Předmět: Dokumentace 2. projektu do předmětu PRL 2015/2016
Autor: Daniel Žůrek
Login: xzurek12

1. Rozbor a analýza algoritmu

Minimum Extraction sort je paralelní řadící algoritmus pro seřazení vstupní posloupnosti délky n, na řazení se tedy podílí více procesorů najednou. Topologie procesorů je binární strom, který je tvořen $(\log n) + 1$ úrovněmi, n listy a celkový počet procesorů je $2n - 1$.

Každý listový procesor (uzel) obsahuje jeden řazený prvek. Každý nelistový procesor porovná hodnoty svých dvou synů a hodnotu menšího z nich si uloží.

Do kořenového procesoru se nejmenší z řazených prvků dostane po $(\log n) + 1$ krocích, přičemž kořen potřebuje jeden krok na porovnání a jeden krok na uložení výsledku. Následně do kořenového procesoru s každým dalším krokem putují nové nejmenší hodnoty a každý ze zbylých $n - 1$ prvků potřebuje dva kroky. Algoritmus seřadí vstupní posloupnost v $n + \log n$ krocích. Jedná se tedy o algoritmus s lineární časovou složitostí.

Počet procesorů:

$$p(n) = 2.n - 1$$

Časová složitost:

$$t(n) = 2.(n - 1) + (\log n) + 1 = \Theta(n)$$

Celková cena:

$$p(n).t(n) = \Theta(n^2)$$

2. Popis implementace

Základ představuje struktura `Node`, která obsahuje proměnné popisující stav jednoho procesoru (rank (id) levého a pravého syna, rank procesu atd.). Začátek programu je tvořen inicializací této struktury, zjištěním ranku prvního listového procesoru pomocí funkce `getFirstLeaf()` a zjištěním ranku levého a pravého syna pro daný procesor.

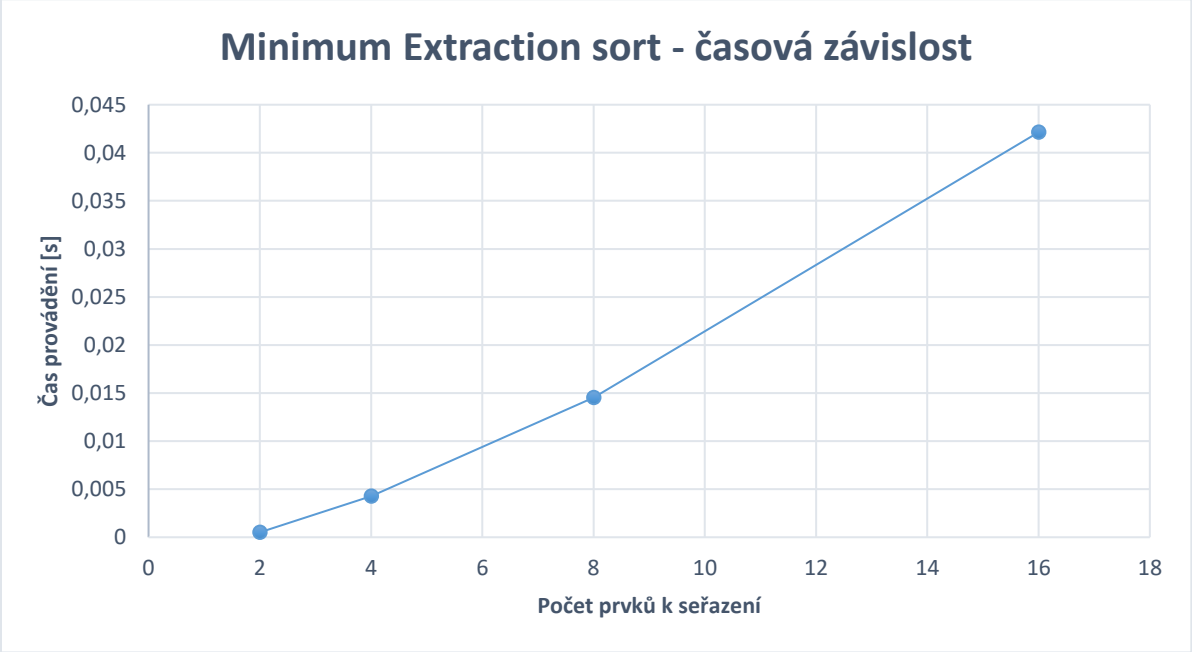
Procesor s rankem 0 (kořen stromu) si otevře vstupní soubor, který byl vytvořen utilitou `dd`, a začne načítat hodnoty vstupní posloupnosti. Každou načtenou hodnotu odešle pomocí funkce `MPI_Send()`, funkce knihovny Open MPI, na jeden z listových procesorů. Pokud bylo načteno méně hodnot, než je listových procesorů, je zbylým procesorům zaslána neutrální hodnota, která se neobjeví v seřazené posloupnosti. Tato hodnota je reprezentována globální proměnnou `NEUTRAL_VALUE`. Ostatní vytvořené procesory (procesy) čekají pomocí blokující funkce `MPI_Recv()`, dokud neobdrží svou příslušnou hodnotu. Řazení prvků probíhá v cyklu, dokud se v proměnné `stopFlag` obsažené ve struktuře `Node` neobjeví hodnota `STOP_CYCLE`.

Každý nelistový procesor očekává zprávu od svých dvou synů a následně je provedena série podmínek, která zajistí vybrání menšího ze dvou prvků. Synům jsou poté zaslány jejich nové hodnoty, funkce `setNewValue()`, v případě menšího prvku `NEUTRAL_VALUE`, u většího prvku je zaslána jeho původní hodnota. Procesory s rankem větším než 0 vypočtou rank svých otců a na tyto procesory je zaslána jejich hodnota. Tisk hodnoty na výstup, provádí procesor s rankem 0, obsahuje-li korektní hodnotu.

3. Experimenty

Experimenty byly prováděny na systému Manjaro 4.1.15-1, kde každé měření bylo provedeno 10x a výsledek zprůměrován. Parametry testů byly:

./test.sh 2 3 - 2 prvky, 3 procesory
./test.sh 4 7 - 4 prvky, 7 procesorů
./test.sh 8 15 - 8 prvků, 15 procesorů
./test.sh 16 31 - 16 prvků, 31 procesorů



Graf 1: Časová závislost na počtu řazených prvků

4. Komunikační protokol

5. Závěr

Funkčnost projektu byla ověřena na serveru Merlin.