

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 upravljanje s podatki (ni SQL)

- ▶ 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, pri čemer je polno usklajevanje (npr. zaradi ACID) med njimi 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 izvedemo na vozlišču in če je ta uspešno izvedena, dobimo potrditev.
- ▶ 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 nekem 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 Brewer v 90-ih letih.
- ▶ V osnovi pravi, da za realne podatkovne baze z vsaj dvema vozlišči ne moremo hkrati doseči vseh treh lastnosti.
- ▶ Če želimo C in A, lahko to izvedemo samo, če imamo le eno vozlišče (ni 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)