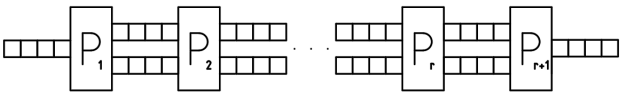
Paralení a distribuované algoritmy (PRL) 2020/2021
Pipeline Merge Sort

Daša Nosková (xnosko05)
xnosko05@stud.fit.vutbr.cz

4. apríl 2021

1 Rozbor algoritmu

Pipeline merge-sort[1] sa radí k paralelným radiacim algoritmom. Radenie prebieha pomocou liniek. Postupnosť má dĺžku $n = 2^r$ $(r > 0)$ a k dispozícii je $r + 1$ procesorov. Procesory (ozn. P) pracujú synchrónne a v jednom cykle vedia porovnať dve celé čísla a zároveň prečítať a zapísať jedno celé číslo.



Obrázek 1: Prepojenie procesorov

Ako vidieť na obrázku 1, P_1 má jednu vstupnú linku s nezoradenou postupnosťou čísiel a dve výstupné linky, P_{r+1} má naopak dve vstupné linky a jednu výstupnú, na ktorú sa predá konečná zoradená postupnosť. Všetky ostatné procesory majú dve vstupné aj výstupné linky.

Počas každého cyklu P_1 načíta jedno celé číslo zo vstupnej postupnosti a spracuje ho podľa pravidiel. Každý procesor okrem P_1 začína s radením, keď každý procesor pred ním vyprodukol čiastočne zoradenú postupnosť na jednej výstupnej linke a na druhej má 1 prvok druhej postupnosti. $P_1 - P_r$ vkladajú čiastočne zoradené postupnosti na hornú alebo dolnú výstupnú linku.

1.1 Analýza zložitosti

Procesor P_i začína radenie, keď na jednom z dvoch vstupov je postupnosť s dĺžkou 2^{i-1} a na druhom vstupe 1 prvok, tzn. po procesore P_{i-1} začne $2^{i-1}+1$ cyklov. Ak procesor P_1 začne radenie počas prvého cyklu, P_i začína spracovávanie o

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

cyklov neskôr. P_i skončí v cykle

$$(n-1) + 2^{i-1} + i - 1$$

Posledným procesorom je P_{r+1}, a teda radenie končí v cykle

$$n + 2^r + r - 1$$

z ktorého sa dá pomocou vzťahov 1 a 2 odvodiť rovnica č. 3 z ktorej vyplýva lineárna časová zložitosť $O(n)$.

$$log_2\, n = r \tag{1}$$

$$2^r = n \tag{2}$$

$$2n + log_2\, n - 1 \tag{3}$$

Priestorová zložitosť je určená ako $p(n) = log_2\, n + 1$. Cena algoritmu je vypočítaná v rovnici 4 pomocou vzťahu $c(n) = t(n) * p(n)$.

$$O(n) * (log_2\, n + 1) = O(n\, log_2\, n) \tag{4}$$

Z výsledku je zistená optimálna cena algoritmu, nakoľko je splnená podmienka $c_{optim}(n) = t_{seq}(n)$, pretože sekvenčný algoritmus Mergesort má časovú zložitosť $O(n\, log_2\, n)$.

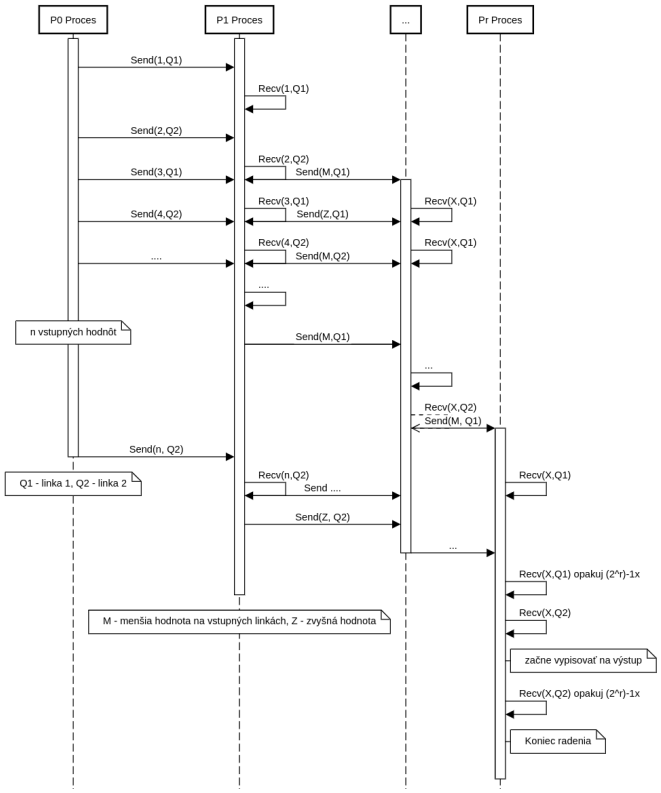
2 Implementácia

Pri implementácii bol použitý jazyk C++ a knižnica OPEN MPI. Algoritmus je implementovaný v súbore `pms.cpp`. Pred spustením samotného programu, je treba určit potrebný počet procesov potrebných k vykonaniu algoritmu. Počet procesov je automaticky vypočítaný v skripte `test.sh` ako $log_2\, n\, + 1$, kde n je dĺžka postupnosti.

Program začína zistením veľkosti binárneho súboru, z ktorej sa dá odvodiť veľkost vstupnej postupnosti, pretože veľkost súboru = počet prvkov postupnosti, nakoľko jedno číslo má veľkosť 1B. Program pokračuje zistením čísla práve pracujúceho procesu. Proces s číslom 0 vypíše postupnosť na štandardný výstup a prvok po prvku odošle

ďalšiemu procesu. Ak má proces iné číslo ako 0, pokračuje do funkcie `merge()`. Proces ostáva v danej funkcii kým nespracuje všetky prvky. V prvom rade potrebuje proces prijať prvok od predchádzajúce procesu pomocou funkcie `MPI_RECV` a prvok zaradiť do správneho radu. Vždy sa do každého radu zaradí prvých 2^{i-1} prvkov, kde i je číslo aktuálneho procesu. Táto časť je zabezpečená cez počítadlo counter a premennú tag. Proces začína posielať prvky, ak je prvý rad naplnený na 2^{i-1} prvkov a druhý obsahuje jeden prvok. Čísla sa porovnajú a menšie(príp. ak sa rovnajú, druhé) číslo je odoslané ďalšiemu procesu pomocou `MPI_SEND`. Algoritmus pracuje tak, že radí vždy podpostupnosti, takže ak algoritmus už vyčerpal všetky čísla z jedného alebo druhého radu, musí poslať zvyšnú časť podpostupnosti (zvyšný rad) skôr ako čísla z novej podpostupnosti. Tento mechanizmus zabezpečujú premenné Q1_send a Q2_send, ktoré sú inicializované na hodnotu 2^{i-1}, ktorá odpovedá hodnote maximálnej dĺžke radu. Vždy keď sa odošle prvok z rady Q1, tak sa Q1_send zmenší o jedno a tak isto pre radu Q2 a Q2_send. Ak je proces posledný, tak sa postupnosť neodosiela, ale začína výpis postupnosti na výstup.

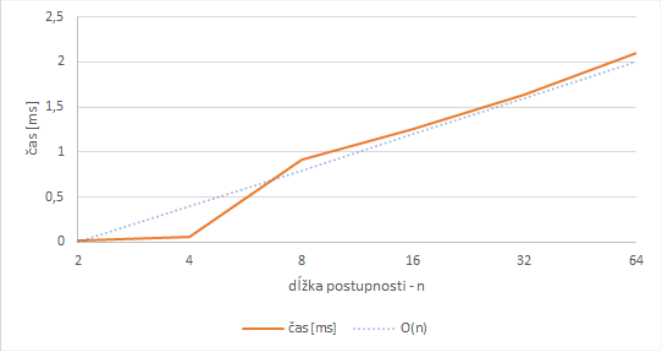
3 Komunikačný protokol



Obrázek 2: Komunikácia medzi procesmi

4 Experimenty

Boli vykonané experimenty s účelom overenia teoretickej časovej zložitosti trvania algoritmu v reálnom čase pomocou funkcie `chrono::high_resolution_clock::now()` pre postupnosti o dĺžke $n = 2^1$ až 2^6. Pre každé n boli experimenty opakované, následne bola odstránená najmenšia a najväčšia hodnota a zvyšné hodnoty boli priemerované, aby sa predišlo veľkým odchýlkam. Experimenty boli vykonané na počítači s procesorom Intel(R) Core(TM) i7-10510U 1.80GHz. Ako je vidieť na obrázku 2, reálna časová zložitosť rastie približne lineárne.



Obrázek 3: Výsledok experimentov

5 Záver

V rámci projektu bol úspešne implementovaný radiaci algoritmus pipeline merge-sort a overená jeho teoretická časová zložitosť, aj napriek tomu, že namerané hodnoty odpovedajú teoretickej časovej zložitosti až u dlhších postupností, čo sa dá predpokladať vzhľadom k tomu, že porovnať 2 či 4 hodnoty je rádovo menej náročné ako zoradiť viac prvkov.

Referencie

[1] Akl, S. G. *Parallel Sorting Algorithms*. Orlando: Academic Press, 1985. ISBN 0-12-047680-0.