

PDV 11 2020/2021

Konsensus

Michal Jakob

michal.jakob@fel.cvut.cz

Centrum umělé inteligence, katedra počítačů, FEL ČVUT



Co mají tyto příklady společného?

Skupina procesů usilujících o :

- udržování jejich lokálních seznamů aktivních procesů aktuálních [detekce selhání]
- zvolení lídra a zajištění, že každý ví, kdo je lídrem [volba lídra]
- zajistit vzájemně exkluzivního přístupu ke sdílenému prostředku (např. souboru) [vyloučení procesů]
- usilujících o doručení stejných aktualizací ve stejném pořadí [uspořádaný multicast]

Souvisí s konsensem

Ve všech těchto případech se procesy snaží vzájemně koordinovat, aby se shodly na nějaké hodnotě

- na stavu každého procesu (aktivní/neaktivní)
- kdo je lídrem
- kdo má přístup k sdílenému prostředku
- pořadí zpráv

Všechny tyto problémy souvisejí s problémem **konsensu**

Problém konsensu

N procesů

Každý proces P má

- vstupní proměnou x_p (výchozí návrh): zpočátku buď 0 nebo 1
- výstupní proměnou y_p : může být změněna pouze jednou

Cílem je **shodnout na hodnotě** výstupní proměnné.

Algoritmu pro řešení konsensu

Algoritmus pro řešení konsensu musí splňovat následující vlastnosti:

- **Shoda:** všechny *nehavarované* procesy se shodnou na výstupní proměnné
- **Validita:** mají-li všechny *nehavarované* procesy stejnou hodnotu vstupní proměnné, musí se na této hodnotě shodnout i jako na hodnotě výstupní proměnné
- **Ukončení:** každý *nehavarovaný* proces musí v konečném čase nastavit hodnotu výstupní proměnné

Proč je konsensus důležitý?

Mnoho problémů v DS je **ekvivalentních** konsensu

- perfektní detekce selhání
- volba lídra
- vyloučení procesů
- spolehlivý nebo totálně uspořádaný multicast
- ...

Vyřešení konsensu by tedy bylo velmi užitečné.

Konsensus v systémech bez selhání

Problém konsensu v systémech **bez selhání je řešitelný**.
(v synchronním i asynchronním případě).

1. Jednotlivé procesy si vzájemně pošlou hodnoty svých vstupních proměnných.
2. Následně provedou stejný výpočet nad posbíranými hodnotami (až budou mít všechny): např. max, min, majority apod.
3. Výstupní proměnnou nastaví na výsledek výpočtu.

Konsensus v synchronním systému se selháními

Problém konsensu v **synchronních** systémech se selháními je řešitelný.

Uvažujeme maximálně f selhání procesů

Algoritmus pro proces P_i

$z_i \leftarrow x_i$ // vstupní proměnná

for *round* **from** 1 **to** $f + 1$

if aktuální hodnota z_i **nebyla** **rozeslána**

 broadcast(z_i)

$t_{j,i} \leftarrow$ hodnota obdržená z procesu j v tomto kole (pokud nějaká byla)

$z_i \leftarrow \min_j(x, t_{j,i})$

$y_i \leftarrow z_i$ // nastav výstupní proměnnou

Konsensus v asynchronních systémech se selháními

Problém konsensu v asynchronních systémech **se selháními není řešitelný.**

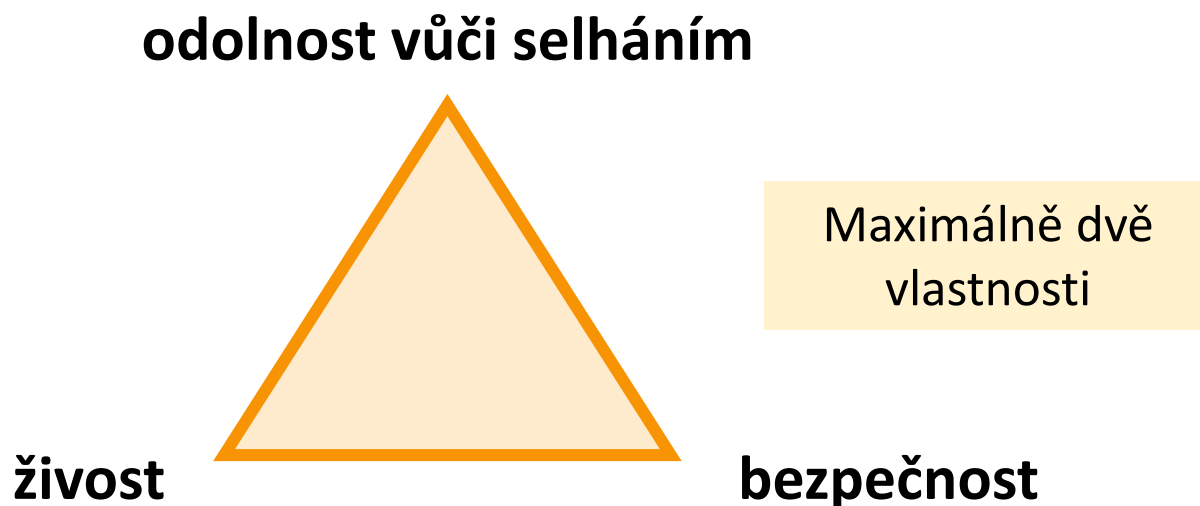
V asynchronních DS není konsensus řešitelný.

- nemůžeme využívat time-outů k rozlišení mezi selháním a zpožděním
- pro jakýkoliv algoritmus existuje nejhorší možný průběh (se selháními procesů nebo kanálů), který zabrání dosažení konsensu
- vyplývá v tzv. **FLP teorému**

FLP teorém

FLP teorém

V asynchronním distribuovaném systému **nelze** dosáhnout **současně bezpečnosti a živosti** distribuovaného výpočtu, pokud v něm může docházet k **selháním** (byť i jediného procesu).



Algoritmus Paxos

Nejstarší a nejznámější algoritmus pro distribuovaný konsensus (Leslie Lamport)

Pracuje v kolech; každé kolo má unikátní číslo volebních zpráv

Kola jsou asynchronní

- Je-li proces v kole j a dorazí-li mu zpráva z kola $j + 1$: přeruší činnost v rámci kola j a posuň se do kola $j + 1$
- využívá časová time-outy (může být pesimistický)

Každé kolo rozděleno do třech fází (které jsou taky asynchronní)

1. Fáze ELECTION: Je zvolen lídr
2. Fáze BILL: zvolený lídr navrhne hodnotu, ostatní procesy potvrzují
3. Fáze LAW: Lídr rozešle všem ostatním procesům finální (potvrzenou) hodnotu

Řešitelnost konsensu

Ale: V praxi vždy vyžadujeme bezpečnost a díky částečné synchronicitě ve velkém množství běhů DS konsensu dosáhneme v **konečném čase (tzv. *konečná živost* – *eventual liveness*)**.

- existují i pravděpodobnostní algoritmy mající **konečnou středního** hodnotu běhu

Souhrn

Problém konsensu je v jádru mnoha problémů v DS.

V asynchronním DS **nelze** při přítomnosti selhání **konsensus vyřešit** ve smyslu bezpečnosti a živosti.

Praktická řešení většinou garantují **bezpečnost**.