

Dokumentácia projektu číslo 1.

Paralelné a distribuované algoritmy

Autor : Adam Švenk (xsvenk00)

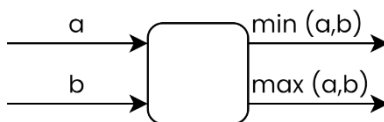
Dátum: 10.04.2022

Úvod

Predmetom tejto dokumentácie je analýza algoritmu „odd-even merge sort“. Okrem všeobecnej analýzy je súčasťou taktiež popis implementácie algoritmu pre zoradenie 8 vstupných hodnôt.

Analýza algoritmu „odd-even merge sort“

Algoritmus „odd-even merge sort“ je paralelný radiaci algoritmus, ktorý spája dve zoradené polovičné postupnosti do finálnej zoradenej sekvencie. Na rozdiel od algoritmu „merge sort“ nie je závislý na type porovnávaných dát, vďaka čomu je univerzálnejší. Na samotné radenie používa špeciálnu sieť procesorov, z ktorých každý obsahuje dva vstupy a dva výstupy. Ich prepojenie zobrazuje *Obrázok 1*. Každý procesor dokáže porovnať vstupné hodnoty a následne ich odoslať na korektný výstup.



Obrázok 1 Sieť procesorov 1x1

Analýzou algoritmu som dospel k jeho nasledovným vlastnostiam:

Počet procesorov: $p(n) = O(n \cdot \log^2 n)$

- Počet procesorov p potrebných k riešeniu úlohu v závislosti na veľkosti inštancie n

Časová zložitosť: $t(n) = O(m^2) = O(\log^2 n)$

- Čas riešenia potrebný k riešeniu úlohy v jednotkách krokov

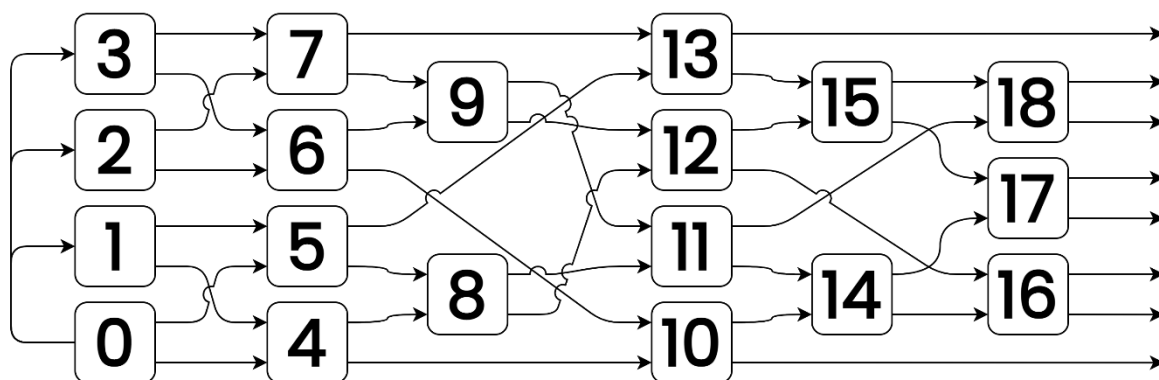
Celková cena: $c(n) = O(n \cdot \log^4 n)$

- Celková cena je vypočítaná ako $c(n) = p(n) \cdot t(n)$

Vďaka týmto vlastnostiam som prišiel na to, že tento algoritmus nie je optimálny.

Implementácia

Algoritmus je implementovaný v programovacom jazyku C. Na synchronizáciu samotných procesov je použitá knižnica `Open MPI`. Inicializácia exekučného prostredia prebieha pomocou funkcie `MPI_Init`. Nasleduje inicializácia získanie identifikátora každého procesu funkciou `MPI_Comm_rank`. Obe spomínané funkcie sú spúšťané všetkými procesmi. Po inicializácii prebieha načítanie 8 vstupných hodnôt zo súboru `numbers` prvým procesom (identifikátor 0) funkciou `fread`. Prvý proces v ďalšom kroku vypíše načítané hodnoty na výstup `stdout`. Po výpise prebehne synchronizácia procesov funkciami `MPI_Send` a `MPI_Recv`, pomocou ktorých prvý proces s identifikátorom 0 odošle procesom s identifikátorom 1 – 3 načítané vstupné hodnoty. Tieto procesy následne priradia svoje vstupné hodnoty na príslušné výstupy a odošlú ich pomocou funkcie `MPI_Send` čakajúcim procesom 4, 5, 6 a 7. Rovnakým spôsobom prebieha v ďalších krokoch porovnanie vstupných hodnôt a ich následné priradenie na správne výstupy. Spôsob prepojenia a komunikácie jednotlivých procesov je zobrazený na obrázku *Obrázok 2*. Po dokončení poslednej operácie každého procesu nasleduje posledný krok, a to odoslanie výsledných hodnôt. Táto operácia sa týka procesov 10, 13, 16, 17, 18 a ich výstupných hodnôt. Tieto výstupné hodnoty následne odošlú procesu s identifikátorom 0, ktorý získané hodnoty vypíše na `stdout`.



Obrázok 2 Prepojenie procesov pre radenie 8 vstupných hodnôt

Záver

Algoritmus bol implementovaný v programovacom jazyku C a následne s pomocou dodaného testovacieho skriptu `test.sh` preložený a otestovaný. Testovanie prebiehalo taktiež aj na referenčnom serveri `Merlin`. Výsledkom testovania bolo úspešné overenie funkcionality samotného programu, ktorý využíva radiaci algoritmus „odd-even merge sort“.