Tolerance chyb a shoda v DS

Paralelní a distribuované systémy, přednáška 9

Tomáš Urbanec

Katedra informatiky PřF UPOL

27.11.2024

Co nás čeká?

1. Chyby v DS
   - Chyby a jejich klasifikace
   - Byzantské chyby
   - Tolerance chyb a redundance
2. Shoda v DS s chybami
   - Základní pohled
   - Raft
   - Paxos
   - Byzantské chyby podruhé

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?

  $\rightarrow$ 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
  - UdržovatelnsotUdržovatelnsot – 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
  - Přechodná (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
- Fail-noisy – eventuálně spolehlivě detekovatelné
- Fail-silent – nelze rozlišit pád a vynechání
- Fail-safe – o chybě nevíme nic, ale chyby jsou neškodné
- Fail-arbitrary – o chybě nevíme nic

Redundance

Chyby v DS

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

Shoda v distribuovaném systému

Shoda v DS

Základy

- Systém se musí shodnout na výsledku, stavu, další akci, ...
- Bez chyb $\rightarrow$ triviální.
- S chybami? Záleží na okolnostech.
  - Typy chyb,
  - Model detekce,
  - ...

Skupiny

Shoda v DS

- Zavedení redundance.

→ Úlohu uzlu převezme skupina uzlů.

  - Protokol primární-záloha
    - Hierarchická skupina
    - Pracuje primární (zápisy)
    - Výpadek primárního → záloha převezeme roli
  - Protokol replikovaného zápisu
    - 'Plochá' skupina,
    - Všichni stejná role,
    - Nemají kritický bod,
    - . . . ale nutná koordinace.

- Skupina je odolná vůči chybám, pokud všechny bezchybné procesy vykonávají stejné operace ve stejném pořadí.

Algoritmus Flooding consensus

Shoda v DS

- Základní algoritmus pro shodu.
- Omezení na fail-stop model.
- Shoda se hledá v navazujících kolech.
  - V každém kole si uzly mění návrhy.
  - Každý uzel z návrhu vybere volbu deterministicky a všichni stejně
  - Pokud někdo nedostane některou odpověď, začne nové kolo.
  - Všichni dostanou všechny odpovědi $\rightarrow$ volba $\rightarrow$ konec.
  - (tabule)
- Existuje i uniform flooding consenzus (rozdíly)

Byzantské chyby podruhé

Shoda v DS

- Co s byzantskými chybami?
- Idea problému
  - n-generálu se musí domluvit
  - m z nich je zrádných
  - Co s tím?
- Základní řešení
  - (tabule 2, 3, 4, . . . generálové)
  - $n > 3 \cdot m$
  - Exponenciální počet zpráv
- Jiná řešení
  - Mají další předpoklady a omezení.
  - HoneyBadgerBFT (dig. podpisy) $\rightarrow n > 2 \cdot m$
  - Blockchain (cena) $\rightarrow n > 2 \cdot m$
  - MinBFT (speciální HW) $\rightarrow n > m + 1$
  - Kryptografické protokoly (důvěra) $\rightarrow n > m + 1$

Raft algoritmus

Shoda v DS

- Ongaro, Osterhout, 2014
- Fail-noisy model.
- Shoda v DS ve formě replikovaného logu operací.

$\rightarrow$ uzly ve stejném stavu.

- Moderní, často používaný.
- 'Náhrada Paxosu' (dále).
- Skupina má lídra.
- Stavy uzlů: Lídr, následovník, kandidát.
- Koncepty: volba lídra, replikace logu, provedení operace.
- Většinové kvórum ($n > 2 \cdot m$).
- (tabule)
- Vizualizace: https://raft.github.io

Paxos algoritmus

Shoda v DS

- Lamport, 1989. Mnoho variant.
- Fail-noisy model.
- Často používaný, ale komplikovaný ($\rightarrow$ raft).
- Více typů procesů (client, proposer, leader, learner, acceptor).
- Vnitřní dělení uzlů (proposer + acceptor + learner).
- Většinové kvórum ($n > 2 \cdot m$).
- Předpoklady:
  - I asynchronní DS.
  - I nespolehlivé spoje.
  - Chybné zprávy detekvatelné.
  - Operace deterministické.
  - Nemáme náhodné chyby.
- (tabule - zjednodušená verze)

Changelog