

01 – CAP teorém a jeho vztah k NoSQL databázím

Distribuované systémy

Hardware je levný, co se může pokazit, tak se nejspíš pokazí. Případný výpadek je brán jako očekávaný stav.

- **Systémy s distribuovaným výpočetním výkonem**
 - o paralelizace = úspora času
- **Systémy s distribuovaným úložištěm**
 - o vyšší redundance, propustnost a dostupnost
 - o vyšší kapacita



Author of CAP theorem is bald so he needs a CAP

NoSQL (Not only SQL) databáze

- nemá předem definované schéma tabulky (x tradiční relační DBMS)
- Tradiční RDBMS vyžadují vertikální škálování
 - o Nákup lepšího HW stojí exponenciálně víc
- moderní aplikace vyžadují škálovat horizontálně
 - o Nákup dalšího serveru roste lineárně s počtem uživatelů
- Vyvinuty pro běžný HW, který je levný

CAP teorém

CAP teorém je věta popisující 3 základní vlastnosti distribuovaných systémů (např. NoSQL databázi). Je ovšem dokázáno, že v jednu chvíli může systém splňovat pouze 2 ze 3 daných vlastností.

Consistency

- každé čtení vrátí aktuální data nebo chybu

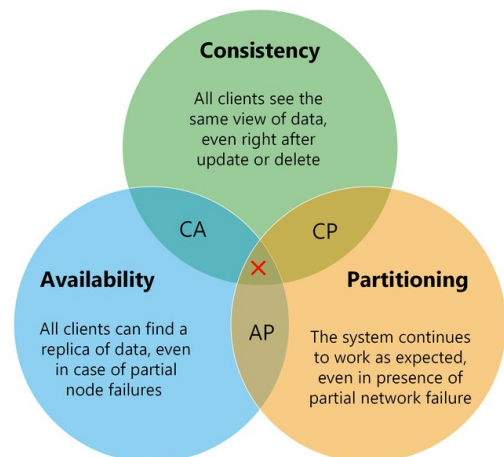
Availability

- na každý požadavek přijde správná odpověď

Partition tolerance

- systém funguje i po výpadku (partition je ve smyslu rozdělení sítě)

Reálně by každý systém měl splňovat Partition tolerance, tudíž jde v podstatě o tradeoff mezi prioritou Availability (PA) a Consistency (PC).



P + C = not available (MongoDB)

- Pokud systém nedokáže vrátit aktuální data tak vrátí chybu.

P + A = not consistent (Cassandra)

- Neexistuje záruka, že čtená data jsou aktuální.
- Tolerantní k výpadku dílčí služby.

C + A = not partition tolerant (RDBMs)

- V případě výpadku části (nebo sítě) je to problém.
- Než aby systém vracel nekonzistentní výsledky, radši neodpoví vůbec.

Segmentovaná konzistence a dostupnost

- Každá komponenta systému splňuje část CAP teorému.
- podle dat (filesystems)
 - o jiná data mohou vyžadovat jinou úroveň dostupnosti a segmentace
- podle operací (databases)
 - o zápis může splňovat něco jiného, než čtení

