Volba lídra - Bully - python sockety - distribuovaný výpočet

DSV - semestrální práce

Jan Hošťálek

01/01/2023

1 Návrh struktury

Cílem mé práce bylo navrhnout řešení distribuovaného výpočtu. Výpočet je řízen koordinátorem (leaderem), který má za úkol rozdělit zadanou úlohu na dílčí výpočty, které jsou následně rozdistribuovány a paralelně zpracovány ostatními uzly (followery, workery). Obsahem práce je, kromě samotné volby lídra, i výstavba distribuované sítě uzlů, její údržba a následný zánik po ukončení výpočtu. Předmětem samotného výpočtu je transkripce audio souborů.

Návrh jsem rozdělil do několika částí:

- Inicializace a propojení výpočetních uzlů
- Volba leadera využitím Bully algoritmu
- Tvorba a zpracování dílčích úloh
- Ukončení výpočtu

Navržený přístup je implementován v pythonu s využitím python socketů pro síťovou komunikaci. Jednotlivé uzly jsou spuštěny v oddělených virtuálních strojích. Což slouží kromě oddělení výpočetních prostředků i k virtualizaci síťové komunikace - každý uzel má unikátní síťovou adresu. Konkrétně jsou adresy přidělovány v rozsahu 192.168.56.101 až 192.168.56.255.

2 Inicializace a propojení výpočetních uzlů

2.1 Node

Distribuční síť je složena z několika navzájem propojených uzlů. Při inicializaci jsou si všechny uzly rovny a kandidátem na koordinátora výpočtu může být libovolný z nich. Proto je implementace sjednocena v souboru Node.py. Uzel je identifikován přidělenou adresou a portem a udržuje si všechny potřebné atributy po celou dobu běhu.

2.2 Sender a Receiver

Pro lepší čitelnost kódu jsou ve třídě Node obsaženy pouze části spojeny s logikou údržby topologie sítě, volba leadera (implementace bully algoritmu) a samotné zpracování dílčího úkolu. O posílání zpráv pro zvoleného souseda se stará metoda MessageSender, která zprávy, potomky třídy Message, pomocí TCP socketů posílá. Naopak třída MessageReceiver obsahuje po celou dobu běhu aktivní vlákna čekající na zprávy.

2.3 Datové úložiště

Datové úložiště slouží nejen k uchovávání samotných zpracovávaných audio souborů, ale také jako stavový automat. To znamená, že v průběhu výpočtu od leadera přicházejí CheckpointMessage, tak aby pří selhání koordinátora mohl nově zvolený pokračovat a nemusel výpočet začínat od začátku. Datový uzel běží odděleně na jednom libovolném virtuálním stroji a předpokladem navrženého přístupu je, že úložiště neselhává. Při spouštění jednotlivých uzlů je třeba poskytnout skrze `--data_center_ip` ip adresu stroje na kterém tento proces běží. Komunikace s datovým úložištěm probíhá pomocí TCP na portu 5556, případně libovolném `--data_center_port`.

2.4 Propojení uzlů

Při startu uzlu se pošle pomocí UDP broadcastu zpráva RequestConnectionMessage všem posluchačům - již spuštěným uzlům. Jako reakce na tuto zprávu se otevře TCP spojení a pošle se zpět ConnectionAcceptanceMessage. Po příjetí této zprávy na straně nového uzlu se uloží adresa odesílatele do self.neighbors a handshake se potvrdí odesláním ConnectionEstablishedMessage, viz obrázek 1. Uzly již existující sítě si uloží adresu nově připojené nody a zkontrolují podmínky volby leadera.

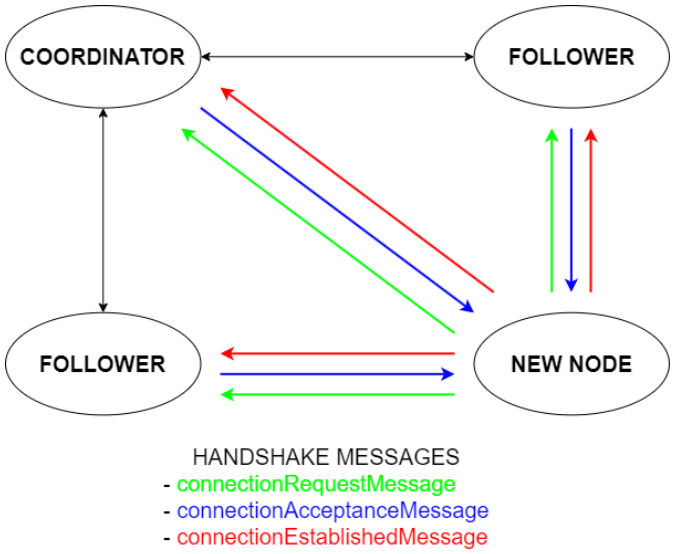


Figure 1: Připojení uzlu do sítě.

Druhou situací, která může nastat je, že již existující síť má leadera a vykonává úlohy. V takovém případě se novému uzlu v rámci zprávy ConnectionAcceptanceMessage pošle i adresa již existujícího leadera. Node rovnou přejde do režimu follower a zažádá koordinátora o práci.

3 Volba leadera Bully algoritmem

Předpoklady [1]:

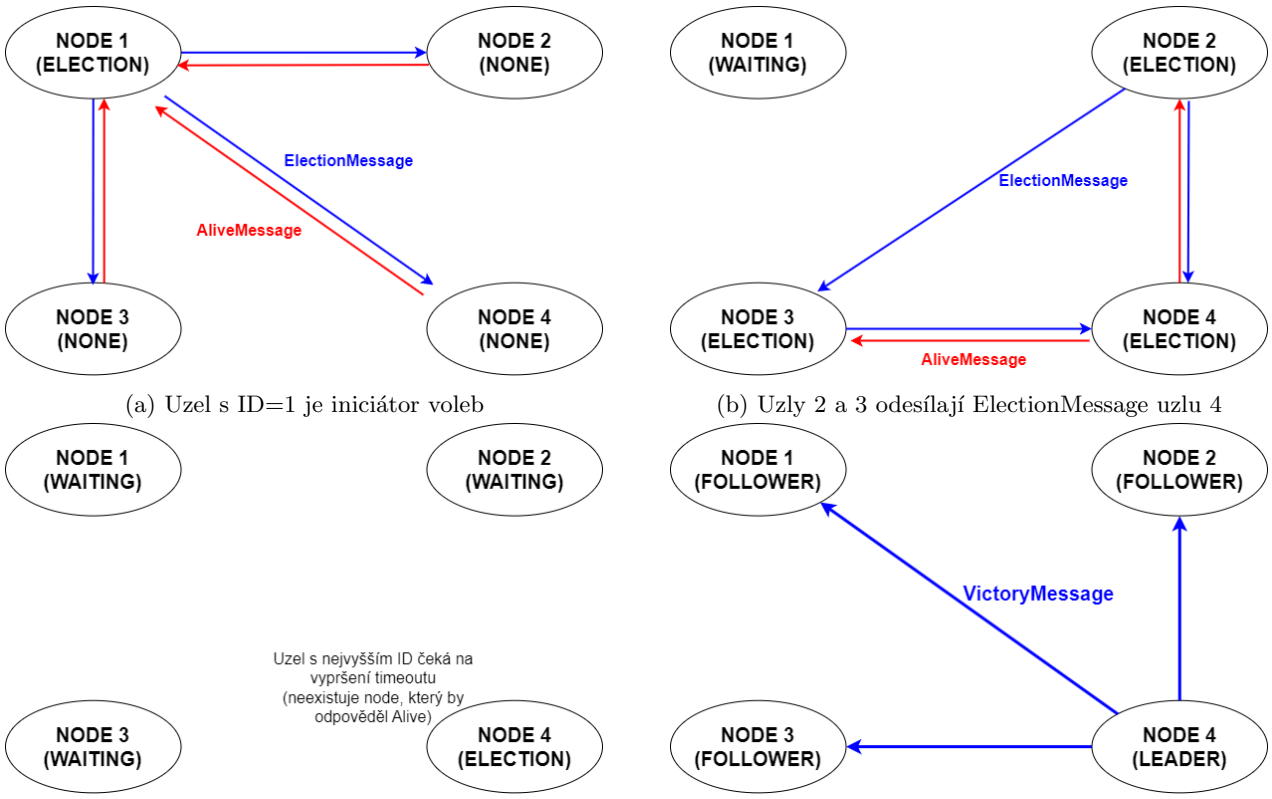
- systém je synchronní
- procesy mohou v průběhu voleb selhávat
- zprávy se neztrácejí
- každý uzel má svoje id a ukládá si id všech ostatních procesů

Ke spuštění procesu voleb dojde pokud má existující síť alespoň 3 navzájem propojené uzly, tzn. uzel má v self.neighbors alespoň 2 sousedy. Všem sousedům s vyšším id se rozešle zpráva ElectionMessage a přejde se do stavu ELECTION. Na straně příjemců se v případě že má odesílatel menší id (v mém případe 192...101 < 192...115) vytvoří a pošle zpět AliveMessage a rozešle se všem sousedům s vyšším id ElectionMessage. Node se prohlásí za koordinátora pokud v předem stanovené době nepřijde žádna AliveMessage. Což znamená, že neexistuje žádný kandidát s vyšším id. V takovém případě se volby ukončují a rozesílá se VictoryMessage. Na tuto zprávu čekají uzly, kterým v průběhu voleb přišla alespoň jedna AliveMessage.

Leader po skončení voleb zažádá datové centrum o aktuální checkpoint, případně vytvoří nové tasky. Naopak follower po přijetí VictoryMessage zažádá nově zvoleného leadera o task.

4 Distribuovaná transkripce audio souborů

Follower si po přijetí TaskMessage, která obsahuje identifikátor požadovaného tasku, vyžádá od datového centra samotné audio, které následně začne zpracovávat. Po dokončení odesílá ResultMessage, společně s výsledkem, leaderovi. Ten si mezivýsledek uloží a zaznamená změnu stavu - pošle data centru CheckpointMessage.



(a) Uzel s ID=1 je iniciátor voleb

(b) Uzly 2 a 3 odesílají ElectionMessage uzlu 4

(c) ID=4 timeout

(d) Uzel 4 vyhrál volby a odesílá VictoryMessage

Figure 2: Proces volby koordinátora Bully algoritmem.

Vzhledem k velmi vysokým nárokům na výpočetní zdroje pro transkripci audia je defaultně reálné audio nahrazeno pouze náhodným řetězcem. Transkripci reálných mp3 souborů lze zapnout `--real_audio True`. Ani v tomto případě ale nedochází k posílání audio souborů přes síť, ale načte se pouze příslušný soubor z paměti. Tzn. proces přeposílání audia je vynechán a každý node má ve složce data předpřipravených 20 tasků, ze kterých vybírá na základě přiřazeného task id.

5 Ukončení výpočtu

Leader v případě, že jsou všechny tasky ve stavu DONE, posílá poslední checkpoint datovému centru včetně zprávy TerminationMessage. Datové centrum vypíše celý transkript, uloží ho do souboru a ukončí běh. Koordinátor také ve stejnou chvíli rozešle všem následovníkům TerminationMessage, na základě které ze uzavře komunikace a ukončí běh všech uzlů.

References

[1] Bully algorithm. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 24. 11.2022 [cit. 2023-01-07]. Dostupné z: https://en.wikipedia.org/wiki/Bully_algorithm