

Semestrální projekt MI-PAR 2010/2011

Paralelní algoritmus pro řešení problému

Jaroslav Líbal

Martin Venuš

FIT ČVUT

magisterské studium

Kolejní 550/2, 160 00 Praha 6

29. listopadu 2010

Obsah

1	Definice problému a popis sekvenčního algoritmu	3
1.1	Vstupní data	3
1.2	Úkol	3
1.3	Výstup algoritmu	3
1.4	Sekvenční algoritmus	3
1.5	Paralelní algoritmus	4
1.6	Výhledově odstranit	4
2	Popis paralelního algoritmu a jeho implementace v MPI	4
3	Naměřené výsledky a vyhodnocení	4
3.1	Sekvenční řešení	5
3.2	Paralelní řešení	5
4	Závěr	6
5	Literatura	7
A	Návod pro vkládání grafů a obrázků do Latexu	7

1 Definice problému a popis sekvenčního algoritmu

Úloha MBG: minimální obarvení grafu

1.1 Vstupní data

$G(V, E)$ = jednoduchý neorientovaný neohodnocený k -regulární graf o n uzlech a m hranách

n = přirozené číslo představující počet uzlů grafu G , $n \geq 5$ k = přirozené číslo řádu jednotek představující stupeň uzlu grafu G , $n \geq k \geq 3$; n a k nejsou současně obě liché

Doporučení pro generování G :

Použijte generátor grafu s volbou typu grafu "-t REG", který vygeneruje souvislý neorientovaný neohodnocený graf.

1.2 Úkol

Nalezněte minimální uzlové obarvení grafu G , či-li zobrazení $B: V \rightarrow N$ takové, že $b = |\{B[x]; x \in V\}|$ je minimální. Řešení existuje vždy, nemusí být jednoznačné.

1.3 Výstup algoritmu

Počet použitých barev a výpis uzlového obarvení $B[1..n]$ grafu G ve formě matice sousednosti (barvy uzlů píše na diagonálu matice).

1.4 Sekvenční algoritmus

Sekvenční algoritmus je typu BB-DFS s omezenou hloubkou prohledávaného prostoru. Cena, kterou minimalizuje, je počet barev b . Mezistav je dán částečným obarvením. Mezistav či koncový stav je přípustný, jestliže žádné 2 sousední uzly nejsou obarvené stejnou barvou. Návrat v mezistavu provádíme, pokud nelze žádný neobarvený uzel obarvit, aniž by se porušila podmínka uzlového

barvení.

Triviální horní mez na $b = k + 1$ (každý k -regulární graf lze uzlově obarvit $k + 1$ barvami).

Těsná spodní mez na $b = 2$. Pokud je G bipartitní, existuje obarvení dvěma barvami.

1.5 Paralelní algoritmus

Paralelní algoritmus je typu G-PBB-DFS-D.

1.6 Výhledově odstranit

Popište problém, který váš program řeší. Jako výchozí použijte text zadání, který rozšiřte o přesné vymezení všech odchylek, které jste vůči zadání během implementace provedli (např. úpravy heuristické funkce, organizace zásobníku, apod.). Zmiňte i případně i takové prvky algoritmu, které v zadání nebyly specifikovány, ale které se ukázaly jako důležité. Dále popište vstupy a výstupy algoritmu (formát vstupních a výstupních dat). Uveďte tabulku naměřených časů sekvenčního algoritmu pro různě velká data.

2 Popis paralelního algoritmu a jeho implementace v MPI

Popište paralelní algoritmus, opět vyjděte ze zadání a přesně vymezte odchylky, zvláště u algoritmu pro vyvažování zátěže, hledání dárce, či ukončení výpočtu. Popište a vysvětlete strukturu celkového paralelního algoritmu na úrovni procesů v MPI a strukturu kódu jednotlivých procesů. Např. jak je naimplementována smyčka pro činnost procesů v aktivním stavu i v stavu nečinnosti. Jaké jste zvolili konstanty a parametry pro škálování algoritmu. Struktura a sémantika příkazové řádky pro spouštění programu.

3 Naměřené výsledky a vyhodnocení

Pro měření výsledků byly vygenerovány následující matice:

REG = k -regularni graf

Matice	Typ matice	Počet vrcholů	Stupeň vrcholu
Matice 1	REG	42	18
Matice 2	REG	44	29
Matice 3	REG	43	12

Tabulka 1: Popis jednotlivých vygenerovaných matic.

3.1 Sekvenční řešení

Typ matice	čas 1 [s]	čas 2 [s]	čas 3 [s]	prům. čas [s]
Matice 1	364.7	364.9	365.0	364.86
Matice 2	652.3	651.5	651.4	651.7
Matice 3	1184.7	1184.0	1184.9	1184.53

Tabulka 2: Doba běhu sekvenčního algoritmu pro jednotlivé matice.

3.2 Paralelní řešení

1. Zvolte tři instance problému s takovou velikostí vstupních dat, pro které má sekvenční algoritmus časovou složitost kolem 5, 10 a 15 minut. Pro měření čas potřebný na čtení dat z disku a uložení na disk neuvažujte a zakomentujte ladící tisky, logy, zprávy a výstupy.
2. Měřte paralelní čas při použití $i = 2, \cdot, 32$ procesorů na sítích Ethernet a InfiniBand.
3. Z naměřených dat sestavte grafy zrychlení $S(n, p)$. Zjistěte, zda a za jakých podmínek došlo k superlineárnímu zrychlení a pokuste se je zdůvodnit.
4. Vyhodnoďte komunikační složitost dynamického vyvažování zátěže a posuďte vhodnost vámi implementovaného algoritmu pro hledání dárce a dělení zásobníku při řešení vašeho problému. Posuďte efektivnost a škálovatelnost algoritmu. Popište nedostatky vaší implementace a navrhněte zlepšení.
5. Empiricky stanovte granularitu vaší implementace, tj., stupeň paralelismu pro danou velikost řešeného problému. Stanovte kritéria pro stanovení

Typ matice	Počet procesorů	čas - ethernet [s]	čas - infiniband [s]
Matice 1	2	0.293	0.237
Matice 2	2	0.486	0.473
Matice 3	2	0.361	0.361
Matice 1	4	0.240	0.325
Matice 2	4	0.553	0.550
Matice 3	4	0.419	0.362
Matice 1	8	0.290	0.254
Matice 2	8	0.382	0.573
Matice 3	8	0.359	0.595
Matice 1	12	-	-
Matice 2	12	-	-
Matice 3	12	-	-
Matice 1	16	-	-
Matice 2	16	-	-
Matice 3	16	-	-
Matice 1	24	-	-
Matice 2	24	-	-
Matice 3	24	-	-

Tabulka 3: Doba běhu paralelního algoritmu.

mezí, za kterými již není účinné rozkládat výpočet na menší procesy, protože by komunikační náklady převážily urychlení paralelním výpočtem.

4 Závěr

Celkové zhodnocení semestrální práce a zkušenosti získaných během semestru.

5 Literatura

A Návod pro vkládání grafů a obrázků do Latexu

Nejjednodušší způsob vytvoření obrázku je použít vektorový grafický editor (např. xfig nebo jfig), ze kterého lze exportovat buď

- postscript formáty (ps nebo eps formát) nebo
- latex formáty (v pořadí prostý latex, latex s macry epic, eepic, eepicemu). Uvedené pořadí odpovídá růstu komplikovanosti obrázků který formát podporuje (prostá latex macra umožňují pouze jednoduché, epic makra něco mezi, je třeba vyzkoušet).

Následující příklady platí pro všechny případy.

Obrázek v postscriptu, vycentrováný a na celou šířku stránky, s popisem a číslem. Všimnete si, jak řídit velikost obrazku.

Obrázek pouze vložený mezi řádky textu, bez popisu a číslování.

Latexovské obrázky mají přípony *.latex, *.epic, *.eepic, a *.eepicemu, respective.

Obrázek 1: Popis vašeho obrázku

bez čísla a popisu.

Takhle jednoduše můžete poskládat obrázky vedle sebe.

Řídit velikost latexovských obrázků lze příkazem

```
\setlength{\unitlength}{0.1mm}
```

které mění měřítko rastru obrázku, Tyto příkazy je ale současně nutné vyhodit ze souboru, který xfig vygeneroval.

Pro vytváření grafu lze použít program gnuplot, který umí generovat postscriptovy soubor, který vložíte do Latexu výše uvedeným způsobem.