Paralelní a distribuované algoritmy (PRL)

Pipeline Merge Sort

Daniel Konečný (xkonec75)
Vysoké učení technické v Brně
Fakulta informačních technologií

9. dubna 2021

1 Algoritmus Pipeline Merge Sort

Pipeline Merge Sort je paralelní řadicí algoritmus založený na slučování postupně delších už seřazených posloupností prvků (obdoba sekvenčního algoritmu Merge Sort). Prvně se seřadí jednotlivé prvky do dvojic, seřazené dvojice poté do čtveřic a analogicky až do potřebné délky posloupnosti. Každé slučování posloupnosti do jiné o dvojnásobné délce obstarává jeden procesor. Algoritmus tedy vyžaduje $log_2(n)+1$ procesorů v lineární struktuře, kde n je délka posloupnosti seřazovaných prvků. Sloučenou sekvenci prvků umístí jeden procesor tomu následujícímu na stejnou frontu a to tak, že fronty střídá pro každou seřazenou sekvenci.

První procesor má tedy na starost pouze umístit prvky neseřazené posloupnosti na fronty druhému procesoru a to tak, že po jednom prvku střídá tyto fronty. Každý další nasledující procesor i může začít pracovat jakmile na jedné z jeho posloupností je 2^{i-1} prvků a na druhé alespoň jeden prvek. Takový procesor musí zpracovat z každé své fronty přesně 2^{i-1} prvků předtím, než může začít zpracovávat další následující. Toto zpracování se skládá z odeslání menšího (při řazení prvků vzestupně) z čelních prvků na frontách dalšímu procesoru, přesněji jeho frontám. Případně, pokud už z jedné z front bylo odesláno 2^{i-1} prvků, odesílají se zbylé prvky z druhé fronty. Poslední procesor aplikuje stejný postup výběru prvků ke zpracování, pouze je místo posílání dalšímu procesoru skládá do výstupní seřazené posloupnosti.

Složitost algoritmu

Procesor na indexu i začíná s řazením až ve chvíli, kdy má na svých frontách $2^{i-1}+1$ prvků. Tato hodnota tedy značí počet cyklů, které musí udělat předchozí procesor před tím, než bude moci začít pracovat tento. Celkový počet cyklů před

začátkem řazení procesoru i je tedy:

$$1 + \sum_{j=0}^{i-1} 2^j + 1 = 2^i + i.$$

Procesor i poté běží po $n - 1$ cyklů, skončí tedy v cyklu

$$2^i + i + n - 1.$$

Celý algoritmus tedy pro k procesorů skončí po krocích daných níže uvedeným vzorcem, kde $k = log_2(n)$. To je tedy časová složitost algoritmu:

$$t(n) = 2^k + k + n - 1 = 2^{log_2(n)} + log_2(n) + n - 1 = 2n + log_2(n) - 1 = \mathcal{O}(n).$$

Prostorová složitost algoritmu je: $p(n) = log_2(n) + 1 = \mathcal{O}(log(n))$.
Cena algoritmu je: $c(n) = t(n) * p(n) = n * log_2(n) + n = \mathcal{O}(n * log(n))$.

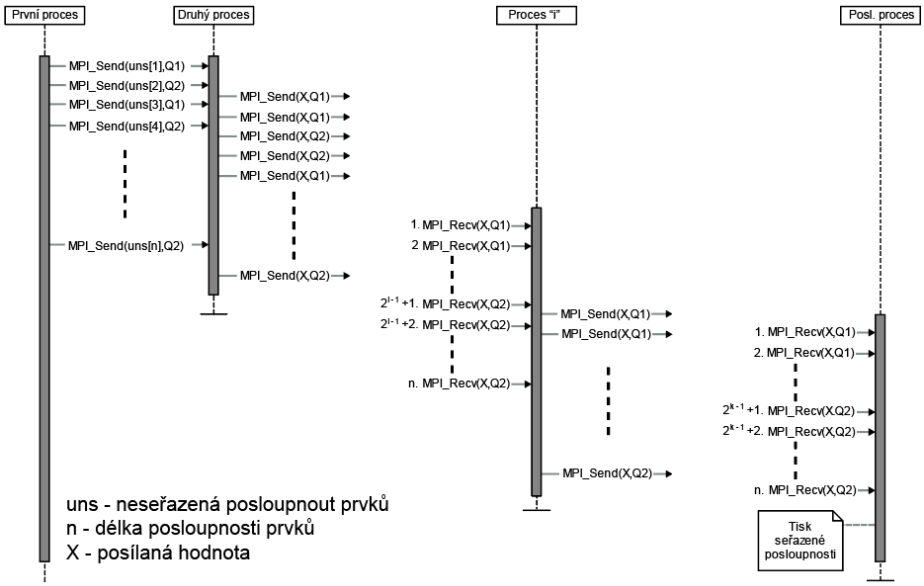
2 Implementace

Hlavní částí implementace je algoritmus pro zpracování činnosti procesoru spojujícího dvě seřazené posloupnosti. Tento algoritmus je zjednodušeně uvedený níže.

Algorithm 1: Procesor i spojující seřazené posloupnosti

```
while počet odeslaných prvků < 16 do
    instructions;
    if počet přijatých prvků > 16 then
        čekej na nový prvek;
        umísti prvek na správnou frontu podle počtu přijatých prvků;
        inkrementuj počet přijatých prvků;
    end
    if procesor není připravený then
        if je dostatečný počet prvků na frontách then
            nastav procesor jako připravený;
        else
            continue
        end
    end
    if na horní frontě je nižší prvek nebo je prázdná dolní fronta then
        odešli prvek z horní fronty;
    else if na dolní frontě je nižší prvek nebo je prázdná horní fronta
     then
        odešli prvek z dolní fronty;
    inkrementuj počet odeslaných prvků;
end
```

3 Komunikační protokol



4 Závěr

Algoritmus Pipeline Merge Sort byl plně naprogramován a dokáže řadit posloupnosti 16 jednobajtových čísel. Jeho rychlost nedosahuje tak dobrých výsledků jako je teoretická rychlost dle matematických výpočtů, a to kvůli nutnosti dodatečné režie. Vzhledem k povaze algoritmu není doba řazení závislá na podobě neseřazené vstupní posloupnosti. Může být tedy jak už seřazená, tak naopak úplně pozpátku, a na dobu běhu to nebude mít vliv.