

# CAP teorém

**CAP teorém** poprvé formuloval informatik **Eric Brewer** v roce 2000. Teoreticky dokazuje, že distribuovaná databáze (běžící na více uzlech/serverech naráz) nemůže současně zaručit Konzistenci, Dostupnost a Odolnost vůči rozdělení.

Tento princip je naprosto zásadní při návrhu moderních cloudových aplikací a výběru databázových technologií (SQL vs. NoSQL).

## Tři pilíře CAP

Název je akronymem pro tři vlastnosti:

- **C - Consistency (Konzistence):** Každé čtení z libovolného uzlu vrátí nejaktuálnější zápis. Všichni klienti vidí ve stejný okamžik stejná data, bez ohledu na to, ke kterému serveru jsou připojeni.
- **A - Availability (Dostupnost):** Každý požadavek na systém obdrží odpověď (o úspěchu či selhání), i když jsou některé uzly v síti nedostupné. Systém je vždy „v provozu“.
- **P - Partition Tolerance (Odolnost vůči rozdělení):** Systém pokračuje v práci i v případě, že dojde k přerušení komunikace mezi servery (tzv. network partition).

## Proč nelze mít všechno?

V distribuovaném světě je **Partition Tolerance (P)** nutností – sítě nejsou stoprocentně spolehlivé. Pokud nastane rozdělení sítě (servery spolu nemluví), musíme si vybrat:

1. CP (Consistency + Partition Tolerance): Zvolíme konzistenci. Pokud servery nemohou synchronizovat data, systém raději odmítne požadavek (stane se nedostupným), aby předešel zobrazení neaktuálních dat.
2. AP (Availability + Partition Tolerance): Zvolíme dostupnost. Systém odpoví uživateli daty, která má aktuálně k dispozici, i když riskuje, že nejsou nejnovější, protože synchronizace s ostatními uzly selhala.

**Poznámka:** Kombinace **CA** (Konzistence + Dostupnost) je možná pouze v systémech, které nejsou distribuované (běží na jednom stroji), protože tam nemůže nastat rozdělení sítě.

# Praktické příklady a databáze

Výběr mezi těmito vlastnostmi definuje charakter konkrétních databázových systémů:

Typ	Priorita	Příklady	Vhodné pro
<b>CP</b>	Konzistence	MongoDB, Redis, Google Spanner	Bankovní systémy, inventury.
<b>AP</b>	Dostupnost	Cassandra, DynamoDB, CouchDB	Sociální sítě, komentáře, nákupní košíky.
<b>CA</b>	Bez sítě	Tradiční RDBMS (MySQL, PostgreSQL)	Aplikace běžící na jednom serveru.

## Moderní pohled: PACELC

Protože k rozdělení sítě (P) nedochází neustále, byl CAP teorém rozšířen o model **PACELC**. Ten říká: Pokud nastane rozdělení (**P**artition), volíme mezi Dostupností (**A**vailability) a Konzistencí (**C**onsistency). **E**lse (jinak, v běžném stavu), volíme mezi Latencí (**L**atency) a Konzistencí (**C**onsistency).

Tento model lépe vystihuje realitu, kde se často obětuje trocha konzistence pro bleskovou rychlost odezvy (nízkou latenci), i když síť funguje perfektně.

*Související pojmy: Distribuované systémy, Databáze, SQL, NoSQL, Latence, Caching, Cloud Computing.*

From:  
<https://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:  
[https://serviceit.cz/doku.php?id=cap\\_theorem](https://serviceit.cz/doku.php?id=cap_theorem)

Last update: **2026/06/17 19:04**

