

# Globální stav distribuovaného systému

---

23. května 2018

B4B36PDV – Paralelní a distribuované výpočty

Díky, že chodíte na  
cvičení :-)

Vyplňte nám prosím  
**FINÁLNÍ ANKETU**  
předmětu PDV! →



[https://goo.gl/forms/  
cuMSyDjbm30WohCx2](https://goo.gl/forms/cuMSyDjbm30WohCx2)

- Opakování z minulého cvičení
- Globální stav distribuovaného systému
- Chandy-Lamportův algoritmus
- Konzultace semestrální práce

## Opakování z minulého cvičení

---

<http://goo.gl/a6BEMb>

### Jakým způsobem Raft zpracovává klientské požadavky?

*Zvolte, které z následujících možností platí*

- všechny požadavky splní
- splní jen požadavky, které leader klientovi potvrdí
- splní jen požadavky, které si zapíše do logu nadpoloviční většina serverů
- potvrzené požadavky může ze svého logu mazat jen nový leader
- nepotvrzené požadavky si může z logu smazat jakýkoli server

### Jakým způsobem Raft používá leadera?

*Zvolte, které z následujících možností platí*

- leader má vždy nejvyšší index z běžících procesů
- kandidát na leadera musí mít nejnovější log
- pouze leader může posílat požadavky o zápis do logů followerům
- při výpadku leadera Raft přestane fungovat navždy
- v systému může být vždy nanejvýš jeden leader
- systém může být několik epoch bez leadera

## **Globální stav v DS**

---



Detekovat vlastnost systému bývá **zásadní**

→ Musíme umět reagovat na události!

- Nevyužívaná paměť
- Deadlock
- Ukončení výpočtu
- ...

V *paralelním systému* již víme jak na to

Detekovat vlastnost systému bývá **zásadní**

→ Musíme umět reagovat na události!

- Nevyužívaná paměť
- Deadlock
- Ukončení výpočtu
- ...

V *paralelním systému* již víme jak na to

---

Jak to vyřešit v případě DS?

Musíme detekovat globální stav systému!

Jak to udělat v jednoduchém synchronním systému?

## Musíme detekovat globální stav systému!

Jak to udělat v jednoduchém synchronním systému?

→ Použijeme fyzické hodiny nebo tiky simulace!

A co když je nemáme?

## Musíme detekovat globální stav systému!

Jak to udělat v jednoduchém synchronním systému?

→ Použijeme fyzické hodiny nebo tiky simulace!

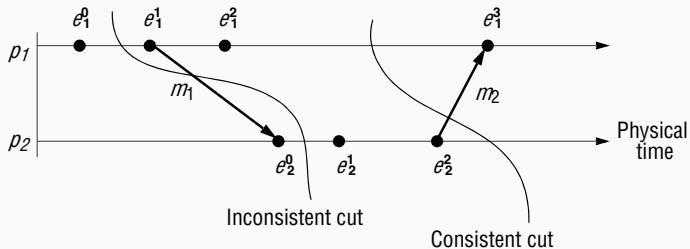
A co když je nemáme?

→ Musíme použít lokální stavy!

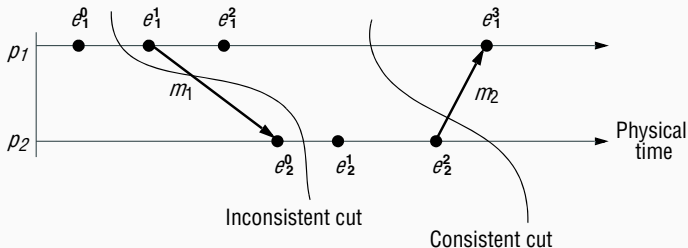
---

Globální stav potom bude sjednocením lokálních stavů  
≈ řez distribuovaným systémem

Kdy prohlásíme řez (globální stav) za konzistentní?



Kdy prohlásíme řez (globální stav) za konzistentní?



→ Musí být splněná kauzalita

Jak vynutit aby globální stav byl vždy konzistentní?

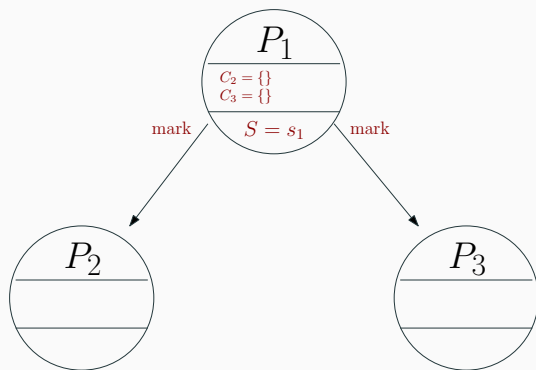
Zaznamenávání lokálních stavů budeme spouštět postupně...

...prohledáváním do hloubky!

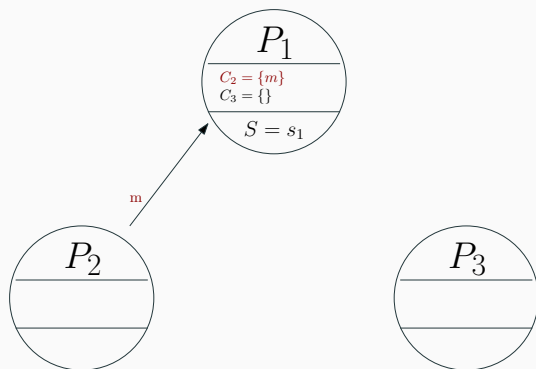


## Chandy-Lamportův algoritmus

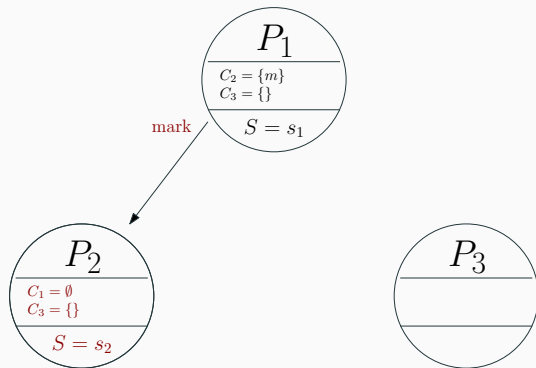
---



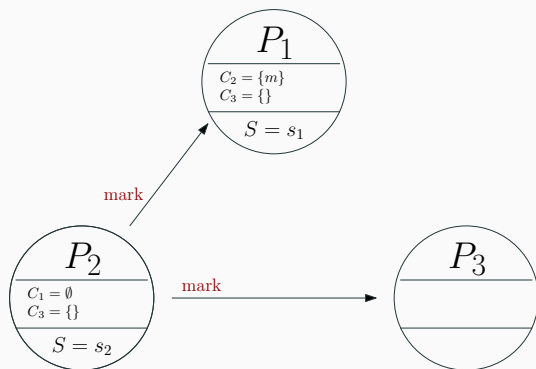
- Pokud chce proces  $P_i$  začít snapshot, zaznamená svůj stav  $S = s_i$ , otevře nahrávání zpráv na všech vstupních kanálech a pošle zprávu **mark** všem procesům, se kterými může komunikovat.



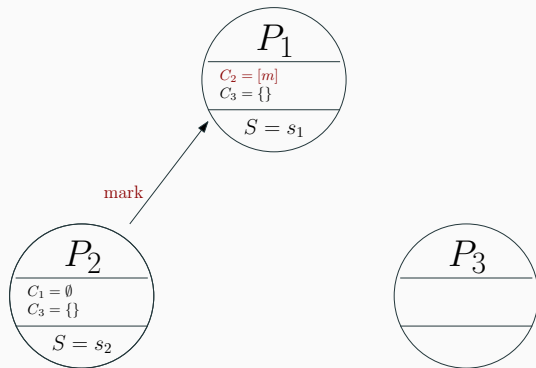
- 
- Pokud proces  $P_i$  přijme zprávu po kanálu, který nahrává, tak ji zaznamená.



- Pokud proces  $P_j$  přijme zprávu **mark** a ještě nenahrává, pak zaznamená svůj stav  $S = s_j$ , otevře nahrávání zpráv na všech vstupních kanálech (krom odesílatele **mark**) a pošle zprávu **mark** všem procesům, se kterými může komunikovat.



- Pokud proces  $P_j$  přijme zprávu **mark** a ještě nenahrává, pak zaznamená svůj stav  $S = s_j$ , otevře nahrávání zpráv na všech vstupních kanálech (krom odesílatele **mark**) a pošle zprávu **mark** všem procesům, se kterými může komunikovat.



- Pokud chce proces  $P_i$  přijme zprávu **mark** a nahrává, pak nahrávání na tomto kanále ukončí.

## Chandy-Lamportův algoritmus (shrnutí)

Když přijde zpráva **mark** od procesu  $P_i$

- Pokud ještě není zaznamenaný lokální stav: tak ho proces zaznamená a začne nahrávat na všech vstupních kanálech kromě kanálu od  $P_i$ .
- jinak: ukončí nahrávání na kanálu od  $P_i$ .

Kdy poslat zprávu **mark**

- Jakmile proces zaznamená svůj lokální stav, tak pošle zprávu **mark** všem procesům, se kterými může komunikovat.  
(před tím, než pošle jakoukoli jinou zprávu)

### Doprogramujte Chandy-Lamportův algoritmus

Doimplementujte logiku Chandy-Lamportova algoritmu ve třídě `snapshot/BankProcess.java` v následujících 2 metodách:

`processRecordRequest(sender)` zpracovává přijaté **mark** zprávy, zatímco `recordTransaction(hodnota, sender)` zpracovává **ne-mark** zprávy. Po skončení nahrávání reportujte sumu hodnot v nahraných zprávách + vlastní hodnotu (= stav) pomocí

`stateSnapshotCollector.sendSnapshotOfBalance(hodnota)`. Scénář spustíte v `snapshot/SnapshotRun.Main`.

Díky, že chodíte na cvičení :-)

Kromě finální PDV ankety vyplňte prosím i

**OFICIÁLNÍ ANKETU FEL**

Podle ní jsme hodnoceni fakultou MY



Díky za pozornost!

Pokud nám chcete něco sdělit  
ohledně posledního cvičení →



[https://goo.gl/forms/  
DnKuKvur3lsrW4dI3](https://goo.gl/forms/DnKuKvur3lsrW4dI3)