Semestralni projekt MI-PDP 2020/2021:

Paralelni algoritmus pro reseni problemu

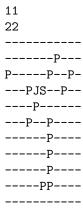
$\label{eq:martin Safránek} {\it Martin Šafránek}$ magisterske studijum, FIT CVUT, Thakurova 9, 160 00 Praha 6 ${\it April 22, 2021}$

1 Definice problemu a popis sekvencniho algoritmu

Program řeší problém nalezení optimální posloupnosti tahů pro střelce a jezdce, která vede k sebrání všech pěšců rozmístěných na šachovnici. Jedná se o analogii problému obchodního cestujícího. Nalezení optimálního řešení je proto NP těžký úkol. Řešení v této práci používá bruteforce s heuristikami pro ořezávání stavového prostoru.

Příklad vstupu je na obrázku 1. Obsahuje vždy

- 1. přirozené číslo k, reprezentující délku strany šachovnice S o velikosti $k \times k$,
- 2. pole souřadnic rozmístěných figurek na šachovnici S,
- 3. horní mez délky optimální posloupnosti d_{max}^* .



Obrázek 1: Příklad vstupních dat pro k=11, $d_{max}^*=22$. Střelec je označen S, jezdec J, pěšák P a prázdné políčko -.

Sekvenční algoritmus je popsán v 1. Používá dvě heuristiky.

Heuristika střelec. Z množiny možných políček, kam je možné střelce přemístit jsou preferována ty, která obsahují pěšce. Pokud takové políčko neexistuje, jsou preferována políčka s alespoň jedním pěšákem na diagonále. Jinak se pohyb střelce rozhodne náhodně.

Heuristika kůň. Z množiny možných políček, kam je možné koně přemístit jsou preferována ty, která obsahují pěšce. Pokud takové políčko neexistuje, jsou preferována políčka, z kterých kůň ve svém následujícím tahu může vzít pěšáka. Pokud ani takové políčko neexistuje, jsou preferováno políčka, z kterých kůň v následujícíh dvou tazích může vzít pěšáka. Jinak se pohyb koně rozhodne náhodně.

```
input: k \times k pole, mez d_{max}^*
   output: optimální posloupnost tahů
 1 if abc or def then
2 else
3 ;
4 end
5 while While condition do
      instructions:
      if condition then
 7
          instructions1;
          instructions2;
 9
      else
10
11
         instructions3;
12
      end
13 end
```

Algoritmus 1: sekvenční

2 Popis paralelniho algoritmu a jeho implementace v OpenMP - taskovy paralelismus

Popiste paralelni algoritmus, opet vyjdete ze zadani a presne vymezte odchylky, ktere pri implementaci OpenMP pouzivate. Popiste a vysvetlete strukturu celkoveho paralelniho algoritmu na urovni procesuu v OpenMP a strukturu kodu jednotlivych procesu. Napr. jak je naimplemtovana smycka pro cinnost procesu v aktivnim stavu i v stavu necinnosti. Jake jste zvolili konstanty a parametry pro skalovani algoritmu. Struktura a semantika prikazove radky pro spousteni programu.

3 Popis paralelniho algoritmu a jeho implementace v OpenMP - datovy paralelismus

Popiste paralelni algoritmus, opet vyjdete ze zadani a presne vymezte odchylky, ktere pri implementaci OpenMP pouzivate. Popiste a vysvetlete strukturu celkoveho paralelniho algoritmu na urovni procesuu v OpenMP a strukturu kodu jednotlivych procesu. Napr. jak je naimplemtovana smycka pro cinnost procesu v aktivnim stavu i v stavu necinnosti. Jake jste zvolili konstanty a parametry pro skalovani algoritmu. Struktura a semantika prikazove radky pro spousteni programu.

4 Popis paralelniho algoritmu a jeho implementace v MPI

Popiste paralelni algoritmus, opet vyjdete ze zadani a presne vymezte odchylky, zvlaste u Master-Slave casti. Popiste a vysvetlete strukturu celkoveho paralelniho algoritmu na urovni procesuu v MPI a strukturu kodu jednotlivych procesu. Napr. jak je naimplemtovana smycka pro cinnost procesu v aktivnim stavu i v stavu necinnosti. Jake jste zvolili konstanty a parametry pro skalovani algoritmu. Struktura a semantika prikazove radky pro spousteni programu.

5 Namerene vysledky a vyhodnoceni

- 1. Zvolte tri instance problemu s takovou velikosti vstupnich dat, pro ktere ma sekvencni algoritmus casovou slozitost mezi 1 a 10 minutami. Pro mereni cas potrebny na cteni dat z disku a ulozeni na disk neuvazujte a zakomentujte ladici tisky, logy, zpravy a vystupy.
- 2. Merte paralelni cas pri pouziti $i=2,\cdot,60$ vypocetnich jader.
- 3. Tabulkova a pripadne graficky zpracovane namerene hodnoty casove slozitosti měernych instanci behu programu s popisem instanci dat. Z namerenych dat sestavte grafy zrychleni S(n, p).

4. Analyza a hodnoceni vlastnosti paralelniho programu, zvlaste jeho efektivnosti a skalovatelnosti, pripadne popis zjisteneho superlinearniho zrychleni.

6 Zaver

Celkove zhodnoceni semestralni prace a zkusenosti ziskanych behem semestru.

7 Literatura