KIV/DS 2020/2021

2. SAMOSTATNÁ PRÁCE

Mezní termín odevzdání: 1.2.2020 – 0:00

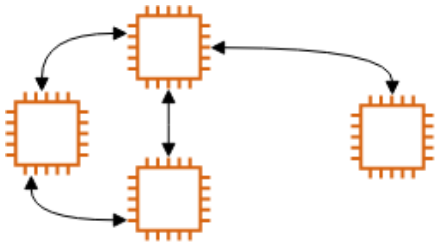
Maximální počet bodů: 25 **Minimální počet bodů: 15**

Zadání:

Implementace ZeroMQ služby s integrovaným Chandy-Lamportovým algoritmem pro získání konzistentního snímku globálního stavu

- Implementujte bankovní službu s operacemi (lze využít implementaci z předchozí úlohy)
  - credit – přičtení částky k zůstatku na účtu
  - debit – odečtení částky od zůstatku na účtu

  tak, že obě operace lze provést na základě zprávy doručené do vstupní ZeroMQ fronty následovně: pokud klient chce poslat částku do jiné banky, odečte částku a pošle zprávu CREDIT s částkou do jiné banky (fronty). Pokud chce klient inkasovat částku z jiné banky, pošle pouze zprávu DEBIT s částkou, přičemž druhá strana částku odečte a pošle ji ve zprávě CREDIT.
- Všechny uzly jsou identické, kromě IP adresy a jména uzlu, jejich počet je v rozmezí 2-5
- Každý uzel po startu začíná se zůstatkem na účtu 5.000.000.
- Uzly náhodně vyberou jednu z operací CREDIT či DEBIT s náhodnou částkou v intervalu <10.000, 50.000> a odešlou zprávu s operací na náhodně vybraný sousední uzel. Pokud uzel na svém účtu nemá dostatečnou částku, operaci odmítne.
- Uzly spolu komunikují pouze přes vstupní ZeroMQ fronty (podle vzoru PAIR, viz https://learning-0mq-with-pyzmq.readthedocs.io/en/latest/pyzmq/patterns/pair.html), které představují komunikační kanály. Zprávy přenášejte v JSON formátu (viz https://pyzmq.readthedocs.io/en/latest/serialization.html)
- Topologii si zvolte sami, ale je nutné ji uvést v průvodní dokumentaci, např:



- Jako servisní službu implementujte Chandy-Lamportův distribuovaný algoritmus pro získání konzistentního snímku globálního stavu podle stejného principu jako bankovní API (zpráva “marker” ve vstupní frontě – viz popis algoritmu)
- Spuštění Chandy-Lamportova algoritmu zahajte posláním zprávy MARKER do libovolné fronty (z příkazové řádky na libovolném uzlu).
- **Ošetřete i případ, že může být v jednom okamžiku spuštěno více snapshotů**
- **Stavem uzlu** se rozumí zůstatek na účtu v okamžiku přijetí prvního markeru, **stavem kanálu** seznam zpráv s bankovními operacemi přijatými po prvním markeru (viz popis algoritmu)
- Jakmile detekujete ukončení algoritmu, všechny uzly pošlou svoje uložené stavy do vyhrazené fronty, ze které je vyhrazený proces vybírá a zobrazuje (stačí na konzoli)
- K vytvoření infrastruktury použijte nástroje Docker a Kubernetes (Kubelet)
- Aplikaci můžete implementovat v Jazyce Java nebo Python s využitím již existujícího software, který vám usnadní implementaci jednotlivých modulů a jejich vzájemnou komunikaci.
- Zdrojové kódy udržujte a publikujte v repozitáři https://gitlab.kiv.zcu.cz/