Alternativni modeli podatkovnih baz

Podatkovne baze NoSQL

- NoSQL - not only SQL (ang).
- Relacijske podatkovne baze temeljijo na tabelah (relacijah).
- Lahko pa bi izbrali tudi kak drug podatkovni model npr.:
  - slovar
  - graf
  - vrsta
  - ...
- Alternativni modeli imajo ponavadi drugačne (specifične) jezike za opis podatkovnih modelov in uporavljanje s podatki (ni SQL).

Izzivi

- Pri relacijskih podatkovnih bazah smo navajeni na sočasnostni model ACID.
- Če relaksiramo ta model, lahko, ob sprejetju določenih kompromisov, delovanje baz pohitrimo, poenostavimo, ...
- Velike količine podatkov (ang. Big data) zahtevajo shranjevanje na večih strežnikih.
- Polno usklajevanje (npr. zaradi ACID) med njimi je lahko nesprejemljivo zahtevno in počasno.

Porazdeljeni sistemi

- Tipične operacije, ki jih izvajamo na kakršnih koli podatkovnih bazah so: vstavljanje (INSERT), popravljanje (UPDATE), brisanje (DELETE) in branje oz. poizvedovanje (SELECT).
- Porazdeljen sistem je sestavljen in vozlišč, ki hranijo podatke.
- Vozlišča imenujemo tudi particije.
- Uporabnik tipično kontaktira neko vozlišče in na njem izvede operacijo.
- Ko operacijo izvedemo na vozlišču in če je ta uspešno izvedena, dobimo potrditev (prejema operacije, transakcije).
- Npr. pri ACID podatkovnih bazah, ko dobimo potrditev (trans)akcije točno vemo, kaj je zadnje stanje baze.

Lastnosti porazdeljenih sistemov CAP

- **Consistency** - (konsistenca za branje) vsako branje na katerem koli vozlišču sistema vrne zadnjo v sistem shranjeno/sprejeto vrednost.
- **Availability** - (razpoložljivost) vsako branje podatkov na nekem vozlišču vrne odgovor, a ne nujno čisto zadnje stanje podatkov v sistemu (npr. podatki so zapisani v neko vozlišče, uporabnik je dobil potrditev, a do konkretnega vozlišča še niso prišli).
- **Partition tolerance** - (odpornost na odpoved posameznih particij) - sistem lahko še vedno deluje, tudi če v komunikaciji med vozlišči izgubimo poljubno število sporočil za operacije.

Izrek CAP

- Podal ga je Eric Brower v 90-ih letih.
- V osnovi pravi, da za realne podatkovne baze z vsaj dvemi vozlišči ne moremo hkrati doseči vseh treh lastnosti CAP.
- Če želimo C in A, lahko to izvedemo samo, če imamo le eno vozlišče (nimamo P).
- Če želimo C in P in zapišemo nek podatek v nekem vozlišču, morajo biti vsa ostala vozlišča začasno nerazpoložljiva, dokler niso podatki sinhronizirani (izgubimo A)
- Če želimo A in P, potem podatki na vseh vozliščih morda še niso sinhronizirani (izgubimo C)

Lastnosti BASE

- Če opustimo lastnost C (konsistenca), lahko za NoSQL podatkovne baz obravnavamo naslednje lastnoti.
- **BA (Basically Available)** - sistem zagotavlja lastnost A.
- **S (Soft state)** - podatki v sistemu na nekem vozlišču se lahko spremenijo tudi, če ne vstavljamo v sistem na tem vozlišču (npr. pridejo z zakasnitvijo iz drugih vozlišč).
- **E (Eventual consistency)** - če ne vstavljamo podatkov v sistem, se sčasoma vsa vozlišča sinhronizirajo in postane sistem konsistenten.
- Pri NoSQL baza, kjer nimamo ACID tipično stremimo vsaj k BASE.

Tipi NoSQL podatkovnih baz

- NoSQL podatkovne baze so postale popularne predvsem zaradi velikih podatkovij, ki jih je treba deliti čez več strežnikov.
- Ključ-vrednost baze (slovar)
- Dokumentne baze (JSON, gnezdeni slovari in seznami)
- Tabelarične (slovar)
- Grafovske baze (graf)
- Objektne baze (graf)
- Sporočilne vrste (vrsta)
- Bločne verige (vrsta)
- ...

Ključ-vrednost baze

- Osnovna struktura je slovar.
- Ključe brez težav delimo na več strežnikov in hitro dostopamo do njih.
- Ključi so lahko iz linearno-urejene množice, kar nam omogoča intervalne poizvedbe.
- Redis, Riak, CouchDB, Couchbase, ...
- Lahko so zgolj v spominu ali pa se hranijo na diske.
- Različni modeli glede CAP (AP, CP, CA).

Dokumentne baze

- Dokumente si lahko predstavljamo kot JSON, XML ali YAML strukture.
- Dokumentne baze hranijo dokumente z različnimi oblikami organizacije nad njimi (zbirke, tagi, mape, ...)
- Nekakšne nadgradnje ključ-vrednost baz.
- MongoDB, Elasticsearch, Couchbase, CouchDB, ...
- Različni CAP modeli.

Tabelarične baze

- Neke vrste ključ-vrednost baze.
- Ključ določata npr. dva niza (vrstica, stolpec) in/ali časovni žig.
- HBase, Google Big Table, Amazon DynamoDB, ...
- Različni CAP modeli.

Grafovske baze

- Vozlišča in povezave opremljene s podatki
- Poizvedbe po grafu (iskanje v širino in globino)
- Neo4j, InfiniteGraph, ...
- Tipično ne prenašajo dobro delitve na več strežnikov.

Objektne baze

- Hranijo grafe objektov, kot so pri objektno orientiranem programiranju grafi objektov v spominu
- ObjectivityDB, ZopeDB, ...
- Podobno kot grafovske baze ne prenašajo najbolje delitve na več strežnikov.

Sporočilne vrste

- Vrste objektov (sporočil).
- Samo v spominu ali hranjene na disk.
- Replicirane na večih strežnikih.
- Lahko deljene na več vrst na večih strežnikih.
- Kafka, RabbitMQ, ...

Bločne verige

- Posebne baze za hranjenje zgodovine dejstev.
- Kar se shrani, se ne da več spreminjati.
- Osnovna enota je transakcija, transakcije se pakirajo v bloke, ki so v nespremenljivem zaporedju (zagotovljenem s pomočjo kriptografskih metod).
- Shranjevanje poteka s pomočjo konsenza večih vozlišč.
- Tipično se kopija replicira preko vseh vozlišč, a se kot konsistentno verzijo smatra dovolj staro stanje.
- Javna baza ali privatna (z določitvijo pravic dostopa)
- javne verige: Bitcoin, Ethereum, . . . ,
- privatne verige: Hyperledger Fabric, Quorum, . . .

Izbira tipa podatkovne baze

- Če delamo nek (poslovni) sistem in rabimo ACID, je vedno varno vzeti relacijsko podatkovno bazo (Sqlite, PostgreSql, MySql, . . .)
- NoSQL podatkovno bazo uporabimo le v primeru, če točno vemo, zakaj jo rabimo!
- Delo z velikimi podatkovji zahteva veliko strežnikov in je tipično “drag šport”.

Uporabe alternativnih podatkovnih baz

- Socialna omrežja - ogromno sporočil, dokumentov; nujna hitra dosegljivost, replikacija na več strežnikov
- Spletne in mobilne aplikacije z zelo veliko uporabniki ()
- Podatkovna skladišča

O podatkovnih skladiščih

- Računanje agregatov (GROUP BY) je časovno preveč zahtevno (O(n)) za izvajanje realno-časnih analiz.
- Analitika in podatkovno-gnana podjetja
- Agregate se tipično preračunava periodično ali pretočno (sproti)
- **Periodično**: npr. ponoči, potem so na voljo rezultatske tabele (kar vrne GROUP BY).
  - Včasih se uporablja t.i. *OLAP kocke*, ki omogočajo hitro intervalsko filtriranje po agregatih.
  - Uporaba metode Map-Reduce

O podatkovnih skladiščih

- **Pretočno**: agregate lahko računamo tudi sproti, če so združevalne funkcije "aditivne" (t.j. ko pride nov podatek, lahko pravilno popravimo agregat)
  - Gradimo t.i. materializirane poglede (Materialized View)
  - Kafka + KSQL, Pogledi (VIEW) na dokumentnih bazah (CouchDB), ...

Učene na NoSQL baza

- Za učne primere si lahko ogledate `NoSQLZoo` https://nosqlzoo.net/
- Primeri na podatkovnih bazah MongoDB (Javascript) in Neo4j (Cypher) https://nosqlzoo.net/