

PDV 13 2019/2020

# Závěr a shrnutí

Michal Jakob

[michal.jakob@fel.cvut.cz](mailto:michal.jakob@fel.cvut.cz)

Centrum umělé inteligence, katedra počítačů, FEL ČVUT



# Hlavní výzvy DS

Asynchronicita

Selhání

# Selhání

Jak procesy, tak komunikační kanály mohou v DS selhat.

## Selhání procesu

- **havárie (crash/fail-stop):** proces přestane vykonávat algoritmus (a reagovat na zprávy)
- **libovolné (byzantské) selhání:** proces může pracovat dále (a reagovat na zprávy), ale vykonává chybný algoritmus (z důvodu softwarové chyby nebo úmyslu)

## Selhání kanálu

- **ztráta zprávy (message drop):** zpráva není doručena cílovému procesu (např. kvůli přetížení sítě nebo přetečení zásobníku v OS u přijímacího procesu)
- **rozdělení (partitioning):** procesy jsou rozdělené do disjunktních množin (oddílů - partitions) tak, že v rámci oddílu je komunikace možná, ale mezi oddíly nikoliv

V případě synchronních DS definujeme ještě **selhání časování**, pokud doba odezvy procesu nebo přenosu zprávy po síti vybočila z dohodnutého **časového rozmezí**.

# Synchronní vs. Asynchronní

## Asynchronní systém

- Žádné časové limity na **relativní rychlost** vykonávání procesů.
- Žádné časové limity na **trvání přenosu** zpráv.
- Žádné časové limity na **časový drift** lokálních hodin



## Synchronní systém

- **Synchronní výpočty:** známe horní limit na relativní rychlost vykonávání procesů.
- **Synchronní komunikace:** známé horní limit na dobu přenosu zpráv.
- **Synchronní hodiny:** procesy mají lokální hodiny a je znám horní limit na rychlosti driftu lokálních hodin vzhledem k globálním hodinám.

Dále: Částečně synchronní systém

# Korektnost v DS

## Živost (Liveness)

Garance, že v DS *časem* dojde k něčemu **dobrému** (bude dosažen žádoucí stav).

(živost prakticky souvisí s **dostupností** systému)

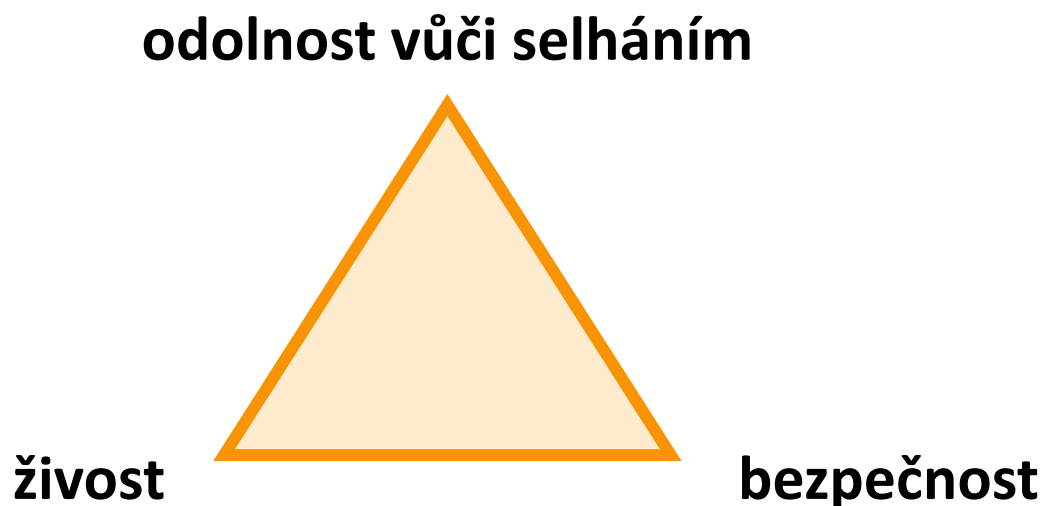
## Bezpečnost (Safety)

Garance, že v DS *nikdy* nedojde k něčemu **špatnému** (nebude dosažen nežádoucí stav).

# 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).



# Řešitelnost problémů

V praxi většinou vyžadujeme bezpečnost a díky částečné synchronicitě ve velkém množství běhů distribuovaných algoritmů dosáhneme výsledků v **konečném čase (tzv. *konečná živost – eventual liveness*)**.

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

# Problémy

Problém (algoritmy)	Model (zjednodušeně)	Garance
<b>Detekce selhání</b> (centrální, kruhový, all-to-all, SWIM)	asynchronost + selhání	živost
<b>Kauzalita a čas</b> (fyzikální, Lamportovy, vektorové hodiny)	asynchronost	bezpečnost + živost
<b>Globální snapshot</b> (Chandy-Lamport)	asynchronost	bezpečnost + živost
<b>Vyloučení procesů</b> (kruhový, Ricart-Agrawala)	asynchronost	bezpečnost + živost
<b>Volba lídra</b> (Raft, kruhový, Bully)	asynchronost + selhání	bezpečnost (za dodatečných předpokladů)
<b>Konsensus</b> (Raft)	asynchronost + selhání	bezpečnost



# Materiály

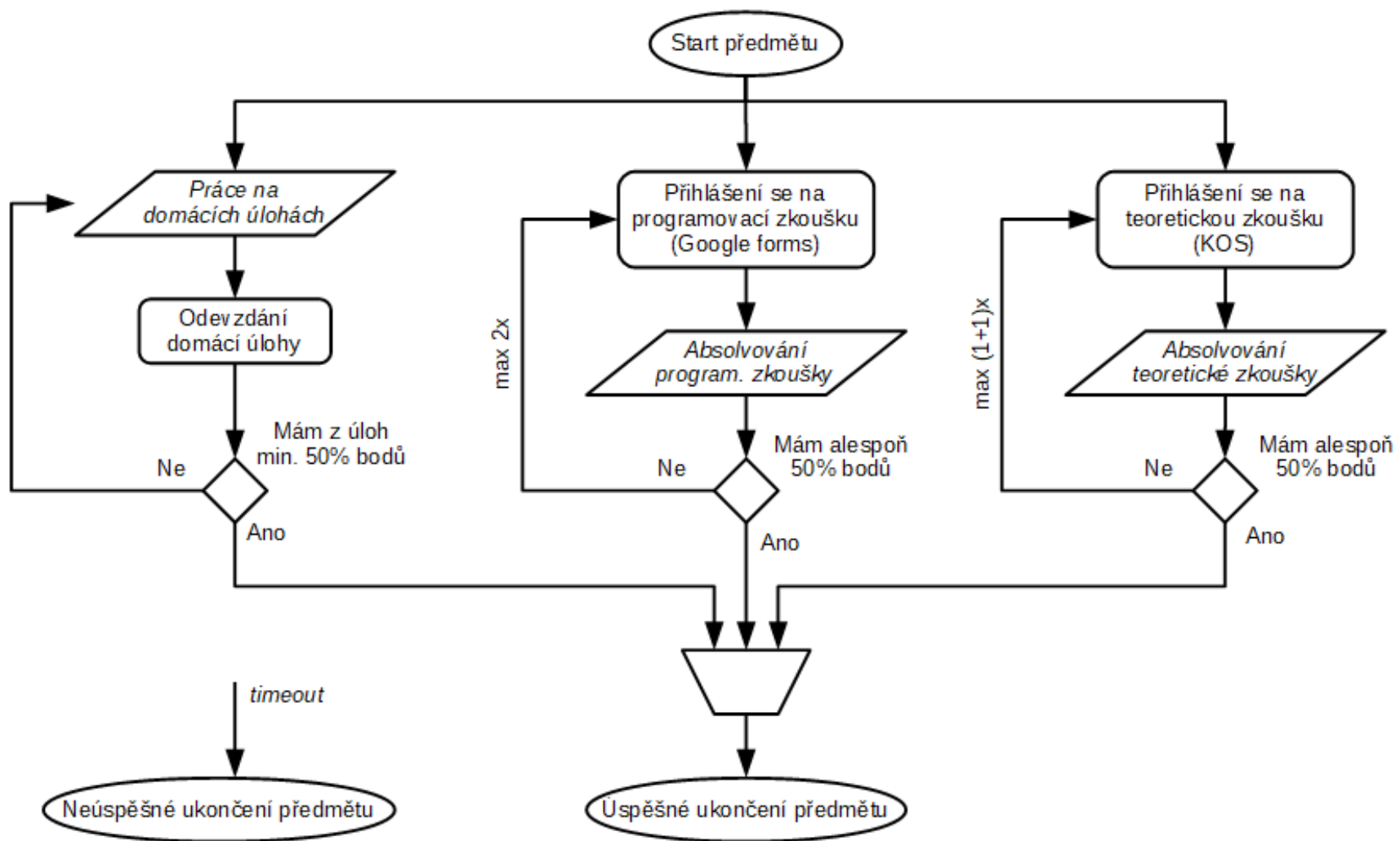
Úvod a modely	[Steen] 1.1,1.3; [Coulouris] 2.4.1, 2.4.2
Detekce selhání	[SWIM]
Kauzalita a čas	[Steen] 6.1-6.2; [Coulouris] 14.1-14.4
Globální snapshot	[Coulouris] 14.5
Vyloučení procesů	[Steen] 6.3; [Coulouris] 15.2
Volba lídra	[Steen] 6.4; [Coulouris] 15.3
Konsensus	[Steen] 8.2; [Coulouris] 15.5 [Raft]

[Steen] Van Steen, M. And Tanenbaum, A.S., 2017. *Distributed systems: principles and paradigms (3.01 Edition)*.[\[link\]](#)

[Colouris] Coulouris, G.F., Dollimore, J. and Kindberg, T., 2005. *Distributed systems: concepts and design*.

[SWIM] Das, A., Gupta, I. and Motivala, A., 2002. Swim: Scalable weakly-consistent infection-style process group membership protocol. In *Dependable Systems and Networks*, 2002. [\[link\]](#)

[Raft] Ongaro, D. and Ousterhout, J.K., 2014, June. In search of an understandable consensus algorithm. In *USENIX Annual Technical Conference*. [\[link\]](#)



# Ukončení předmětu

## **Teoretická** zkouška (max 40 bodů)

- on-line
- přihlašování přes KOS
- termíny: 26.5., 2.6. nebo 9.6., 16.6. + září

## **Programovací** zkouška (max 20 bodů)

- pravděpodobně on-line
- přihlašování skrze formulář
- aktuálně navrženy **pátky** dopoledne, začíná se 28.5.

Další termín bude v září (programovací i teoretická).

Zápočet, programovací a teoretická zkouška jsou **nezávislé**, lze udělat v libovolném pořadí.