

# PACELC teorém

PACELC teorém je koncept v počítačové vědě, který rozšiřuje a doplňuje slavný [CAP teorém](#) pro distribuované databázové systémy. Autorem tohoto teorému je počítačový vědec Daniel J. Abadi z Yaleovy univerzity, který jej poprvé definoval v roce 2010 a formálně publikoval v roce 2012.

Zatímco CAP teorém se zabývá pouze tím, jak se systém chová v krizové situaci při výpadku sítě (Partition), PACELC reflektuje skutečnost, že distribuované systémy musí činit architektonické kompromisy i za běžného, bezchybného provozu.

## Rozbor zkratky PACELC

Název teorému je složen ze dvou logických částí, které popisují rozhodovací strom distribuovaného systému:

**PAC** (Při výpadku): Pokud dojde k rozdělení sítě (Partition - P), systém si musí vybrat, zda obětuje dostupnost (Availability - A), nebo konzistenci dat (Consistency - C). Tato první polovina je v podstatě jen zopakováním klasického CAP teorému.

**ELC** (Za normálního provozu): Jinak (Else - E), tedy v případě, že síť funguje naprosto v pořádku, musí systém volit mezi rychlostí odezvy (Latency - L) a absolutní shodou dat (Consistency - C).

Tento dodatek je klíčový. V globálním měřítku existuje fyzikální limit daný rychlostí světla. Pokud máte datové centrum v Evropě a druhé v USA, synchronizace dat vždy zabere desítky milisekund. Architekt se tak musí rozhodnout, zda databáze odpoví uživateli okamžitě (L), ale s rizikem, že si přečte zlomky vteřin stará data, nebo zda uživatele nechá čekat (C), dokud se data stoprocentně nesynchronizují napříč všemi kontinenty.

## Klasifikace NoSQL databází podle PACELC

Díky PACELC teorému lze mnohem přesněji kategorizovat moderní NoSQL databáze podle toho, jaké kompromisy dělají v obou scénářích (výpadek vs. běžný provoz).

**PA/EL systémy:** Tyto databáze jsou navrženy pro maximální rychlost a plynulost. Při výpadku sítě volí dostupnost (PA) a při běžném provozu volí nízkou latenci na úkor okamžité přesnosti dat (EL). Typickými zástupci jsou Amazon DynamoDB, Apache Cassandra nebo Riak. K dosažení shody využívají koncept konečné konzistence (Eventual Consistency).

**PC/EC systémy:** Tyto systémy jsou nekompromisní v otázce přesnosti. Při výpadku sítě se raději odpojí, aby nevrátily špatná data (PC), a za běžného provozu vás nechají čekat na dokončení synchronizace všech uzlů (EC). Patří sem ACID kompatibilní systémy jako Apache HBase, Google Bigtable nebo VoltDB.

**PA/EC systémy:** Zajímavý hybridní přístup. Pokud dojde k havárii sítě, systém se snaží za každou cenu zůstat dostupný pro uživatele (PA). Nicméně pokud síť funguje normálně, systém dává přednost

absolutní shodě dat a nutí uzly k synchronizaci (EC). Do této kategorie spadá běžná konfigurace databáze MongoDB.

PC/EL systémy: Tyto systémy při pádu sítě raději odmítnou komunikovat, aby zabránily rozpadu dat (PC), ale za normálního provozu se snaží odpovídat co nejrychleji i za cenu drobné lokální nekonzistence (EL). Příkladem je databázový systém PNUTS od společnosti Yahoo!.

*Související pojmy: [CAP teorém](#), [Distribuované systémy](#), [Latence](#), [Eventual Consistency](#), [NoSQL](#), [Cassandra](#), [MongoDB](#), [ACID](#), [BASE](#).*

From:

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

Permanent link:

[https://www.serviceit.cz/doku.php?id=pacelc\\_teorem](https://www.serviceit.cz/doku.php?id=pacelc_teorem)

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

