Odd-even transposition sort
Dokumentácia k 2. projektu

Peter Lukáč - xlukac11

Marec 2020

1 Analýza odd-even transposition sort

Odd-even transposition sort je paralelný algoritmus pre triedenie, ktorý používa lineárne pole procesorov, kde počet procesorov je rovnaký ako počet prvkov: n. Hodnoty vstupu sú označené y_1 až y_n a procesory sú očíslované p_1 až p_n. V našej implementácii na začiatku procesor p_i dostane hodnotu vstupu y_i. Počet procesorov algoritmu má teda lineárnu závislosť na dĺžke vstupu: $p(n) = n$.

Činnosť algoritmu je založená na interprocesorovej komunikácií, kde každý procesor y_i porovnáva a vymiena(transponuje) svoju hodnotu zo susedom y_{i+1} nasledujúcim spôsobom:

1. Nepárne(liché) procesory p_i porovnajú svoju hodnotu y_i so susedom p_{i+1} s hodnotou y_{i+1}. Ak $y_i > y_{i+1}$, tak svoje hodnoty vymenia
2. Párne(sudé) procesory p_i porovnajú svoju hodnotu y_i so susedom p_{i+1} s hodnotou y_{i+1}. Ak $y_i > y_{i+1}$, tak svoje hodnoty vymenia.
3. ...

V každom kroku sú hodnoty presunuté práve o jeden index správnym smerom. Preto presun hodnoty v poli y na správny index potrvá maximálne n krokov. Kroky teada opakujeme maximálne n-krát aby boli všetky hodnoty zoradené.

1.1 Teoretická zložitosť

Vykonávanie každého kroku má konštantný čas k a vykonajú sa n-krát. Teda časová zložitosť $t(n)$ je $O(n * k)$ + ostatné podporné činnosti algoritmu: načítavanie hodnôt hlavným procesorom ($O(n)$), rozoslanie hodnôt ostatným procesorom ($O(n)$) a zozbieranie hodnôt ($O(n)$) majú zložitosť $O(n) + O(n) + O(n) = O(n)$. Celková časová zložitosť je teda lineárna: $t(n) = O(n)$.

Cena algoritmu je $p(n) = p(n) * t(n) = n * O(n) = O(n^2)$ čo nieje optimálne.

2 Implementácia

Algoritmus je implementovaný v jazyku C++ a používa knižnicu `Open MPI` pre komunikáciu procesov.

Podľa popisu, algoritmus používa n procesorov. Pre praktickú implementáciu toto napodobňujeme požitím systémových procesov.

Prvý proces je zodpovedný za načítanie honôt zo súboru a rozoslanie týchto hodnôt medzi ostatné procesy. Následne sa tiež podiela na radení. Po ukončení radenia všetky procesy pošlú svoju hodnotu hlavnému procesu, ktorý vypíše zoradené hodnoty na štandartný výstup.

3 Komunikačný protokol

Komunkačný protokol zobecnený pre n-procesov.

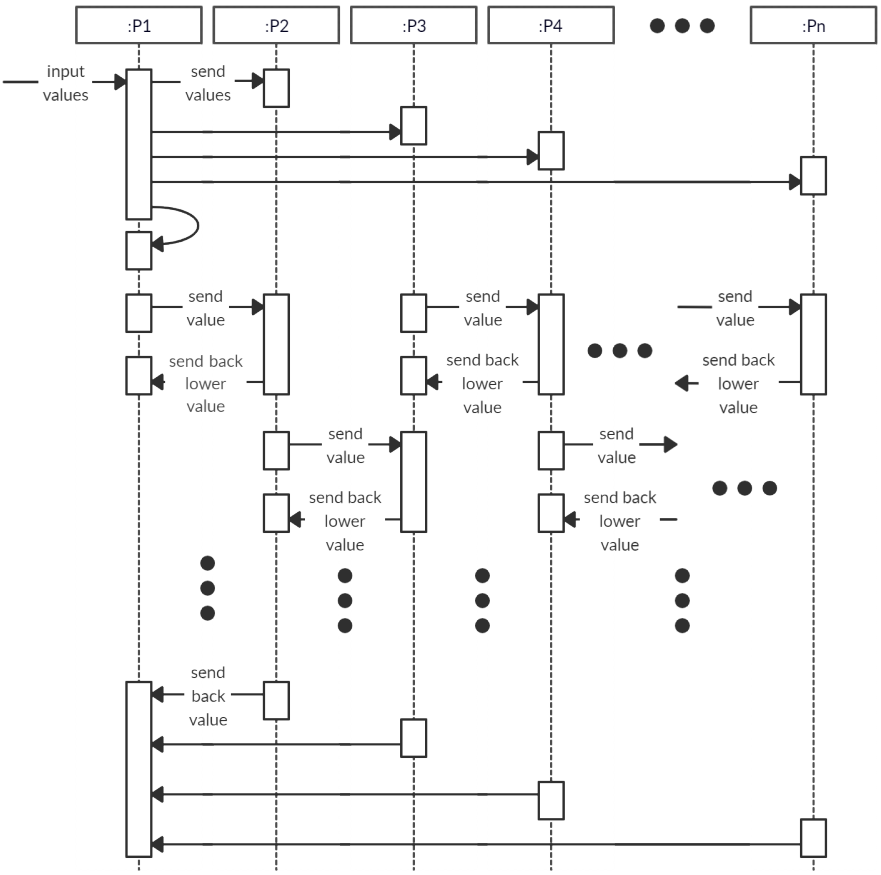

Figure 1: Komunikačný protokol

4 Experimenty a výsledky

Program bol testovaný a meraný na servery `merlin`. `Merlin` povoloval spúšťanie maximálne 25 procesov, preto sme testovali pre vstup maximálnej dĺžky 25.

Na meranie času bola použitá štandartná časová knižnica `chrono`. Časové známky sú merané len v časti radenia hodnôt. Predpokladá sa že načítavanie zo súboru a spätné zasielanie má lineárnu zložítosť a z hľadiska tohto projektu je nezaujímavé. Obzvlášt načítávanie trvá neúmerne dlhšie ako radenie.

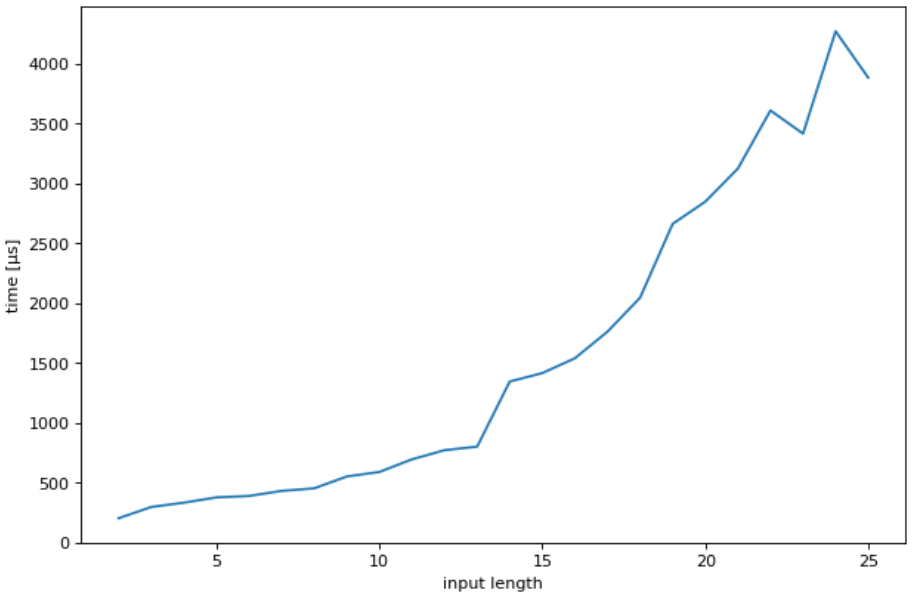

Figure 2: Výsledky merania

5 Záver

Namerané výsledky majú skôr charakteristiku $O(n^2)$ namiesto očakávanej teoretickej $O(n)$. Dôvodom je, že požívame procesy namiesto predpokladaných dedikovaných procesorov. Na 12 vláknovom servery(`merlin`) sme teda doplatili na prepínanie medzi procesmi a nedostali sme skutočný paralelizmus. Vo výsledku teda dostávame časovú zložitosť $O(n^2)$, čo je rovné celkovej cene algoritmu pre dĺžku vstupu väčšiu ako 12.