

IPR
Projekt 2
-
Enumeration Sort

Zadání

Pomocí knihovny Open MPI implementujte algoritmus Enumeration Sort s lineárním polem procesorů.

Vstupem programu *es* je soubor *numbers*, který obsahuje seznam čísel, každé o velikosti 1B.

Výstupem programu je pak na prvním řádku vstupní posloupnost čísel tak, jak je programem načítána, a poté sestupně seřazená posloupnost čísel (každé na jeden řádek).

Součástí řešení projektu je i shellový skript *test*, který přijímá jeden argument (počet hodnot) a který je posléze schopný vytvořit soubor nahodných čísel a následně je seřadit pomocí programu *es*.

Rozbor algoritmu

Vlastní algoritmus je rozdělen do několika fází:

1. V první fázi se všechny registry C nastaví na hodnotu 1
2. Následně se v cyklu ($1 - n$, kde n je počet hodnot) provede načtení vstupu, rozešle se vstupní číslo konkrétnímu procesoru, provede se porovnání hodnot a posun registru Y vpravo.
3. V třetí fázi dochází k posunu registru Y v době, kdy byly již všechny hodnoty ze vstupu načtené a provedou se poslední porovnání hodnot.
4. Ve čtvrté fázi dojde k rozeslání vstupních hodnot jednotlivým procesorům dle stavu registru C. Posloupnost je již seřazená.
5. V poslední páté fázi se obsah registru posouvá vpravo, resp. Rotuje do procesoru 0 (“master”) v případě posledního procesoru. Master pak výslednou sestupně seřazenou posloupnost vypíše.

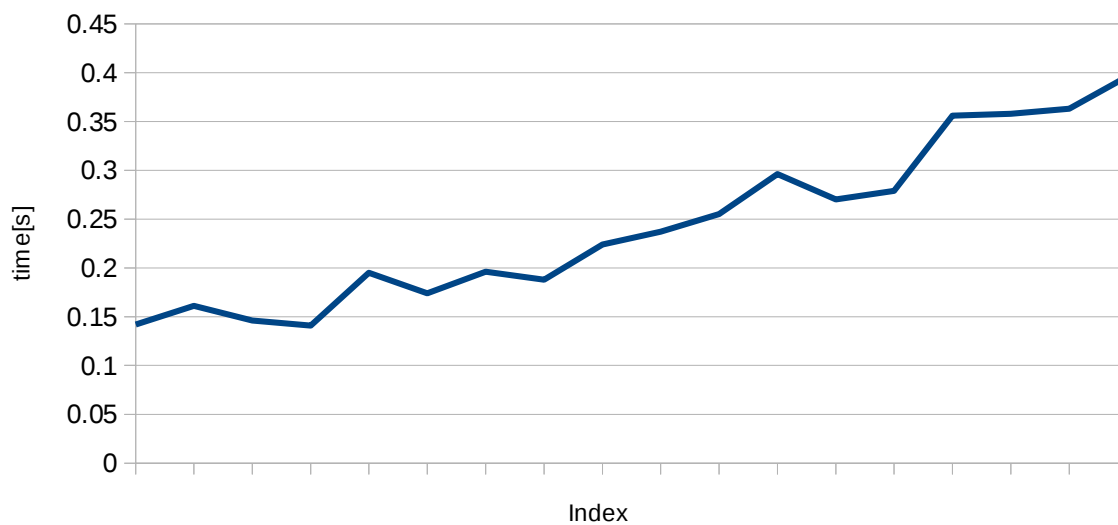
V rámci implementace bylo také navrženo rozšíření porovnávání čísel, které mělo v základu problém s řazením stejně velkých čísel. Každá vstupní hodnota je vynásobena číslem 100 a posléze je k ní přičten index hodnoty (pořadí čísla). Při vypisování se pak číslo celočíselně vydělí číslem 100.

Složitost a testování

Teoretická složitost algoritmu je $O(n)$, avšak implementace dle výsledků testování vylučuje spíše exponenciální složitost.

Testování bylo provedeno získáním několika vzorků pro posloupnosti délek 2 - 20 a následně zprůměrováno. Pro zjištění časové náročnosti implementace bylo použito příkazu *time*.

Enumeration sort time complexity



Závěr

Zadání se podařilo splnit, avšak algoritmus vykazuje o něco vyšší časovou složitost, což může být způsobeno mírnými změnami oproti algoritmu původnímu (přidání několika smyček).