Dokumenácia k 2. projektu z PRL

Merge-splitting sort

Matej Marušák, Marec 2018

1 Úvod

V tejto správe je popísaná aplikácia vytvorená v rámci projektu PRL v akademickom roku 2017/18. V kapitole 2 je úvod do riešenej problematiky, nasledovaný popisom aplikácie v kapitole 3.

2 Merge-splitting sort

Merge-splitting sort je paralelný algoritmus na zoradenie množiny prvkov. Obsahuje pole procesorov, ktorých je menej ako počtu prvkov (v prípade rovnosti počtu procesorov a prvkov sa jedná o algoritmus odd-even trasposition sort). Nakoľko je procesorov menej ako prvkov, každý procesor sa stará o viacero prvkov.

Algoritmus začína rozdelením prvkov pre každý procesor. V ideálnom prípade má každý procesor rovnako prvkov, v najhoršom je maximálny rozdiel prvkov medzi dvoma procesormi 1. Každý procesor si svoje prvky zoradí optimálnym sekvenčným algoritmom. Následne sa vykoná $p/2$ kôl, kde v každom kole sa vykonajú nasledujúce operácie:

1. Párne procesory vykonajú nasledujúcich 5 akcií za sebou:
   (a) Počkajú na všetky hodnoty od pravého (nepárneho) suseda
   (b) Vykonajú spojenie dvoch zoradených postupností do jednej (svoje a susedove čísla)
   (c) Spodnú časť čísel si nechajú a hornú pošlú späť pravému susedovi
   (d) Pošlú všetky svoje prvky ľavému susedovi
   (e) Príjmu od ľavého suseda prvky, ktoré si uložia ako svoje hodnoty
2. Nepárne procesory vykonajú nasledujúcich 5 akcií za sebou:
   (a) Pošlú všetky svoje prvky ľavému susedovi
   (b) Príjmu od ľavého suseda prvky, ktoré si uložia ako svoje hodnoty
   (c) Počkajú na všetky hodnoty od pravého (párneho) suseda
   (d) Vykonajú spojenie dvoch zoradených postupností do jednej (svoje a susedové čísla)
   (e) Spodnú časť čísel si nechajú a hornú pošlú späť pravému susedovi

Po skončení tohto cyklu sú všetky hodnoty zoradené.

2.1 Teoretická zložitosť

Na výpočet teoretickej zložitosti si rozložíme algoritmus na 4 časti:
$m = \frac{n}{p}$ *teda počet prvkov v jednom procesore*

- Počiatočné zoradenie prvkov optimálnym sekvenčným algoritmom $\rightarrow O(m * log(m))$
- Prenos m prvkov do i-teho procesoru $\rightarrow O(m)$
- Spojenie dvoch postupností do jednej optimálnym algoritmom $\rightarrow 2m$
- Spätný prenos časti zoradenej postupnosti $\rightarrow O(m)$

Po sčítaní je to $O(m * log(m)) + 3 * O(m) = O(m * log(m)) + O(m)$
Vieme že $m = n/p$ tak môžme dosadiť a vynásobiť počtom procesorov p.
$c(n) = O(n * log(n) + O(n * p)$ čo je pre $p <= log(n)$ optimálne.

3 Aplikácia

Pre spúšťanie aplikácie bol vytvorený skript `test.sh`, ktorý aplikáciu preloží, pripraví vstupný súbor a spustí aplikáciu. Skript sa spúšťa s dvoma argumentami a to `počet_hodnôt` a `počet_procesorov`.

Samotná aplikácia sa spúšťa s jedným argumentom a to s počtom prvkov. Tento počet prvkov bude prečítaný zo súbora `numbers`.

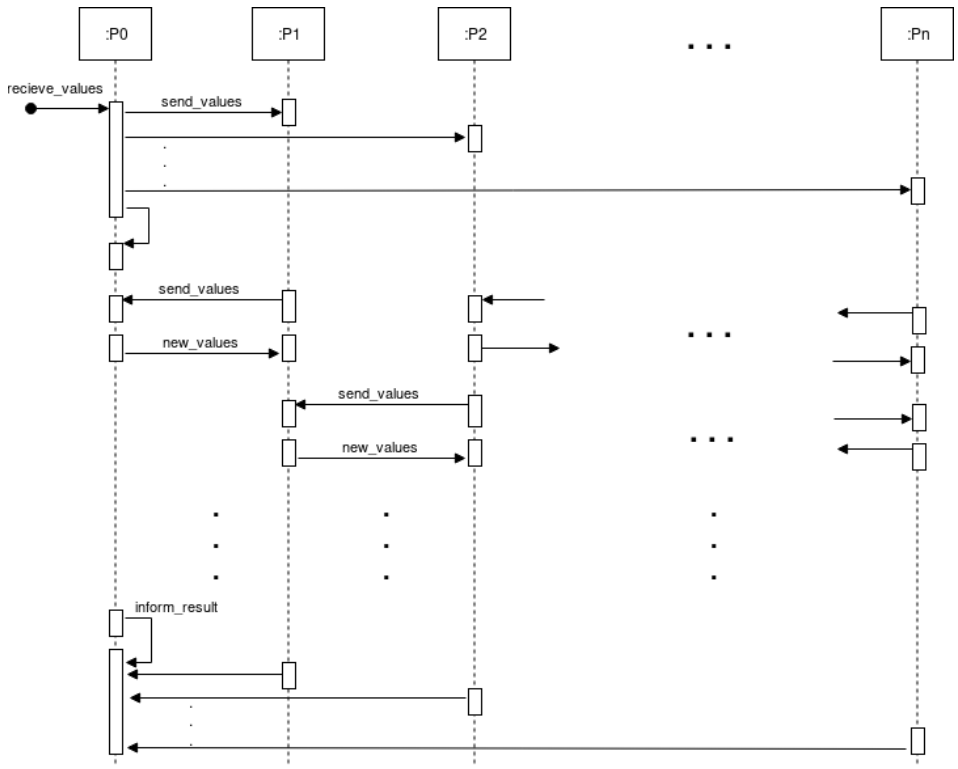
Prvý krok, ktorý každý procesor vykoná, je zistenie si počtu prvkov, ktoré má vyžadovať. Toto sa robí podľa jednoduchého pravidla - každý procesor pracuje s $\frac{pocet\_prvkov}{pocet\_procesorov}$ prvkov a ak je jeho pozícia menšia ako zvyšok po delení daného zlomku, tak očakáva o jeden prvok viac.

Následne prvý procesor číta súbor a posiela hodnoty každému procesoru zvlášť. Ostatné procesory čakajú, až kým nedostanú požadovaný počet prvkov a potom vykonávajú smyčku, ktorá je popísaná v kapitole 2. Prvý procesor po prečítaní hodnôt sa správa ako akýkoľvek iný procesor.

Po skončení smyčky všetky procesory pošlú svoje hodnoty procesoru 0, ktorý ich postupne zbiera a vypisuje na štandardný výstup.

3.1 Komunikačný protokol

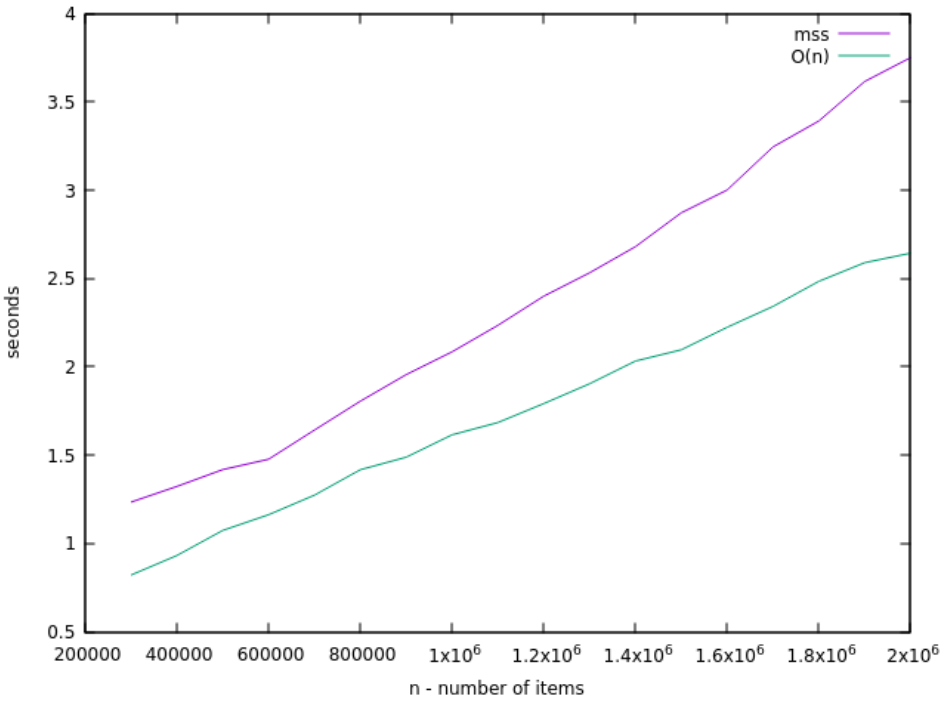
Na obrázku 1 je znázornený protokol komunikácie medzi procesormi. Jedná sa o obecný model pre n procesorov. Každá skupina rovnakých metód je zapísaná len raz pre lepšiu prehľadnosť.



Obr. 1: Komunikačný protokol v aplikácii

3.2 Experimentálne výsledky

S výslednou aplikáciou bolo experimentované aj v sledovaní času, ktorý je potrebný na zoradenie prvkov. Obrázok 2 ukazuje ako sa správa aplikácia pri stúpajúcom počte prvkov. Pre porovnanie je v grafe aj vyobrazené ako dlho trvá len prejsť cez dané pole prvkov. Experimenty boli vykonané na 4och procesoroch.



Obr. 2: Časové trvanie behu v porovnaní s O(n)

4 Záver

V tejto správe bola popísaná aplikácia na paralelné radenie poľa prvkov. Aplikácia funguje podľa očakávaní, jej funkcionalita bola overená na veľkej sade testov a boli zmerané časové výhody danej aplikácie, ktoré sú prezentované v grafe v kapitole 3.2.