

Paralelní a distribuované algoritmy

Pipeline Merge Sort

Autor: Dominik Švač (xsvacd00)

1. Popis algoritmu

Pipeline Merge Sort je paralelný algoritmus. Je to riadiaci algoritmus, ktorý využíva $\log_2(n) + 1$ procesorov. Procesory sú lineárne prepojené pomocou dvoch liniek a obsahujú zoradené postupnosti hodnôt. Výnimkou je prvý procesor, ktorý má iba jednu linku vstup a posledný procesor, ktorý má tiež iba jednu linku, na ktorej sa bude nachádzať výsledná zoradená postupnosť hodnôt.

2. Analýza algoritmu

Algoritmus začína, ak procesor P_i má na prvej linke aspoň 2^{i-2} prvkov a na druhej musí byť aspoň jeden prvok. Z toho vyplýva, že procesor P_i začína proces radenia v cykle:

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

kým nespracuje $n - 1$ hodnôt a zastaví v cykle: $(n - 1) + 2^{i-1} + i - 1$
Keď posledný procesor P_r zastaví, tak tým je radenie dokončené v cykle:

$$n + 2^r + r - 1 = 2n + \log_2 n - 1$$

Priestorová náročnosť tohto algoritmu je $p(n) = \log_2 n + 1$, časová náročnosť je $t(n) = O(n)$ a celková cena $c(n)$ je daná:

$$c(n) = t(n) * p(n) = O(n) * (\log_2 n + 1) = O(n * \log_2 n)$$

3. Implementácia

Algoritmus bol implementovaný v jazyku C++ s knižnicou *Open MPI*. Vstupom programu je súbor s názvom *numbers*, ktorý obsahuje čísla, tie sa načítajú a zoradia. Ako už bolo spomenuté, tento algoritmus využíva $\log_2(n) + 1$ procesorov. Na začiatku prvý procesor načíta súbor s číslami a pošle ich nasledujúcemu procesoru.

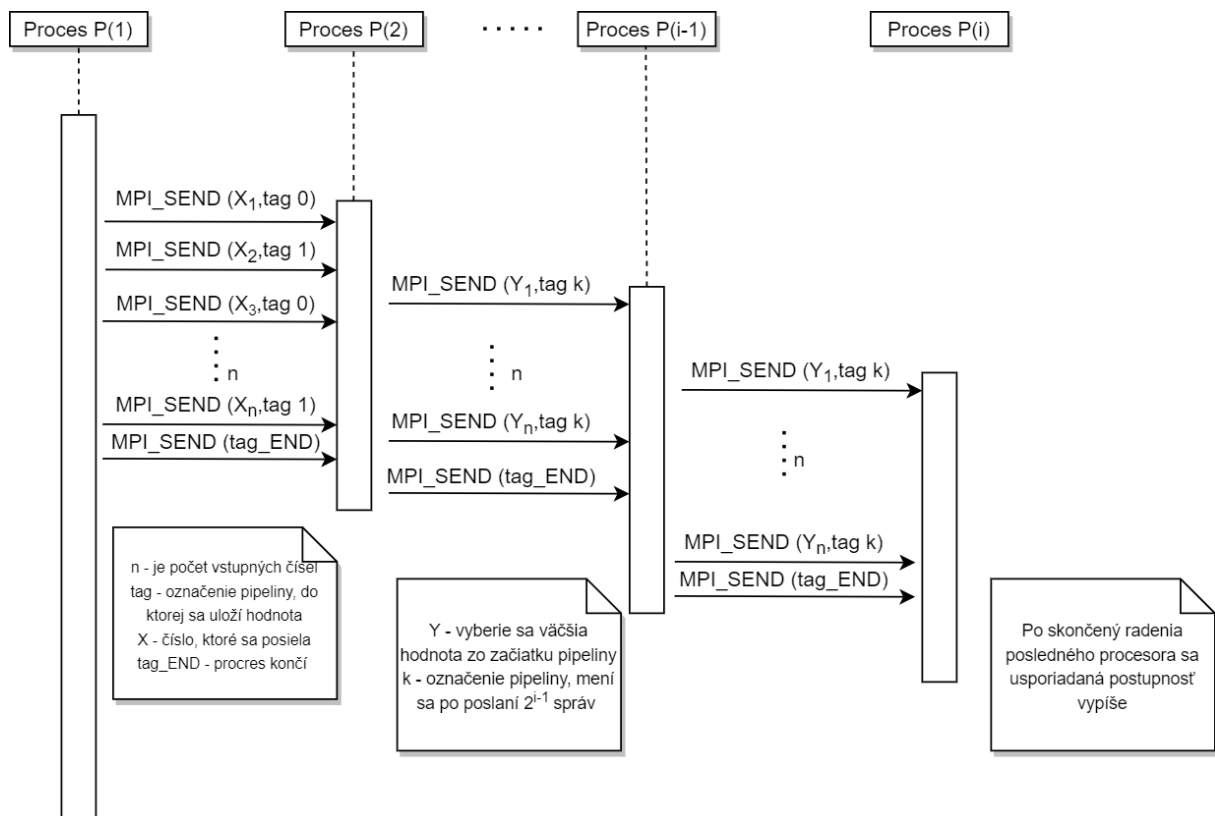
Každý procesor si udržiava dva lokálne zoznamy. Procesy po obdržaní správy si uložia hodnotu na koniec zoznamu, podľa príznaku TAG, ktorý určuje, do ktorého zoznamu má byť uložená daná hodnota. Ak sa naplní požadované množstvo hodnôt v jednotlivých zoznamoch (2^{i-2} prvkov v prvom zozname a aspoň jeden v druhom), začína sa porovnávanie. Vyberie sa väčšia hodnota zo začiatku zoznamov a tá sa pošle nasledujúcemu procesoru spolu s príznakom TAG. Z každého zoznamu sa tak odošle ($2^{i-2}/2$) hodnôt s rovnakou hodnotou TAG. Následne sa hodnota TAG zmení (počíta sa ako TAG mod 2).

Toto sa opakuje kým procesor neobdrží správu s príznakom END_TAG. Potom už nové hodnoty neprijímajú. Dokončí sa odosielanie hodnôt, ktoré sú ešte uložené v zoznamoch a procesor skončil svoju prácu.

Posledný procesor pracuje kúsok odlišne. Namiesto toho, aby odosielať usporiadané hodnoty, ďalej ukladá si ich do zoznamu. Na konci už len vypíše tento zoznam.

4. Komunikačný protokol

Procesory v tomto algoritme komunikujú pomocou správ. Posiela sa hodnota a TAG, ktorý určuje do ktorého zoznamu (Pipeline) sa uloží posiadaná hodnota. Prvý proces končí až keď mu príde správa od posledného procesu, že sa algoritmus radenia skončil. Na samotné radenie to vplyv nemá, ale na prvom procese sa počíta dĺžka trvania algoritmu. Detailnejšie je komunikácia popísaná v nasledujúcom sekvenčnom diagrame.



5. Záver

Práca s knižnicou MPI sa mi páčila. Paralelný algoritmus Pipeline merge sort funguje rýchlo. Tento algoritmus som implementoval a testoval na školskom servery Merlin.