

Koordinace v distribuovaném systému: čas a vzájemné vyloučení

Paralelní a distribuované systémy, přednáška 7

Tomáš Urbanec

Katedra informatiky PřF UPOL

18.11.2025

Co nás čeká?

Koordinace v distribuovaném systému

1. Čas v distribuovaném systému
 - Skutečný čas
 - Logický čas
2. Vzájemné vyloučení v distribuovaném systému
 - Centralizované řešení
 - Distribuované řešení
 - Řešení tokenem
 - Decentralizované řešení

Čas v DS
●○○○○○○○○○

Vzájemné vyloučení v DS
○○○○○○

Příště: Volba leadera
○○○

Čas v DS

Čas a koordinace DS

- Často koordinace na základě času (typicky timestamp).
- Absence globálních hodin.
- UTC (Coordinated Universal Time [sic])
 - radiostanice, satelity,
 - úrovně (vrstva),
 - přesnost ± 10 ms (lze až 50 ns).
- Různý čas na různých uzlech (*skew*).
- Různé důvody: technologie, mimoběžnost, *drift*, ...
- Skutečný čas vs logický čas.
- Omezení: čas nikdy nevracíme zpět.
 - Proč?
 - A co tedy děláme?

Skutečný čas

Základy

- Synchronizace = omezení rozdílu.
- Nelze synchronizovat plně.
- Naivní algoritmus
 - Klient K požaduje čas u serveru S s UTC.
 1. K požádá o čas S .
 2. S odpoví svým aktuálním časem t_s .
 3. K nastaví čas dle t_s .
 - Ale to nefunguje... Proč?

Cristianův algortimus

Skutečný čas

- Klient K požaduje čas od serveru S s UTC.
 1. K požádá o čas S a uloží čas požadavku t_1 .
 2. S odpoví svým aktuálním časem t_s .
 3. K uloží čas přijetí t_2 .
 4. Výsledný čas K : $t_s + (t_2 - t_1)/2$.
- Problém při různém trvání dotazu a odpovědi
 - Vytváření, přenos, ...
- Lze poslat více dotazů a udělat průměr.

Network Time Protocol

Skutečný čas – NTP

- Skutečně a často používané.
- Centrální autorita – UTC.
- Základní myšlenka podobná Cristianovu algoritmu.
- Klient K požaduje čas od serveru S s UTC.
 1. K požádá o čas S v čase t_1
 2. S přijme požadavek v čase t_2
 3. S pošle odpověď s aktuálním časem t_3
 4. K přijme odpověď v čase t_4
 5. Odhad offsetu K vůči S : $\Theta = t_3 + \frac{(t_2 - t_1) + (t_4 - t_3)}{2} - t_4$.
 6. Odhad zpoždění zpráv: $\delta = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$.
 7. K případně upraví rychlost času do srovnání.
- Výpočet Θ a δ se provede vícekrát.
- Použije výsledek s minimální δ .
- Vždy se ptá ten s více skoky od UTC serveru (na vyšší vrstvě).

Reference Broadcast Synchronization

Skutečný čas – RBS

- Distribuované řešení.
- Používané v bezdrátových/senzorových sítích.
- Cíl není skutečný čas (UTC), ale shodnout se na stejném čase.
- Předpoklady
 - Mezičasy generuje až síťové rozhraní – žádné ztráty výpočtem.
 - Předpokládáme stejnou rychlost přenosu mezi všemi páry uzlů.
- Uzel u pošle (broadcast) zprávu m (reference).
 1. Uzly p a q po obdržení m uloží časy přijetí t_p a t_q .
 2. Uzly p a q si vzájemně pošlou zprávu s časem přijetí m (tj t_p a t_q).
 3. Uzly p a q z dodaného času vypočítají offset proti druhému uzlu.
 4. Časem výsledky mohou být neaktuální kvůli driftu → průměry, lin. regrese.
- Uzly pouze ukládají offsety, neupravují svůj čas.

Logické hodiny

- Dosud jsme řešili skutečný čas.
- Stačilo by nám méně?
- Logický čas $\approx$ uspořádání operací.
- Tj. operace a nastala někdy po operaci b .
- Leslie Lamport, 1978

Lamportův algoritmus

Logické hodiny

- Lokální hodiny $\approx$ čítač.
- S každou zprávou se odesílá i aktuální hodnota čítače.
- Čítač se zvedne
 - Při vykonání lokální operace.
 - Před odesláním zprávy.
 - Při přijetí zprávy s vyšší hodnotou čítače.
- Koncept předcházení: a předchází b ($a \rightarrow b$), pokud
 - a a b jsou v jednom procesu a a se stalo před b
 - a je odesláním zprávy a b je přijetím této zprávy
 - tranzitivní uzávěr
- Při souběhu zpráv rozhodujeme dle id procesu.
- Relace $\rightarrow$ je tranzitivní, ale ne nutně úplná.
- (tabule)
- Lze použít pro úplné uspořádání událostí v DS.
- Lze použít pro vzájemné vyloučení.

Vektorové hodiny

Logické hodiny

- Lamportovy hodiny zaručují $a \rightarrow b \Rightarrow C(a) < C(b)$
- Nic ale nezaručuje $C(a) < C(b) \Rightarrow a \rightarrow b$
- Lamportovy hodiny nepopisují kauzalitu
- Vektorové hodiny
 - = vektor $\vec{C}_i$ logických hodin v každém procesu
 - $\vec{C}_i[i]$ je lokální logický čas procesu i
 - $\vec{C}_i[j] = k$ znamená, že proces i ví, že proces j vykonal k operací.
 - Uspořádání
 - $\vec{C}_1 \leq \vec{C}_2$, pokud $\forall i : \vec{C}_1[i] \leq \vec{C}_2[i]$.
 - $\vec{C}_1 < \vec{C}_2$, pokud $\vec{C}_1 \leq \vec{C}_2$ a $\vec{C}_1 \neq \vec{C}_2$.
 - $\vec{C}_i(a) < \vec{C}_j(b)$ znamená $a \rightarrow b$ (ale kauzalita?)
 - Neporovnatelné hodiny znamenají konkurentní operace.
- (tabule)

Skutečný čas skutečně

Malá odbočka

- Hybrid logical clock [1] a HybridTime[2]
 - Kombinace logických hodin a timestampu
 - podobné, ale trochujině zaměřený
 - MongoDB, CockroachDB, atd.
- Google TrueTime [3]
 - Spanner DB
 - Skutečný čas v intervalu
 - Stojí na speciálním HW
 - Datacentra, atp.
- Amazon Time Sync Service apod. [4]
 - Skutečný čas synchronizovaný v řádu mikrosekund
 - ... dokud jste na AWS (či podobné) infrastruktuře

Vzájemné vyloučení v DS

Centralizované řešení

Vzájemné vyloučení

- Velmi jednoduchá myšlenka i provedení.
- Jeden uzel (rozhodčí) rozhoduje o výhradním přístupu ke sdílenému zdroji.
 - Rozhodčí drží frontu žádostí.
 1. Zájemce *i* pošle zprávu *REQUEST*.
 2. Rozhodčí potvrdí *OK*, je-li zdroj volný, jinak dá *i* do fronty.
 3. Proces *i* po dokončení práce pošle *RELEASE*.
 4. Rozhodčí pošle *OK* dalšímu ve frontě, nebo čeká na žádost.
- Rozhodčí může selhat (ostatní nepoznají).
- Rozhodčí může být přetížený (bottleneck).

Distribované řešení

Vzájemné vyloučení

- Ricart-Agrawala algoritmus, 1981
- Inspirován logickými hodinami.
- Potřebuje $2(N - 1)$ zpráv.
- Nejsou nutné FIFO kanály.
- Dva typy zpráv: *REQUEST* a *ACK*
- Proces i chce výhradní přístup:
 1. Zašle *REQUEST* (i, t) ostatním.
 2. Proces j po přijetí odpoví *ACK* (nechce přístup nebo $t < C_j$), nebo přidá i do fronty žádostí.
 3. Pokud čekající i dostane $(N - 1)$ *ACK*, získává přístup.
 4. Při ukončení práce pošle i *ACK* všem ve své frontě.
- Potřebuje multicast.
- Má N bodů selhání (opakované dotazy).

Řešení tokenem

Vzájemné vyloučení

- Vytvoříme kruhový overlay nad sítí.
- V sítí je právě jeden token.
- Výhradní práce se zdrojem = držení tokenu.
 1. Proces i obdrží token.
 2. Chce-li pracovat, pracuje. Jinak odesílá token dále.
 3. Po ukončení práce pošle i token dále (nesmí jej držet).
- Ztráta tokenu? Informace o ztrátě tokenu?
- Udržování kruhu (někdo vypadne $\rightarrow$ přeskočit? potvrzování?).

Decentralizované řešení

Vzájemné vyloučení

- Založeno na hlasování.
- Každý zdroj je v N replikách.
- Každá replika má koordinátora.
- Výhradní práce se zdrojem = hlas více než $N/2$ koordinátorů.
- Koordinátoři mohou selhat (resetovat se)
 - Vydání více povolení?
 - Vydání více povolení u více než $N/2$ koordinátorů?
 - Možné, ale extrémně nepravděpodobné.
 - Ve srovnání s jinými typy chyb zanedbatelné.

Srovnání přístupů

Vzájemné vyloučení

- Nejjednodušší je centralizovaný přístup.
- Př. ZooKeeper – koordinační systém
 - vzájemné vyloučení,
 - volba leadera.
 - monitoring,
 - zotavení z chyb.
 - ...
- token vs. vyžádání přístupu
- centralizované, decentralizované nebo distribuované

Příště: Volba leadera

Základy

Volba leadera

- Leader = koordinátor, vstupní uzel DS, ... významný uzel.
- Úkol?
 - komunikace s uživatelem,
 - koordinace,
 - vzájemné vyloučení,
 - topologie,
 - ...
- Problémy?
 - Všichni se musí shodnout na leaderovi
 - Kdokoliv může chtít být leaderem
 - Leader může vypadnout
 - Síť se může rozpadnout
 - Další požadavky (výkon, poloha, ...)
- Mnoho přístupů
 - Dle typu sítě
 - Dle topologie (případně overlay)
 - Často část složitějších algoritmů



Logical Physical Clocks and Consistent Snapshots in Globally Distributed Databases,

<https://cse.buffalo.edu/tech-reports/2014-04.pdf>.
Citováno 18.11.2025.



Technical Report: HybridTime - Accessible Global Consistency with High Clock Uncertainty,

<https://users.ece.utexas.edu/~garg/pdslab/david/hybrid-time-tech-report-01.pdf>. Citováno 18.11.2025.



Spanner: TrueTime and external consistency,

<https://cloud.google.com/spanner/docs/true-time-external-consistency>. Citováno 18.11.2025.



Amazon Time Sync service, [https:](https://aws.amazon.com/about-aws/whats-new/2023/11/amazon-time-sync-service-microsecond-accurate-time/)

[//aws.amazon.com/about-aws/whats-new/2023/11/amazon-time-sync-service-microsecond-accurate-time/](https://aws.amazon.com/about-aws/whats-new/2023/11/amazon-time-sync-service-microsecond-accurate-time/).
Citováno 18.11.2025.

Čas v DS
oooooooooooo

Vzájemné vyloučení v DS
oooooo

Příště: Volba leadera
oo●

Changelog