

Volba leadera, chyby v DS a chord systém

Paralelní a distribuované systémy, přednáška 8

Tomáš Urbanec

Katedra informatiky PŘF UPOL

25.11.2025

Co nás čeká?

1. Volba leadera

- Algoritmy Bully a Ring
- Raft: Volba leadera
- Ad-hoc síť
- Další možnosti

2. Chyby v DS

- Chyby a jejich klasifikace
- Byzantské chyby
- Tolerance chyb a redundance

3. Chord systém

≈ distribuovaná hashovací tabulka

Volba leadera

Základy

Volba leadera

- Leader = koordinátor, vstupní uzel DS, ... obecně významný uzel.
- Úkol?
 - komunikace s uživatelem,
 - koordinace,
 - vzájemné vyloučení,
 - topologie,
 - ...
- Všechny uzly se musí shodnout, kdo je leader.
- V různých okamžicích mohou být různí leaderi.

Základy

Volba leadera

- Souvisí s dalšími problémy
 - Vzájemné vyloučení
 - Obecná shoda v DS
- Předpoklady
 - Uzly mají ID (preferujeme větší).
 - Všichni o všech ví.
 - Uzly mohou vypadnout.
 - Běžný uzel (následovník)?
 - Leader?

Bully algoritmus

Volba leadera

- „Nejsilnější vyhrává.“
 - Garcia-Molina, 1982
 - Kandidující uzly:
 - ty, které zjistí, že leader nereaguje.
 - nově přidané (zotavené).
 - ty, které převzaly kandidaturu (níže).
- zpráva ELECTION(id) všem s vyšším ID.
- Uzel obdrží zprávu ELECTION(*cid*) a
 - pokud je jeho $id > cid$, odpoví OK a sám se stává kandidátem.
 - Pokud uzel obdrží zprávu OK přestává kandidovat.
 - Nakonec to vzdají všichni kromě jednoho (nejvyšší id)-
 - Ten všem oznámí, že je novým leaderem. Jak to pozná?
 - V nejhorším případě $O(n^2)$ zpráv.
 - (tabule)

Ring algoritmus

Volba leadera

- Logický kruh (overlay).
- Všichni znají následovníka (i další uzly).
- Volby zahajuje ten, kdo zjistí, že leader nereaguje (i více).
- Nakonec kandidují všichni běžící.
- Pošle zprávu ELECTION se svým *id* následovníkovi.
- Pokud následovník nefunguje, přeskočí ho (opakování).
- Uzel obdrží zprávu ELECTION:
 - neobsahuje jeho *id* → přidá své *id* do seznamu a pošle dále
 - obsahuje jeho *id* → zpráva LEADER se stejným seznamem
- Zpráva LEADER všem oznámí, kdo je leader (max *id*).
- Původně odesílající uzel ji nakonec zahodí.
- Vždy $O(n)$ zpráv.
- (tabule)

Raft

Základy

- Raft (Ongaro, Ousterhout, 2013)
- Algoritmus pro shodu v DS s možnými chybami (příště).
- Široce používaný.
- Základ dalších algoritmů.
- ≈ Distribuovaný log aplikovaných operací.
 - Stejné operace ve stejném pořadí všude.
 - Stejný stav všude.
 - Dnes jen část (volba leadera).
 - Více uvidíme příště.

Raft

Volba leadera

- Stavy uzlů:
 - Leader
 - Následovník
 - Kandidát
- Většinové kvórum (!)
- *Term* – Volební období.
- *Timeout* – doba po kterou následovník uznává leadera.
- *Heartbeat* – leader se pravidelně hlasí (obnovuje timeout).
- *Split-vote* – nikdo nezískal kvórum (restart, náhodné zpoždění)
- Na počátku jen následovníci.
- Při vypršení timeoutu následovník vyvolá volby.

Raft

Volba leadera

- Vyvolání voleb – následovník
 - zvýší svůj term,
 - stane se kandidátem,
 - hlasuje sám pro sebe,
 - požádá o hlas všechny ostatní (zpráva).
- Uzel obdrží žádost o hlas
 - Pokud ještě nehlasoval, potvrdí hlas.
 - Jinak ignoruje.
- Výsledek
 - Vyhraje volby
 - kandidát dostane většinu hlasů,
 - prohlásí se leaderem.
 - Jiný kandidát se prohlásí leaderem (zpráva) a
 - má term větší nebo roven mému → návrat k následovnictví.
 - má term menší než můj → odmítnu ho.
 - Dlouho nevyhrává nikdo – timeout → restart voleb.
- (tabule)

Další úvahy

Volba leadera

- Co když leader má mít nějaké další vlastnosti?
 - Vhodnou polohu/latenci
 - Vyšší výkon
 - Dostatek zdrojů
 - ...
- Co když se síť často mění (ad-hoc, mobilní, ...)?
- Co když síť není spolehlivá?

Algoritmus pro ad-hoc síť

Volba leadera

- Vasudevan et al., 2004. Řeší úvahy výše.
- Kandidáti jsou nakonec všichni.
- Vybrán ten, který nejlépe plní požadavky.
- Uzel, který chce volby, pošle ELECTION svým sousedům.
- Uzel, který obdrží ELECTION
 - Je to poprvé?
 - nastaví odesílatele jako rodiče,
 - přepoše všem ostatním,
 - čeká na všechna potvrzení, poté sám potvrdí.
 - Už má rodiče? → jen potvrdí přijetí
- Vzniká strom.
- Všichni okolo mají rodiče?
 - Jsem list.
 - Rychle dostanu odpovědi.
 - Rychle odpovídám rodiči.
- V potvrzení rodiči nejlepší leader z podstromu.
- (tabule)

Poznámky

Volba leadera

- Split brain problem (nejen u volby leadera).
- Rozdělení sítě → v každé části volby → ?
 - (ukázka) Bully algoritmus to nezvládne.
 - (ukázka) Raft to zvládne.

→ nutná většina hlasů.

- Uvidíme ještě u shody v DS.

Další možnosti

Volba leadera

- Zookeeper
- Chang-Roberts (upravený Ring)
- Gossip protokoly
- Velké sítě
 - Proof of work
 - Proof of stake

Chyby v distribuovaném systému

Chyby v DS

Základy

- Chyba $\approx$ selhání:
 - uzlu,
 - uzlů,
 - kanalů.
- Selhání?
 - výpadek,
 - špatný výsledek,
 - ...

$\approx$ jiné, než zamýšlené chování.

Chyby v DS

Základy

- Cíl: systém toleruje chyby (v omezeném množství)
 - Toleruje $\approx$ pracuje dle očekávání
 - Detekování chyby
 - Asynchronní DS – nelze (proč?)
 - Synchronní DS – existuje?
- částečně synchronní systém (timeouty)

Chyby v DS

Základy

- Základní vlastnosti DS
 - Dostupnost – připravenost k práci v daném okamžiku
 - Spolehlivost – schopnost běžet nepřetržitě v daném intervalu
 - Bezpečnost – chyba nezpůsobí katastrofu
 - Udržovatelnost – snadnost opravy v případě chyby
- Vysoká dostupnost vs. vysoká spolehlivost?
- Různé metriky
 - MTTF – Mean time to failure
 - MTTR – Mean time to repair
 - MTBF – Mean time between failures
- Pozn: vše to vyžaduje přesně definovat *chybu*.

Klasifikace a projevy chyb

Chyby v DS

- Podle trvání
 - Přejchodná (transient) – objeví se jednou a zmizí
 - Přerušovaná (intermittent) – chyba se objevue a mizí
 - Trvalá (permanent) – chyba zůstává do vyřešení
- Podle projevu
 - Pád – uzel vypadne, doté doby funguje ok
 - Vynechání – selhání poslání/přijetí zprávy
 - Časování – odpovídá pozdě
 - Chyba odpovědi – odpovídá špatně
 - Náhodná (Byzantská) chyba – odpovídá náhodně v náhodném čase

Byzantské chyby

Chyby v DS

- Byzantská říše (395-1453), převážně balkán/blízký východ
- Chyba uzlu, kterou ostatní nepoznají.
- Různé chování k různým uzlům.
- Problém byzantských generálů
 - Koordinovaný útok/ústup.
 - Jinak katastrofa.
 - Zrádci mohou posílat falešné (a různé) zprávy.
 - ... a ty se mohou ztrácet.
- Interpretace
 - zakeřný bug?
 - útok?
- Reálný problém.
- Systém odolá byzantské chybě pokud:
 - Pošle-li poctivý uzel hodnotu x , systém se shodne na x .
 - Všechny poctivé uzly se shodnu na stejné hodnotě x .

Modely detekce chyb

Chyby v DS

- Fail-stop
 - spolehlivě detekovatelné
 - známe stav uzlu
- Fail-noisy
 - eventuálně spolehlivě detekovatelné
 - chyba je zjevná
- Fail-silent
 - nelze rozlišit pád a vynechání
 - neznáme stav uzlu
- Fail-safe
 - o chybě nevíme nic, ale chyby nezpůsobí škodu
 - systém se chybou přepne do fail-safe módu
- Fail-arbitrary
 - o chybě nevíme nic
 - nelze ji ani spolehlivě detekovat

Redundance

Chyby v DS

- Redundance v systému – základní nástroj umožňující toleranci chyb
 - informační redundance
 - časová redundance
 - fyzická redundance
- Příklad TMR (Triple Modular Redundancy)
- Skupiny uzlů (dále)
- k -tolerance – systém přežije k chybných uzlů

Chord systém

Chord systém

≈ Distribuovaná hashovací tabulka

- Klíč-hodnota (klíč-server obsahující hodnotu)
- Peer-to-peer, ring
- Máme m bitové klíče (2^m klíčů).
- Uzly mají unikátní id z téhož prostoru.
- Záznam s klíčem k je uložen na uzlu s nejmenším id větším než k ($succ(k) = id$).
- Úkol: na dotaz k najít v systému $Succ(k)$ (uzel s hodnotou k)
- Triviálně uzel id zná následovníka ($Succ(id + 1)$)
 - Pak pokud hodnotu neznám já, posílám dále.
 - Ale to je pomalé ($O(n)$)

Chord systém

- Lze řešit rychleji – přidáme zkratky
- Uzel má tzv. *Finger Table* (FT_{id}) – vyhledávací tabulka (velikost omezená m)
- $FT_{id}[i] = Succ(id + 2^{i-1})$
- Dotaz na k musíme přeposlat někomu před $Succ(k)$ ale co nejblíže:
 - $FT_{id}[j] \leq k \leq FT_{id}[j + 1]$
 - Modulární aritmetika pro udržení v kruhu.
- Mají-li FT_{id} jen jeden záznam $\rightarrow$ triviální řešení.
- (tabule)

Changelog