Semestrální projekt MI-PDP 2020/2021:

Paralelní algoritmus pro řesení problému prohledávání stavového prostoru

Martin Šafránek

magisterské studijum, FIT ČVUT, Thákurova 9, 160 00 Praha 6

zdrojové kódy:
`https://github.com/TaIos/ni-pdp-semestralka`

2. května 2021

1 Definice problému a popis sekvenčního 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.

1.1 Popis vstupu

Příklad vstupu je uveden na obrázku 1. Obsahuje popořadě vždy

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

```
11
22
-----------
-------P---
P-----P--P-
---PJS--P--
----P------
---P--P----
------P----
------P----
------P----
-----PP----
-----------
```

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

1.2 Heuristiky

Sekvenční algoritmus používá dvě heuristiky pro pohyb střelce a koně.

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 následujících dvou tazích může vzít pěšáka. Jinak se pohyb koně rozhodne náhodně.

1.3 Pseudokód

```
   input  : k × 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
 6 |  instructions;
 7 |  if condition then
 8 |  |  instructions1;
 9 |  |  instructions2;
10 |  else
11 |  |  instructions3;
12 |  end
13 end
```

Algoritmus 1: sekvenční

2 Popis paralelního algoritmu a jeho implementace v OpenMP - taskový 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.

Taskový paralelní algoritmus je naimplementován pomocí OpenMP. Hlavní rozdíl oproti sekvenčnímu algoritmu popsaném v je rozdělení úlohy prohledávání stavového prostoru na tasky. Task je základní jednotka, kterou je OpenMP schopno přidělit vláknu a provést tak výpočet. Pro zadanou úlohu task znamená šachovnici s pozicí všech figurek a historií tahů. Takto vytvořené tasky OpenMP přidává do svého taskpoolu, z kterého si je vlákna vyzvedávají a řeší. Dále všechny vlákna řešící tasky z taskpoolu sdílejí nejlepší řešení d_{best}. Heuristiky jsou totožné jako v podsekci 1.1.

2.1 Konstanty a parametry pro škálování algoritmu

Taskový paralelní algoritmus implementovaný pomocí OpenMP umožňuje nastavení konstant, které ovlivní logiku funkce programu a tedy i výpočetní čas. Změněny byly pouze zde zmíněné konstanty. Jejich hodnota byla určena empiricky na vstupních datech. Nejedná se o optimální hodnoty, protože jejich nalezení je stejně těžký problém jako nalezení optimální cesty v původním problému.

Konstanta `TASK_THRESHOLD`. Pokud vlákno řeší instanci a délka její cesty je delší než `TASK_THRESHOLD`, nevytváří další OpenMP tasky a nepřidává je do taskpoolu. Zadanou instaci vyřeší použitím sekvenčního algoritmu popsaném v sekci .

název	hodnota
`TASK_THRESHOLD`	6

Tabulka 1: Konstanty použité v OpenMP taskovém paralelismu.

2.2 Pseudokód

```
   input  : k × 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
 6 |  instructions;
 7 |  if condition then
 8 |  |  instructions1;
 9 |  |  instructions2;
10 |  else
11 |  |  instructions3;
12 |  end
13 end
```

Algoritmus 2: OpenMP taskový paralelismus

3 Popis paralelniho algoritmu a jeho implementace v OpenMP - datový 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.

3.1 Konstanty a parametry pro škálování algoritmu

Leberkas ham bacon, pastrami turducken pork belly cupim salami kielbasa doner. Turkey cupim meatball capicola jowl cow shank chicken drumstick kevin salami swine pork belly. Drumstick leberkas corned beef beef short loin boudin. Turkey strip steak bacon, ball tip sirloin pork loin pork.

3.2 Pseudokód paralelního algoritmu — datový paralelismus

Porchetta andouille flank kielbasa. Tail biltong turducken porchetta burgdoggen ground round shoulder ham, hamburger bacon shankle landjaeger fatback pork belly doner. Sirloin doner venison shankle cow, hamburger flank sausage pork belly. Tenderloin venison pancetta corned beef tongue cow pork belly capicola ball tip salami short ribs. Sirloin rump andouille tail shank fatback bresaola.

Leberkas ham bacon, pastrami turducken pork belly cupim salami kielbasa doner. Turkey cupim meatball capicola jowl cow shank chicken drumstick kevin salami swine pork belly. Drumstick leberkas corned beef beef short loin boudin. Turkey strip steak bacon, ball tip sirloin pork loin pork.

3.3 Pseudokód

```
   input  : k × 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
 6 |  instructions;
 7 |  if condition then
 8 |  |  instructions1;
 9 |  |  instructions2;
10 |  else
11 |  |  instructions3;
12 |  end
13 end
```

Algoritmus 3: OpenMP datový paralelismus

4 Popis paralelního 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.

4.1 Konstanty a parametry pro škálování algoritmu

Leberkas ham bacon, pastrami turducken pork belly cupim salami kielbasa doner. Turkey cupim meatball capicola jowl cow shank chicken drumstick kevin salami swine pork belly. Drumstick leberkas corned beef beef short loin boudin. Turkey strip steak bacon, ball tip sirloin pork loin pork.

4.2 Pseudokód

```
   input  : k × 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
 6 |  instructions;
 7 |  if condition then
 8 |  |  instructions1;
 9 |  |  instructions2;
10 |  else
11 |  |  instructions3;
12 |  end
13 end
```

Algoritmus 4: MPI paralelismus

5 Naměřené výsledky a vyhodnocení

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 Závěr

Celkove zhodnoceni semestralni prace a zkusenosti ziskanych behem semestru.

Porchetta andouille flank kielbasa. Tail biltong turducken porchetta burgdoggen ground round shoulder ham, hamburger bacon shankle landjaeger fatback pork belly doner. Sirloin doner venison shankle cow, hamburger flank sausage pork belly. Tenderloin venison pancetta corned beef tongue cow pork belly capicola ball tip salami short ribs. Sirloin rump andouille tail shank fatback bresaola.

Leberkas ham bacon, pastrami turducken pork belly cupim salami kielbasa doner. Turkey cupim meatball capicola jowl cow shank chicken drumstick kevin salami swine pork belly. Drumstick leberkas corned beef beef short loin boudin. Turkey strip steak bacon, ball tip sirloin pork loin pork.

Doner tenderloin chislic rump shank sausage beef andouille. Frankfurter boudin strip steak ribeye, venison biltong beef. Drumstick ham hock pork chop buffalo rump. Beef ribs sausage filet mignon pork chop flank jerky.

7 Skript pro spouštění a sekvenčního a OpenMP řešení

Bash skript v 2 spustí program, kterému v argumentu předá vstupní data jako textový soubor. Formát vstupních dat viz podsekce 1. Pro sekvenční řešení skript spouští všechny řešení najednou. Pro paralelní čeká vždy na dokončení běhu aktuálního vstupu.

```
#!/usr/bin/bash

PROGRAM='./run.out'
OUT="out/parallel/data_epoch3_guided"

mkdir "$OUT"

for filename in ../data/*.txt; do
base=$(basename -- "$filename")
echo "$base"

# sekvencni
#  $PROGRAM "$filename" >"$OUT/$base" &

# paralelni
$PROGRAM "$filename" > "$OUT/$base"

done
```

Obrázek 2: Bash skript pro spuštění sekvenčního a OpenMP programu.

8 Literatura

Tenderloin pork belly ham leberkas doner rump. Filet mignon beef pastrami pork belly drumstick. Beef ribs filet mignon porchetta pork turducken spare ribs tri-tip corned beef strip steak turkey capicola. Venison hamburger ball tip, buffalo fatback pork alcatra doner pork belly.