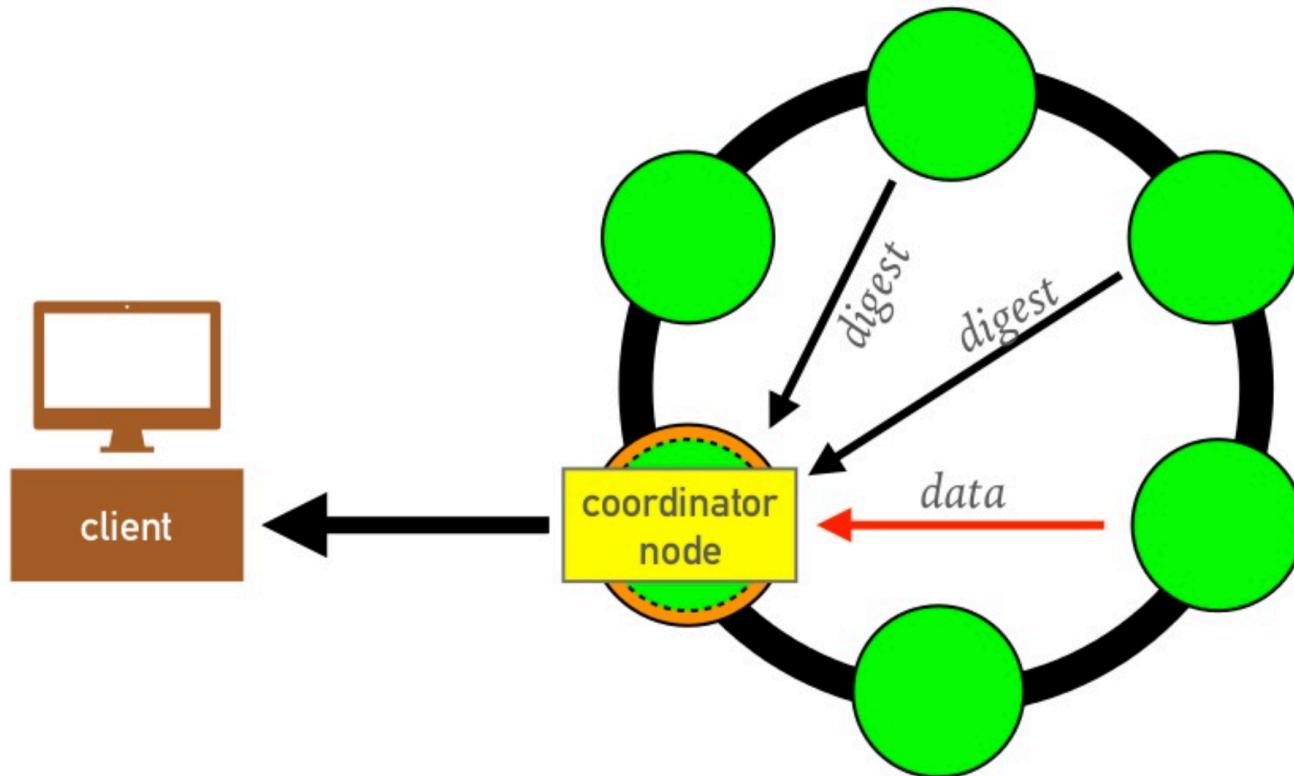
Cassandra - operacije

- WRITE operacija (na nivou čvora)



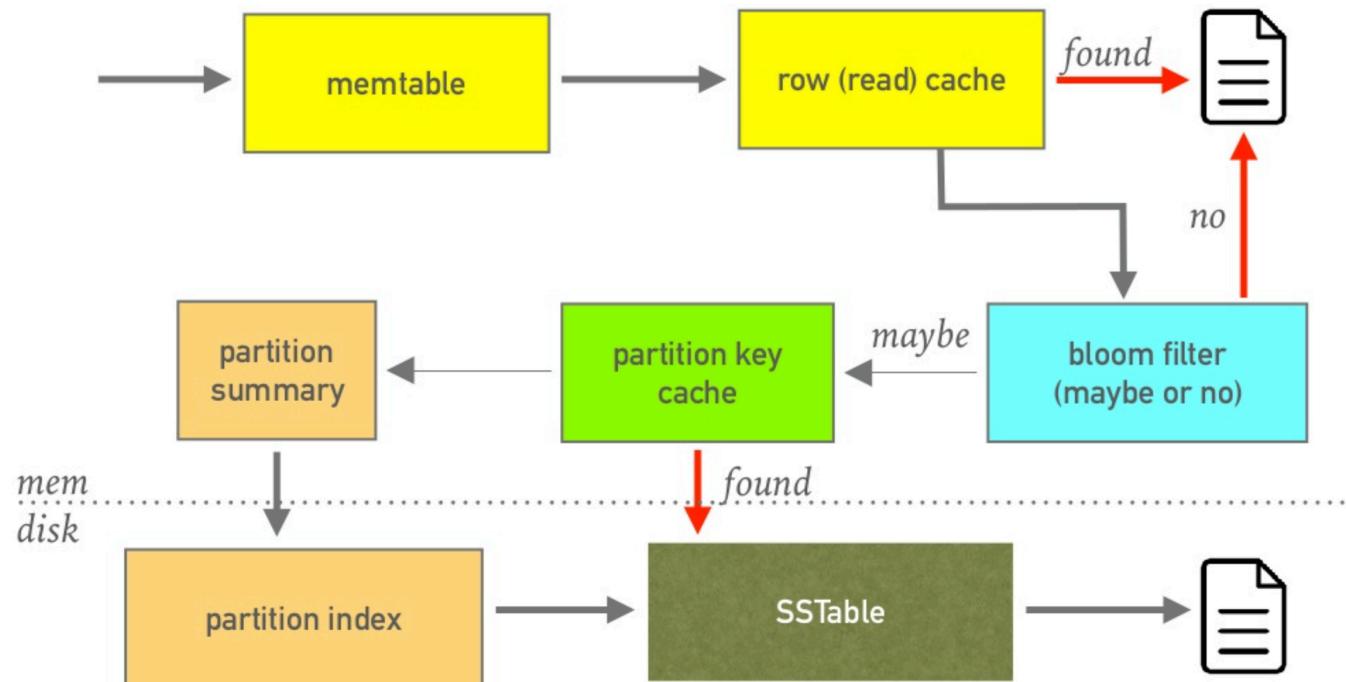
Cassandra - operacije

- READ operacija (na nivou čitavog klastera)
 - Podržani svi nivoi konzistentnosti, izuzetak je ANY
 - Read repair – koristi se hash digest i timestamp

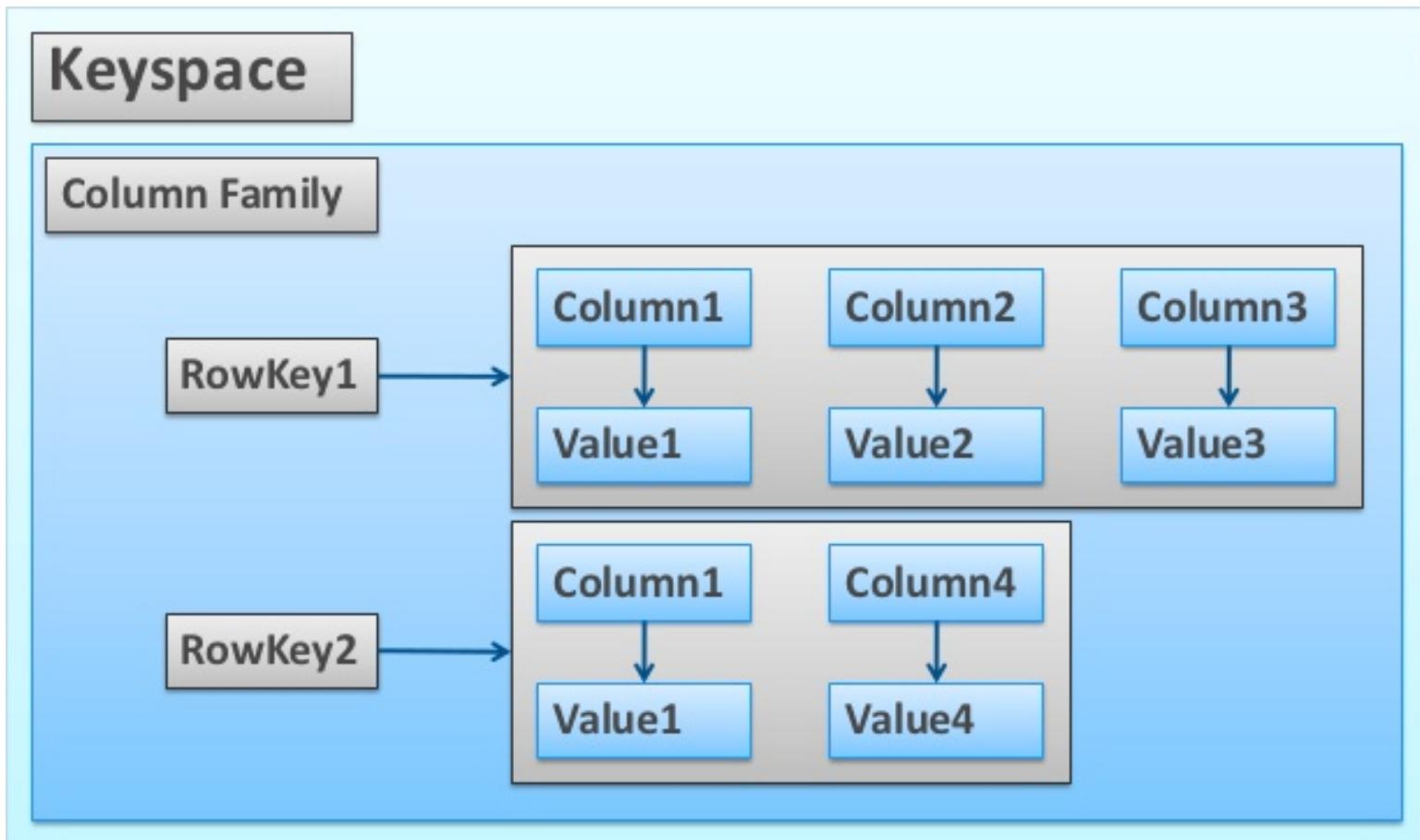


Cassandra - operacije

- READ operacija (na nivou čvora)

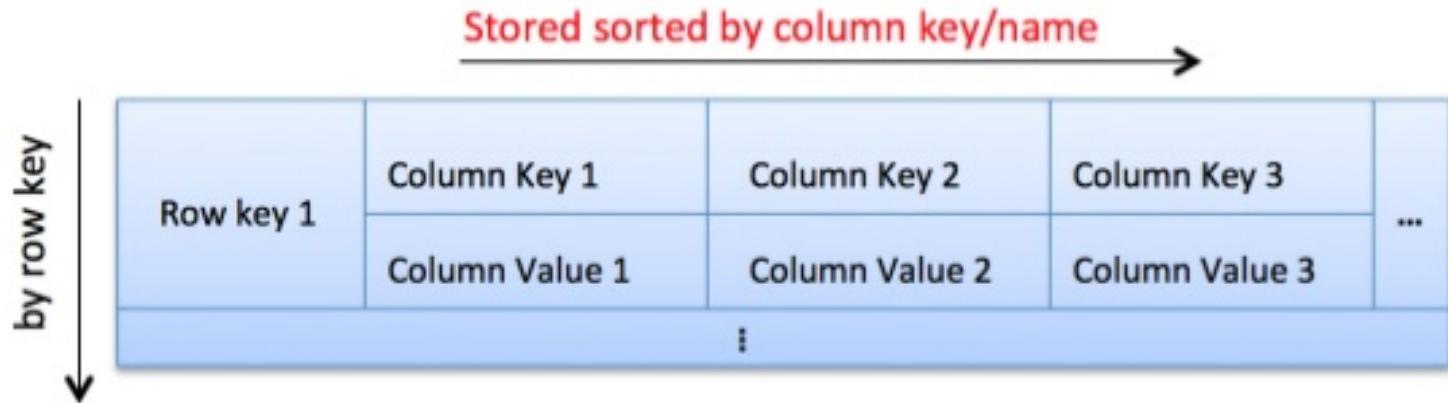


Cassandra model podataka





Cassandra model podataka



Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family (CF)
Primary key	Row key
Column name	Column name/key
Column value	Column value



Cassandra model podataka

- Osnovu modela podataka čini sortirana hash mapa.
- Mapa obezbeđuje efikasne operacije pretage podataka po ključu.
- Sortiranost obezbeđuje efikasno skeniranje opsega.
- Neograničeni broj kolona.
- Ključ može da predstavlja podataka sam po sebi.



Cassandra model podataka

- Pojam keyspace odgovara konceptu relacione baze podataka.
- Keyspace atributi:
 - Faktor replikacije (replication factor) – broj kopija svakog podataka
 - Faktor distribucije replikacija (Replica placement strategy)
 - Column families
- Za potrebe aplikacije moguće je kreirati veći broj keyspace-ova (ukoliko je potrebno imati različite attribute).



Cassandra model podataka

- Column family odgovara konceptu tabele kod relacionih baza podataka.
- Predstavlja kontejner za kolekciju vrsta.
- Svaki podataka se može tretirati kao četvorodimenzionalna hash mapa:

[Keyspace][Column family][Key][Column]

- Column family nije strogo definisana odnosno kolone se mogu dodati bilo kojoj vrsti u bilo kom trenutku.
- Column family sadrži kolone ili super kolone (kolona koja se sastoji od podkolona).



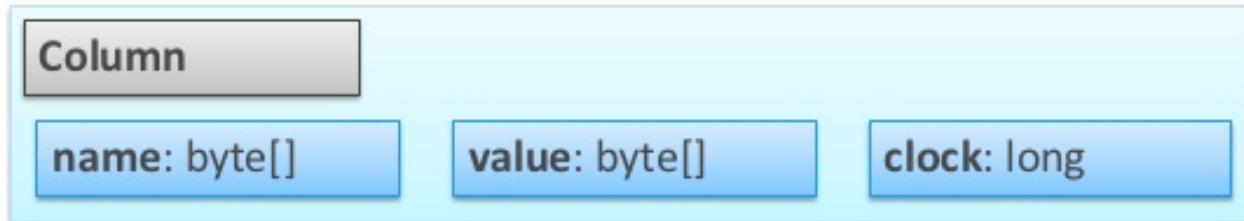
Cassandra model podataka

- Za svaki column family definisan je komparator (comparator) atribut koji specificira kako će kolone biti sortirane u rezultatima upita (long, byte, ASCII, ...)
- Svaka column family se skladišti u zasebnoj datoteci na disku pa je preporučljivo da se povezane kolone skladište u okviru iste column family.



Cassandra model podataka

- Osnovna struktura podataka:



- Maksimalna veličina za ime kolone je 64KB.
- Maksimalna veličina podatka koji se može smestiti u kolonu je 2GB. Preporučeno je da veličina podatka ne treba da prelazi par MB.



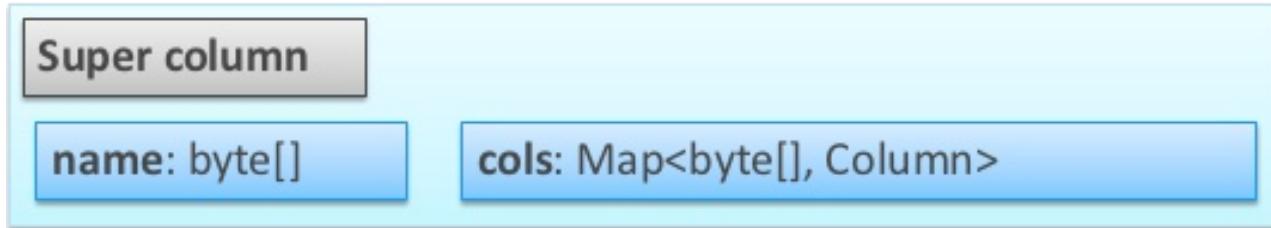
Cassandra model podataka

- Tipovi vrsta:
 - Wide rows – veliki broj različitih kolona i mali broj vrsta (koriste se za čuvanje lista objekata)
 - Skinny rows – mali broj kolona, veliki broj vrsta (bliže relacionom modelu podataka)
- Sortiranje kolona je značajno kod wide rows.
- Cassandra podrazumevano podržava sledeće tipove komparatora:
 - AsciiType
 - ByteType
 - LexicalUUIDType
 - IntegerType
 - LongType
 - TimeUUIDType
 - UTF8Type



Cassandra model podataka

- Superkolona predstavlja mapu podkolona.



- Superkolona ne može da sadrži druge superkolone.
- Petodimenzionalna hash mapa:
[Keyspace][Column family][Key][SuperColumn][Subcolumn]
- U pojedinim situacijama je bolje koristiti kompozitne ključeve.
- **Kod novih verzija Cassandra-e odustaje se od podrške za super kolone (CQL ne podržava superkolone)**



Cassandra model podataka

- Tipovi column families:
 - Standardne (podrazumevno)
 - Sadrži kolone i super kolone
 - Super column family
 - Sadrži samo super kolone
 - Striktnija ograničenja šeme podataka
 - Poseban komparator se može specificirati za svaku podkolonu



Cassandra model podataka

User

123456	Name	Email	Phone	State
	Jay	jay@ebay.com	4080004168	CA
⋮				

Static column family

ItemLikes

123456	121212	343434	...
	iphone	ipad	⋮
⋮			

Dynamic column family
(aka, wide rows)



Cassandra model podataka

User

123456	UserInfo		Likes		
	Name	Email	121212	343434	...
	Jay	jay@ebay.com	iphone	ipad	
:					

Grouping using
Super Column

User

123456	UserInfo Name	UserInfo Email	Likes 121212	Likes 343434	...
	Jay	jay@ebay.com	iphone	ipad	
:					

Grouping using
Composite column
name



Cassandra model podataka

- Korišćenje kompozitnih kolona umesto superkolona (preporučuje se).

Stored sorted by column name - 'subcolumn_one' first, 'subcolumn_two' second,...

Row Key 1	>	
	<Subcolumn_one Subcolumn_two ...> 1	<Subcolumn_one Subcolumn_two ...> 2
	Column Value 1	Column Value 2
	⋮	⋮
		...

e.g., <state/city>: <CA/San Diego>, <CA/San Jose>, <NV/Las Vegas>, <NV/Reno>



Cassandra model podataka

- Korišćenje dela podataka za kreiranje ključa vrste
- Korišćenje Universal Unique ID (UUID)
 - 128bitni broj predstavljen u formi stringa
 - Jednostavno se generiše na strani klijenta
 - Globally Unique ID u Microsoft svetu
- **PRIMARY KEY** = Partition key, Clustering key
 - **Partition key** – identificuje particiju/čvor gde će podaci biti smešteni, koristi se partitioning funkcija koja na osnovu vrednosti partition key-a računa hash vrednost i određuje čvor na kome se podaci smeštaju
 - **Clustering key** – definiše kolone po kojima se podaci sortiraju prilikom skladištenja, time s epostiže efikasna pretraga po ključu vrste

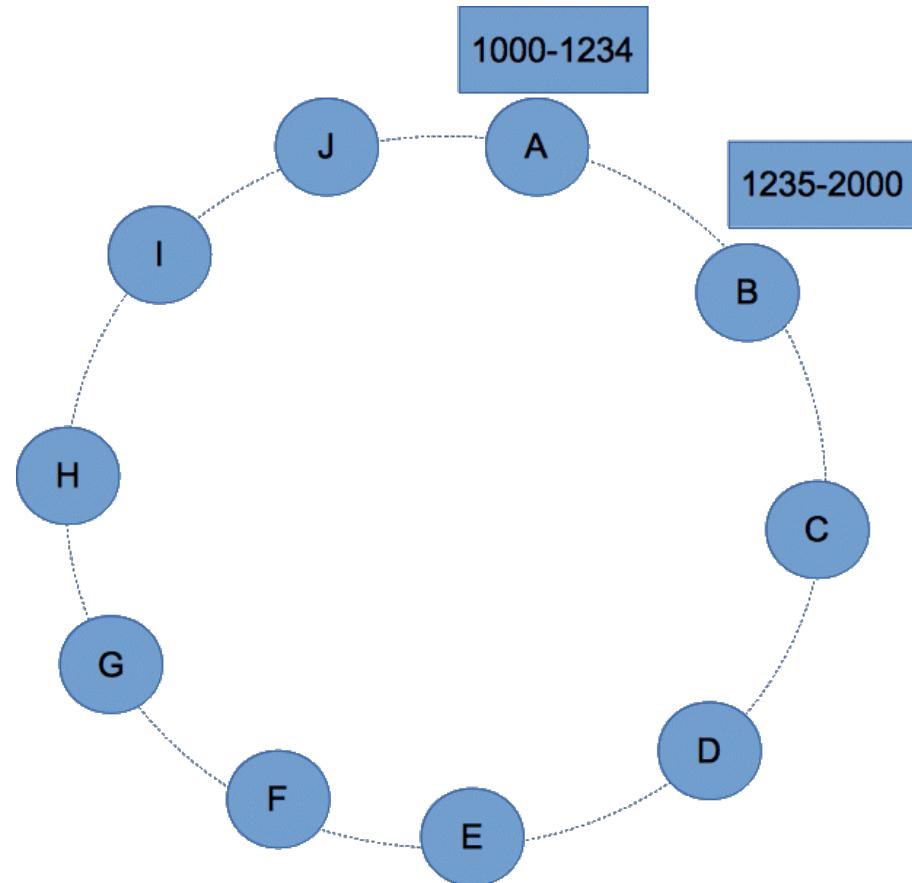


Cassandra model podataka

- Ukoliko su C1, C2, C3, C3, ..., Cn kolone
 - PRIMARY KEY(C1): Primarni ključ sadrži samo partition key
 - PRIMARY KEY(C1, C2): Kolona C1 je partition key, kolona C2 je cluster key.
 - PRIMARY KEY(C1,C2,C3,...): Kolona C1 je partition key, kolone C2, C3, ... predstavljaju cluster key.
 - PRIMARY KEY(C1, (C2, C3,...)): Kolona C1 je partition key, kolone C2, C3, ... predstavljaju cluster key.
 - PRIMARY KEY((C1, C2,...), (C3,C4,...)): Kolone C1, C2 predstavljaju partition key, kolone C3,C4,... predstavljaju cluster key.

Cassandra model podataka

- Partitioning key



Cassandra - primena

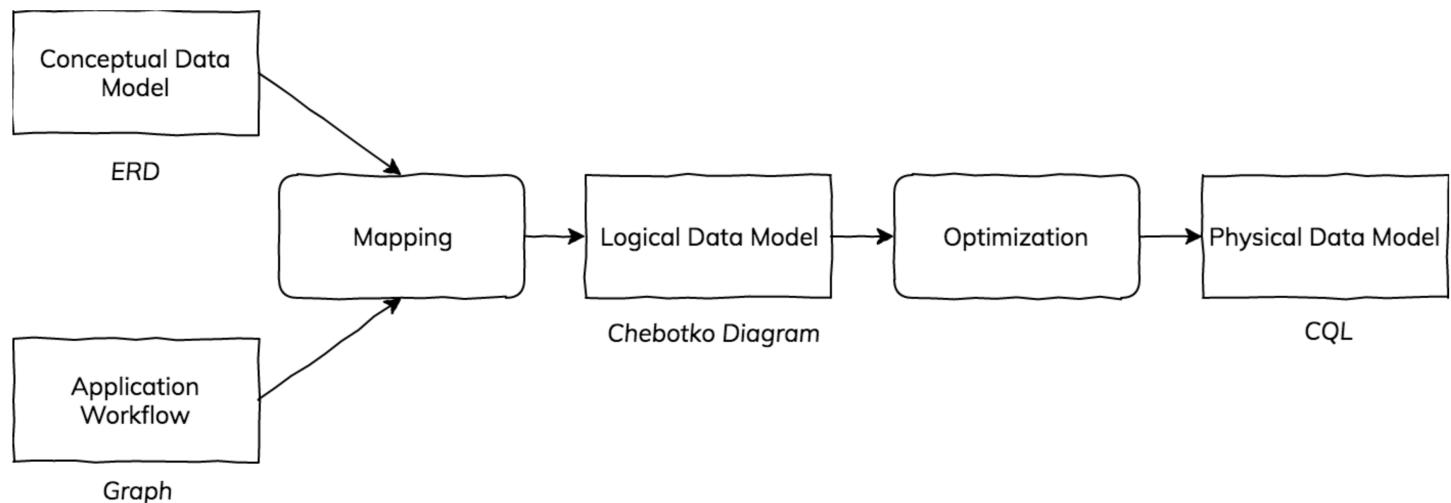
- Katalozi proizvoda
- Playliste
- Recommender/Personalization sistemi (Sistemi za davanje preporuka i personalizaciju)
- Podaci prikupljeni sa senzora (IoT)
- Vremenske serije podataka (time series data)
- Detekcija prevara
- Keširanje podataka

Cassandra - primena

- Kada ne treba koristiti Cassandra kao rešenje:
 - Stroga konzistentnost
 - ACID transakcije
 - Funkcije agregacije
 - Pretraga (skeniranje) podataka bez korišćenja primarnog ključa
 - Veliki broj čitanja podataka
 - Puno ažuriranja i brisanja podataka

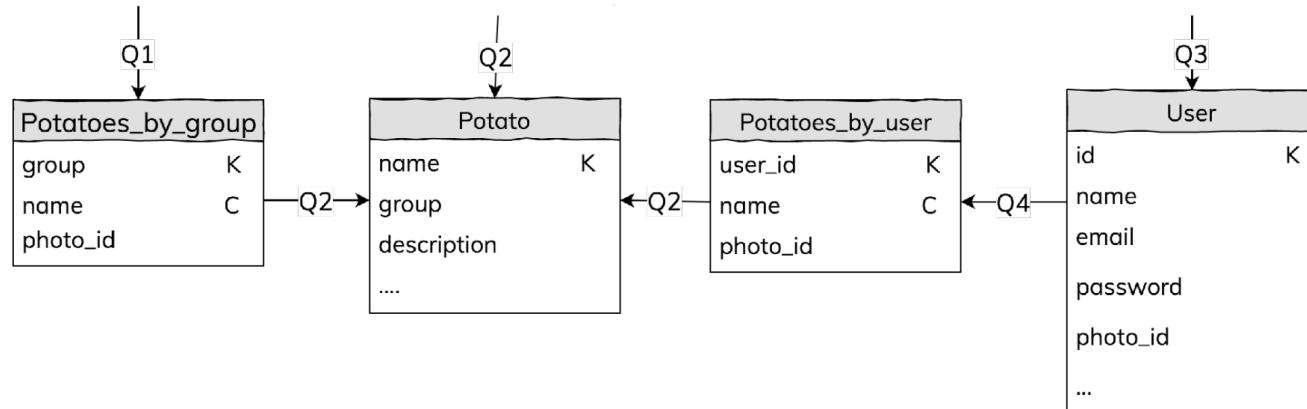
Cassandra primena

- Ne treba krenuti samo od domenskog modela. Moraju se u obzir uzeti i upiti koje će aplikacija koristiti (query driven)



Cassandra primena

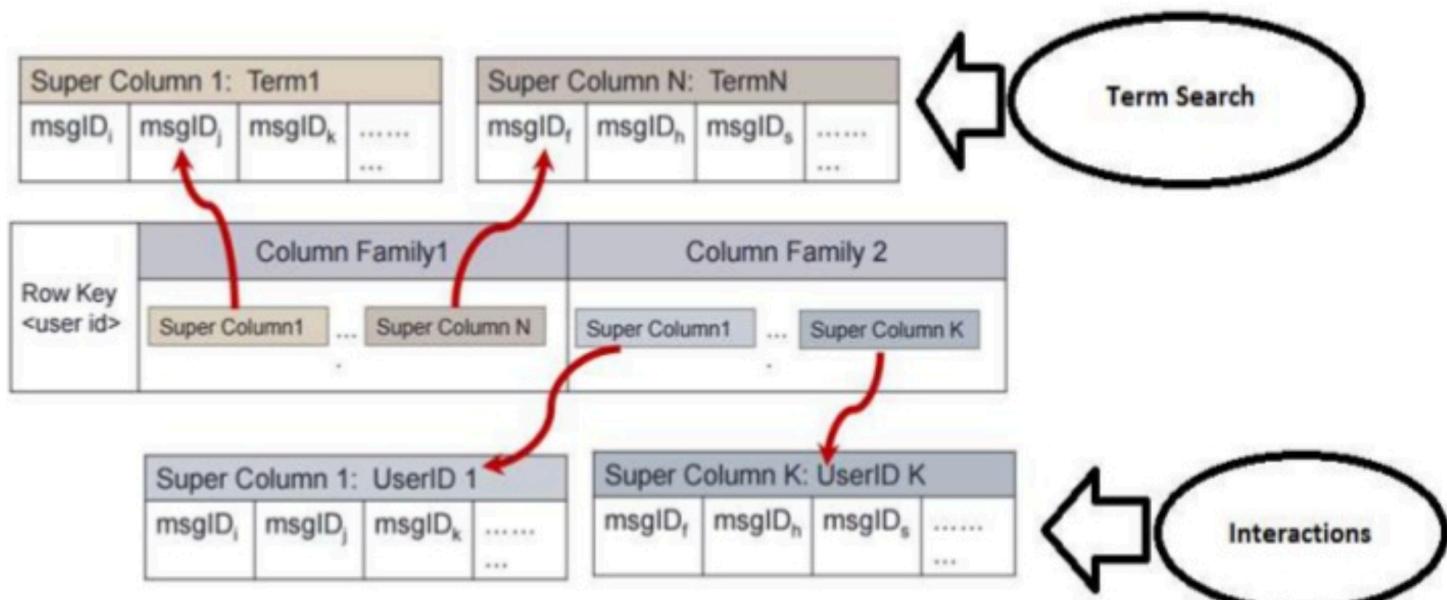
- Chebotko diagram



Q1: Find potatoes by group
Q2: Find details about specific potato type
Q3: Find user details
Q4: Find user's favourite potatoes types

Cassandra primena

- Facebook - Inbox search (namenski razvijeno rešenje, nikada nije iskorišćena)



Cassandra primena

- Instagram
 - Fraud detection
 - Inbox
 - Cassandra zamenila Redis zbog kapaciteta memorije
- Spotify
 - Katalog
 - Online analiza
 - High availability
 - Cassandra zamenila PostgreSQL

Cassandra primena

- Netflix
 - Horizontalna skalabilnost
 - Fleksibilnost šeme
 - Cassandra zamenila Oracle
- Coursera
 - Horizontalna skalabilnost
 - Fleksibilnost šeme
 - Cassandra zamenila MySQL
- Weather channel
 - Vremenske serije
 - Linerana skalabilnost

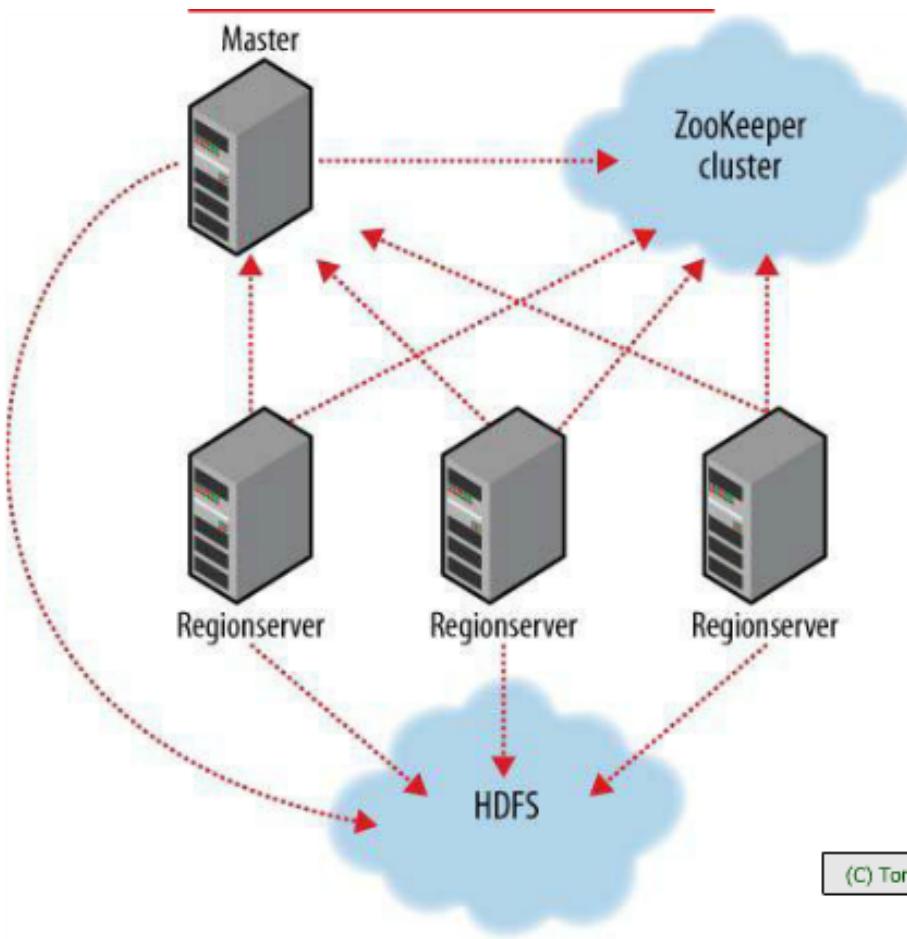
HBase

- Powersoft, Apache
- Bazira se BigTable specifikaciji
 - HDFS (GFS), ZooKeeper (Chubby)
 - MasterNOde (MasterServer), Region Servers (Tablet Servers)
 - Hstore (tablet), memcache (memtable), MapFile (SSTable)

HBase

- Svojstva
 - Podaci se čuvaju sortirani (nema pravih indeksa)
 - Automatsko particionisanje
 - Automatsko re-balansiranje/re-particionisanje
 - Tolerancija na greške/otkaze (HDFS održava tri replike podataka)

HBase - arhitektura



(C) Tom White



HBase - operacije

- Operacije
 - Lokacija HStore/tablet (== BigTable)
 - Klijenti komuniciraju sa ZooKeeper servisom
 - Root metadatata, metadata tabletii, keširanje na strani klijenta
 - Write (== BigTable)
 - Prvo se upisuje u commit log, zatim u memcache (memtable), na kraju u MapFile (SSTable)
 - Read (== BigTable)
 - Najpre se čitaju podaci iz memcache, ako pročitani podaci nisu kompletni obraća se MapFile
 - Kompakcija (== BigTable)



HBase - operacije

- API
 - Thrift
 - Rest API
 - GET / PUT / POST / DELETE
`<table>/<row>/<column>:<qualifier>/<timestamp>`

HBase - MapReduce

- Hbase može da bude ulaz u MapReduce job
- Svaka vrsta predstavlja ulazni slog MapReduce job-a
- MapReduce omogućava da se paralelizuje obavljanje operacija nad većim brojem vrsta.

HBase as a MapReduce input

