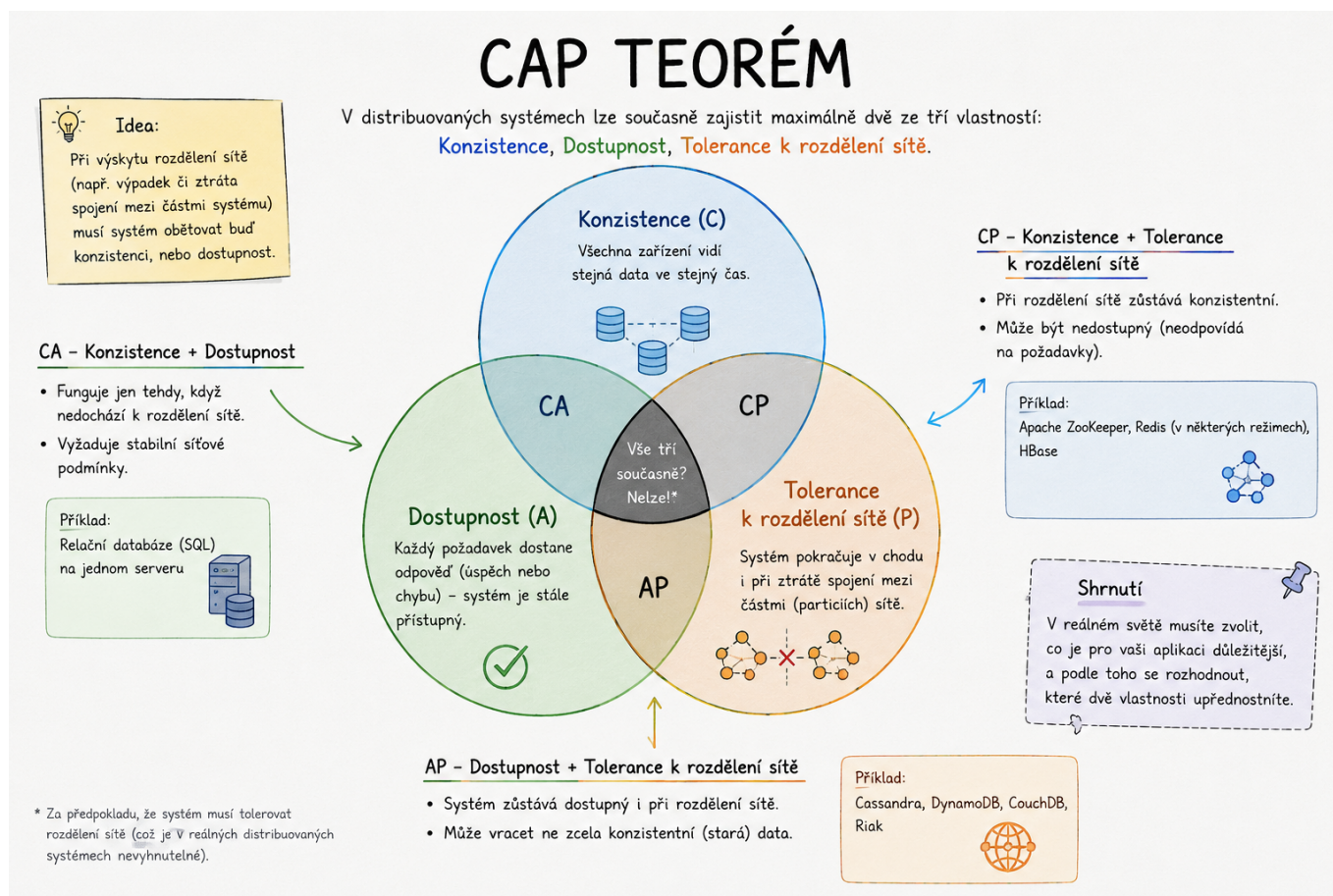


CAP teorém (Brewerův teorém)

CAP teorém, v IT literatuře často označovaný jako Brewerův teorém, je jeden ze základních a nejdůležitějších teoretických konceptů pro návrh distribuovaných počítačových systémů a databází. Autorem této myšlenky je počítačový vědec **Eric Brewer**, který ji poprvé prezentoval v roce 2000. O dva roky později (2002) tento koncept formálně a matematicky dokázali vědci Seth Gilbert a Nancy Lynch z MIT.

Teorém definuje fyzikální a logické limity systémů, které sdílejí data napříč více servery (uzly). Říká, že v jakémkoliv distribuovaném systému je nemožné současně a bezvýhradně garantovat všechny tři následující vlastnosti.



Tři pilíře CAP teorému

Zkratka CAP se skládá ze tří anglických termínů, které představují klíčové vlastnosti distribuovaného systému:

C (Consistency - Konzistence): Každý požadavek na čtení dat vrátí buď informaci o absolutně nejnovějším (posledním) zápisu, nebo vrátí chybu. To znamená, že všichni klienti, bez ohledu na to, ke kterému serveru v clusteru se připojí, vidí v daný okamžik naprosto stejná data. Systém funguje tak, jako by existovala pouze jedna kopie dat.

A (Availability - Dostupnost): Každý požadavek, který dorazí na neselehaný uzel (server) v systému,

obdrží platnou odpověď (bez chyby a bez ignorování požadavku). Avšak tato vlastnost negarantuje, že vrácená odpověď obsahuje nejnovější zapsaná data. Důležité je, že systém „vždy odpoví“.

P (Partition Tolerance – Tolerance rozdělení sítě): Systém nadále funguje jako celek i přesto, že dojde k výpadku sítě nebo ke zpoždění zpráv, což způsobí, že některé servery v clusteru spolu dočasně nemohou komunikovat (sít se rozdělí na izolované „ostrov“).

Mýtus "Vyberte si dvě" a reálná volba

CAP teorém byl historicky často zjednodušován do poučky: „Ze tří vlastností (C, A, P) si můžete pro svůj systém vybrat pouze dvě.“ Tento výklad je však v kontextu moderního internetu zavádějící.

V distribuovaných systémech běžících přes internet (nebo mezi více datovými centry) není tolerance k rozdělení sítě (P) volitelná. Sítě občas selžou (spadne router, překopne se kabel, dojde k DDoS útoku). Protože k výpadkům spojení (Partition) nevyhnutelně dochází, systém se musí v ten moment rozhodnout, co obětuje. Reálná volba architekta tedy zní: Když spadne síť, obětuje Konzistenci (C), nebo Dostupnost (A)?

Rozhodnutí dělí moderní systémy do dvou hlavních táborů (tzv. CP a AP systémy):

CP systémy (Consistency + Partition Tolerance)

Pokud dojde k výpadku sítě mezi uzly, systém raději odmítne odpovědět (obětuje dostupnost), než aby riskoval, že klientovi pošle zastaralá nebo nesprávná data.

Příklad z praxe: Bankovní systémy. Pokud bankomat ztratí spojení s centrálou, raději vám odmítne vydat hotovost (nedostupnost), než aby riskoval, že vám vydá peníze, které už na účtu nemáte (ne Konzistence).

Typické databáze: MongoDB, Redis, HBase.

AP systémy (Availability + Partition Tolerance)

Pokud dojde k výpadku sítě, systém nadále odpovídá na všechny dotazy, i když ví, že nemůže kontaktovat ostatní uzly. Raději klientovi vrátí zastaralá data, než aby nevrátil nic. Jakmile se síť obnoví, uzly si data vymění (tzv. *Eventual Consistency* – konečná konzistence).

Příklad z praxe: Sociální sítě (Facebook, X/Twitter) nebo nákupní košíky na e-shopech (Amazon). Pokud uvidíte počet „lajků“ u příspěvku o několik vteřin opožděně, nic zásadního se neděje. Hlavní je, že se stránka vůbec načte.

Typické databáze: Cassandra, CouchDB, Amazon DynamoDB.

CA systémy (Consistency + Availability)

Systémy, které se nesnaží tolerovat rozpad sítě. Toto v praxi platí pouze pro systémy, které běží na jednom jediném serveru, nebo jsou propojené extrémně spolehlivou lokální sítí (LAN) bez rizika rozdělení.

Typické databáze: Tradiční relační SQL databáze (PostgreSQL, MySQL, Oracle) v single-node nasazení.

PACELC: Moderní rozšíření teorému

Ačkoliv je CAP teorém fundamentální, definuje pouze chování systému v extrémní situaci - při poruše sítě. V roce 2010 proto představil výzkumník Daniel Abadi tzv. PACELC teorém, který původní Brewerovu myšlenku doplňuje.

Zkratka říká: Při výpadku sítě (Partition) si musíte vybrat mezi Availability a Consistency (to je klasický CAP teorém). Else (Jinak - pokud systém běží normálně a síť funguje), musíte si vybrat mezi Latency (Rychlostí odezvy) a Consistency (Konzistencí).

PACELC reflektuje fakt, že i když síť funguje bezchybně, synchronizace dat napříč celým světem stojí čas (latenci). Architekt tedy musí řešit, zda klientovi odpoví okamžitě (ale možná se starými daty), nebo ho nechá čekat, než se všechny uzly bezpečně sesynchronizují.

Související pojmy: [Eric Brewer](#), Distribuované systémy, NoSQL, Eventual Consistency, PACELC, MongoDB, Cassandra, ACID vs BASE.

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

Permanent link:
https://www.serviceit.cz/doku.php?id=breweruv_teorem&rev=1781716340

Last update: **2026/06/17 19:12**

