

Paralelní a distribuované algoritmy - dokumentácia

Bucket sort

Katarína Grešová

11. marca 2019

1 Úvod

Táto dokumentácia popisuje paralelný radiaci algoritmus Bucket sort – jeho implementáciu, časovú zložitosť, komunikáciu procesov a experimenty na overenie časovej zložitosti.

2 Rozbor a analýza algoritmu

Bucket sort je paralelný algoritmus na zoradenie poľa prvkov. Algoritmus pracuje na procesoroch usporiadaných do binárneho stromu.

Počet použitých procesorov sa odvíja od počtu radených prvkov a je potrebné ho vypočítať pred spustením samotného radenia. Počet radených prvkov označíme n . Strom procesorov potom obsahuje m listových procesorov, kde $n = 2^m$. Strom procesorov celkovo obsahuje $2 * m - 1$ procesorov, čo sa rovná $2 * \log(n) - 1$.

Pre počet radených prvkov $n = 2^k$, kde $k \geq 0$, je pri týchto podmienkach výpočet pomerne jednoznačný. Avšak, ďalej ešte potrebujeme počítať s obmedzením, ktoré vyplýva zo štruktúry binárneho stromu a to, že počet listových procesorov musí byť mocninou čísla 2.

Pre výpočet počtu procesorov bol použitý nasledujúci vzorec:

$$p(n) = \begin{cases} 1 & \text{if } 1 \leq n \leq 2 \\ 2^{\log_2(2 * \log_2 n)} - 1 & \text{otherwise} \end{cases}$$

kde logaritmus zaokrúhľuje výsledok nahor.

Po spustení algoritmu, master procesor načíta vstupnú sekvenciu čísel a spočíta, koľko čísel má spravovať každý listový procesor. Výsledok je celé číslo len v prípade, ak počet vstupných čísel je násobkom počtu listových procesorov. V opačnom prípade je výsledok zaokrúhlený nahor a načítaná vstupná sekvencia je doplnená číslami „-1“ aby každý listový procesor spravoval rovnaký počet prvkov. Informácia o počte prvkov pre listové procesory

je rozoslaná všetkým procesorom. Tento krok je potrebný, aby každý procesor vedel, ako veľkú sekvenciu čísel má očakávať. Následne master procesor rovnomerne rozošle sekvenciu vstupných čísel listovým procesorom.

V ďalšom kroku listové procesory príjmu sekvencie čísel a paralelne ich zoradia optimálnym sekvenčným radiacim algoritmom – v tomto prípade heap sort.

Nasleduje hlavný cyklus algoritmu, ktorý iteruje cez úrovne stromu procesorov počínajúc listovou úrovňou. V každom cykle, procesory aktuálnej úrovne odošlú svoje dáta otcovským procesorom. Otcovské procesory príjmu dáta od dvoch synov a spoja ich do jednej usporiadanej postupnosti. Telo tohto cyklu sa vykonáva paralelne.

Po skončení hlavného cyklu, master procesor obsahuje usporiadanú postupnosť čísel. Túto postupnosť vypíše na výstup s tým, že vynechá pridané čísla „-1“.

3 Teoretická zložitosť

Na výpočet časovej zložitosti budeme analyzovať nasledujúce kroky:

1. Načítanie sekvencie čísel zo vstupu – $O(n)$
2. Rozoslanie informácie o počte prvkov, ktoré dostanú listové procesory – $O(\log_2 n)$
3. Rozoslanie čísel listovým procesorom – $O(n)$
4. Zoradenie sekvencií čísel listovými procesormi – $O((n/\log_2 n) * \log_2(n/(\log_2 n))) = O(n)$
5. Hlavný radiaci cyklus beží $\log_2 m$ -krát. V každej iterácii sú spájané dve postupnosti s dĺžkou $n/2^i$, čo má zložitosť $(k * n)/2^i$. Celý cyklus teda trvá:

$$\sum_{i=1}^{\log_2 m} (k * n)/2^i = O(n)$$

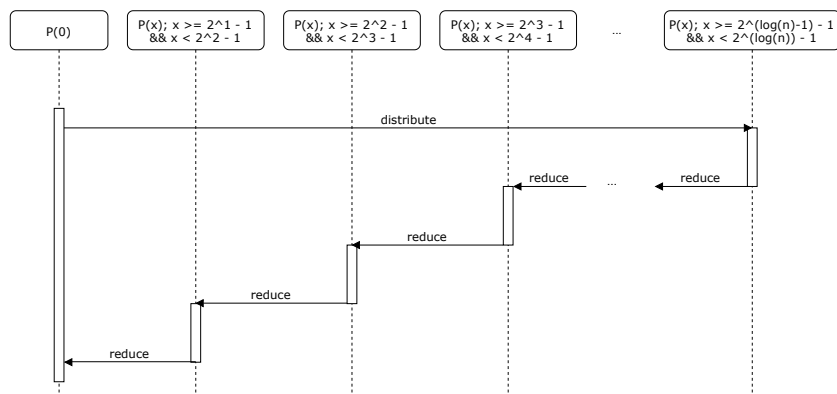
6. Výpis zoradenej postupnosti – $O(n)$

Celkovo:

$t(n) = O(n), p(n) = O(\log_2 n), c(n) = O(n * \log_2 n)$ – čo je optimálne.

4 Komunikačný protokol

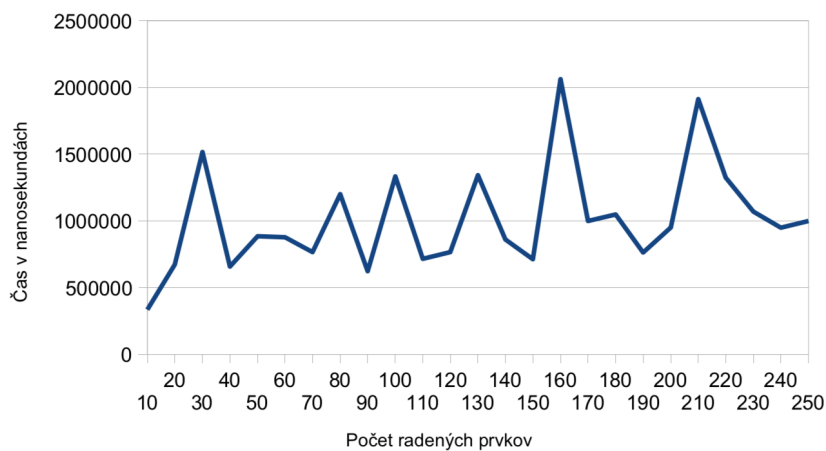
Na obrázku 1 je znázornený komunikačný protokol medzi n procesormi. Pre lepšiu prehľadnosť každý stĺpec znázorňuje všetky procesory jednej úrovne stromu procesorov.



Obr. 1: Komunikačný protokol v aplikácii

5 Experimenty a testovanie

Testovanie časovej zložitosti bolo vykonané so vstupnými sekvenciami od dĺžky 10 do dĺžky 250. Štatistická relevancia bola zaistená spustením algoritmu s každou dĺžkou vstupu 10-krát. Výsledky merania sú znázornené v grafe 2.



Obr. 2: Výsledky merania časovej zložitosti algoritmu

6 Záver

Aj napriek násobnému meraniu časovej zložitosti výsledky pomerne kolísajú, čo môže byť pripísané nerovnomernému vyťaženiu serveru merlin, kde bolo meranie vykonané. Namerané dáta však vykazujú lineárnu závislosť, ktorá odpovedá odvodennej teoretickej zložitosti.