



Implementace programu detekujícího ukončení distribuovaného výpočtu

Semestrální práce z předmětu B6B32DSV – Distribuované systémy a výpočty

Petr Vondrus

1 Téma práce

Předmětem této semestrální práce je implementace programu detekujícího ukončení distribuovaného výpočtu (termination detection). Výpočet ověřuje, zda zadané přirozené číslo je prvočíslem. Toto ověření je prováděno hledáním dělitele hrubou silou, které je rozloženo mezi výpočetní uzly. Pro detekci ukončení výpočtu byl použit algoritmus Dijkstra – Feijen – Van Gasteren (dále jen DFG).

2 Programovací jazyk a vývojové prostředky

Program byl napsán v jazyku Java SE 8 ve vývojovém prostředí IntelliJ IDEA. K přenosu zpráv mezi jednotlivými uzly je použita technologie Java RMI (Remote Method Invocation).

3 Spuštění programu

Před vlastním spuštěním programu je na každém fyzickém stroji, který se bude účastnit výpočtu, nutné spustit instanci RMI registru:

```
rmistart
```

Poté je možné spustit program příkazem:

```
java -jar dcsp.jar par1 par2 par3
```

Jednotlivé parametry příkazové řádky mají tento význam:

par1 ... přirozené číslo k ověření

par2 ... zpožďovací konstanta výpočtu [*ms*]

par3 ... jméno preferovaného síťového rozhraní pro UDP broadcast

Na příkazové řádce může být uvedeno 0 až *n* parametrů. Pokud není uveden žádný parametr, je použito přednastavené číslo 62 710 561 a nulová zpožďovací konstanta.

4 Identifikace výpočetních uzlů

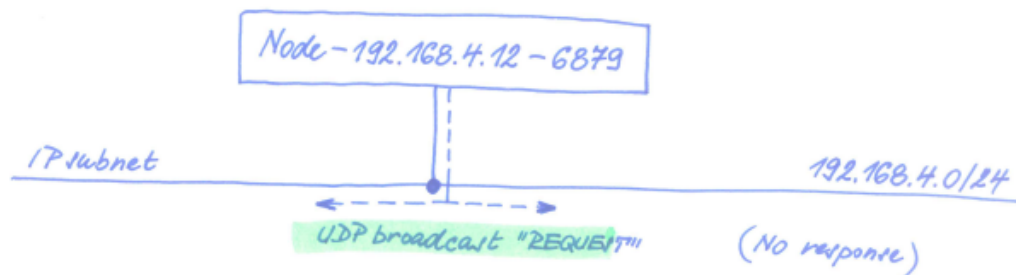
Každý výpočetní uzel má svůj jednoznačný identifikační řetězec ve formátu:

Node - *IP_ADDRESS-PORT* (např. Node - 192.168.4.12 - 6879)

Každému nově spuštěnému uzlu je zároveň přidělen index $i = \{0..n\}$, který slouží při implementaci algoritmu DFG.

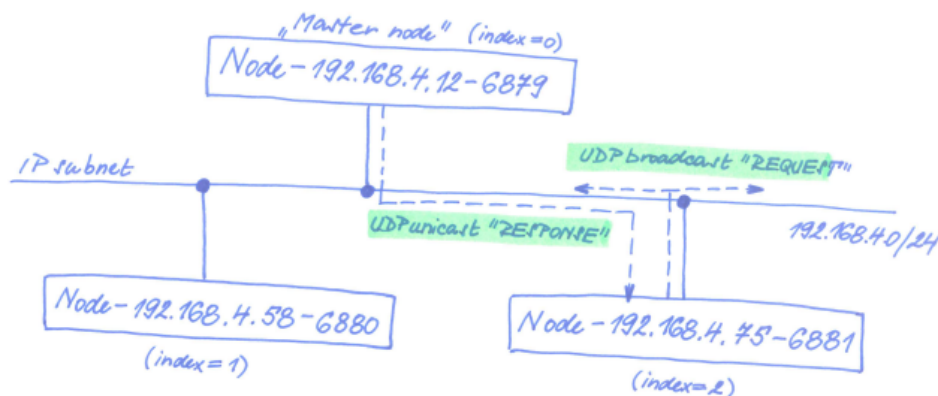
5 Vytvoření komunikační topologie

Každý nově spuštěný uzel nejprve vyšle do síťového IP subnetu, do kterého je připojen, *UDP REQUEST* paket. Pokud na tento požadavek neobdrží do určeného časového limitu odpověď v podobě *UDP RESPONSE* paketu, považuje se za jediný běžící uzel, označí se jako *Master node* (index = 0) a dále sám odpovídá na případné další požadavky nově spouštěných uzlů (Obr. 1).

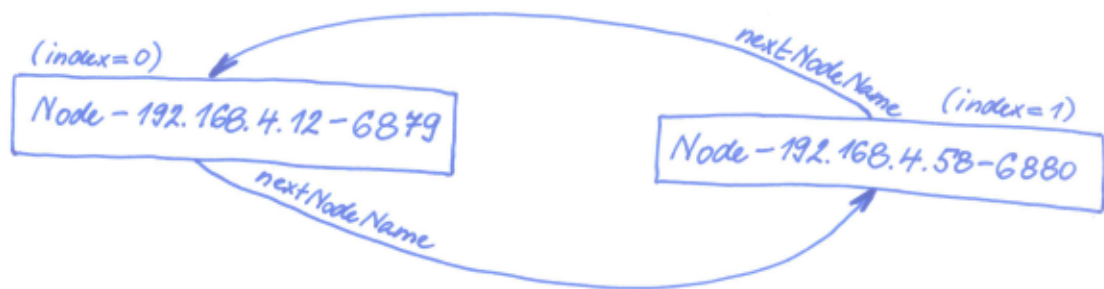


Obr. 1 – Spuštění prvního uzlu

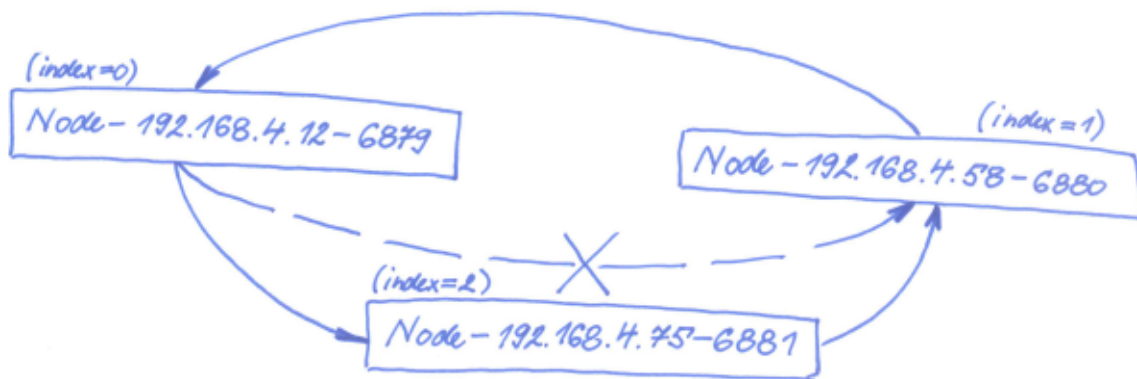
Pokud uzel obdrží odpověď *UDP RESPONSE*, zná tak IP adresu *Master node*, a požádá jej o připojení do komunikační topologie. *Master node* připojuje všechny nové uzly "před sebe", čímž vytváří topologii jednosměrného kruhu (unidirectional ring), ve kterém jsou uzly uspořádány sestupně podle přiděleného indexu (Obr. 2 – 4).



Obr. 2 – Spuštění následujících uzlů



Obr.3 – Připojení druhého uzlu



Obr.4 – Připojení n-tého uzlu

6 Zasílání zpráv

Každý uzel může poslat zprávu jinému uzlu, nebo všem ostatním uzlům (oběžník). Zprávy mohou být následujících typů:

JOB ... žádost o přidělení výpočtu a odpověď na ní (ovlivňuje DFG)

TOKEN ... zpráva algoritmu pro detekci ukončení (ovlivňuje DFG)

TERMINATE... příkaz k ukončení činnosti všech uzlů

DISJOIN ... žádost o ukončení činnosti jednoho uzlu

7 Provozní stavy uzlu

Po připojení do kruhové topologie se uzel může nacházet právě v jednom z těchto stavů:

ASK_FOR_JOB ... žádost o přidělení výpočtu

START_JOB ... spouštění výpočtu

WATCH_JOB ... sledování výpočtu
WAIT_FOR_MESSAGE ... čekání na zprávu
TERMINATION ... ukončování činnosti uzlu

8 Žádost o práci (přidělení výpočtu)

Nově připojený uzel a uzel, který skončí výpočet bez nalezení dělitele, odesílá oběžník se žádostí o přidělení výpočtu (*JOB*). Na tuto zprávu odpovídá první uzel, který zprávu přijme a má výpočet, jenž může přidělit. Uzly následující po něm již neodpovídají.

9 Ukončení činnosti uzlu

Činnost každého uzlu je možné ukončit zasláním signálu OS *SIGTERM*. Uzel odešle zprávu *DISJOIN* svému předchůdci v kruhu. V této zprávě uvede identifikaci svého následovníka, nedokončené výpočty, příp. nalezeného dělitele a atribut *Master node*. Po přijetí potvrzení se korektně ukončí.