Semestrální projekt MI-PPR 2015/2016:

Paralelní algoritmus pro řešení problému Zobecněné bisekční šířky

Petr Elexa
Adam Kučera
magisterské studium, FIT ČVUT, Thákurova 9, 160 00 Praha 6

December 20, 2015

1 Definice problému a popis sekvenčního algoritmu

1.1 Zobecněná bisekční šířka

1.1.1 Vstupní data

- a = přirozené číslo
- n = přirozené číslo představující počet uzlů grafu G, $n \geq 5$
- m = přirozené číslo představující počet hran grafu G, $m \geq n$
- k = přirozené číslo řádu jednotek představující průměrný stupeň uzlu grafu G, $n \geq k \geq 3$ $G(V, E)$ = jednoduchý souvislý neorientovaný neohodnocený graf o n uzlech V a m hranách E (nepovinný, při jeho absenci je graf s danými parametry vygenerován).

1.1.2 Úkol

Nalezněte rozdělení množiny n uzlů grafu G do dvou disjunktních podmnožin X a Y tak, že podmnožina X obsahuje a uzlů, podmnožina Y obsahuje $n-a$ uzlů a počet všech hran $\{u, v\}$ takových, že u je z X a v je z Y, je minimální.

1.1.3 Výstup algoritmu

Výpis disjuktních množin uzlů X a Y a počet hran tyto množiny spojující.

1.1.4 Sekvenční algoritmus

Řešení existuje vždy. Vždy lze sestrojit zobecněný bisekční řez grafu. Sekvenční algoritmus je typu BB-DFS s hloubkou prohledávaného prostoru omezenou na $|a|$. Přípustný mezistav je definovaný rozdělením množiny uzlů na dvě disjunktní podmnožiny X a Y. Přípustná koncová řešení jsou všechna zkonstruovaná rozdělení množiny uzlů grafu G do množin X a Y. Cena, kterou minimalizujeme, je počet hran spojující X a Y.

Těsná dolní mez je rovna 1.

Triviální horní mez je rovna m.

1.2 Naše implementace problému

1.3 Formát výstupních dat

Algortitmus vypíše množinu X o velikosti a jako seznam identifikačních čísel uzlů podle toho, jak byly zadány ve vstupní matici grafu. Zároveň vypíše, kolik hran mezi množinou X a zbytkem grafu (množinou Y) existuje.

1.4 Experimentálně naměřená doba výpočtu

2 Popis paralelního algoritmu a jeho implementace v MPI

Popište paralelní algoritmus, op?t vyjd?te ze zadání a p?esn? vymezte odchylky, zvlášt? u algoritmu pro vyvažování zát?že, hledání dárce, ci ukon?ení výpo?tu. Popište a vysv?tlete strukturu celkového paralelního algoritmu na úrovni proces? v MPI a strukturu kódu jednotlivých proces?. Nap?. jak je naimplementována smy?ka pro ?innost proces? v aktivním stavu i v stavu ne?innosti. Jaké jste zvolili konstanty a parametry pro škálování algoritmu. Struktura a sémantika p?íkazové ?ádky pro spoušt?ní programu.

3 Naměřené výsledky a vyhodnocení

1. Zvolte t?i instance problému s takovou velikostí vstupních dat, pro které má sekven?ní algoritmus ?asovou složitost kolem 5, 10 a 15 minut.

Pro me?ení ?as pot?ebný na ?tení dat z disku a uložení na disk neuvažujte a zakomentujte ladící tisky, logy, zprávy a výstupy.

2. M??te paralelní ?as p?i použití $i = 2, \cdot, 32$ procesor? na sítích Ethernet a InfiniBand.

3. Z nam??ených dat sestavte grafy zrychlení $S(n, p)$. Zjist?te, zda a za jakych podmínek došlo k superlineárnímu zrychlení a pokuste se je zd?vodnit.

4. Vyhodno?te komunika?ní složitost dynamického vyvažování zát?že a posu?te vhodnost vámi implementovaného algoritmu pro hledání dárce a d?lení zásobníku pri ?ešení vašeho problému. Posu?te efektivnost a škálovatelnost algoritmu. Popište nedostatky vaší implementace a navrhn?te zlepšení.

5. Empiricky stanovte granularitu vaší implementace, tj., stupe? paralelismu pro danou velikost ?ešeného problému. Stanovte kritéria pro stanovení mezí, za kterými již není u?inné rozkládat výpo?et na menší procesy, protože by komunika?ní náklady prevážily urychlení paralelním výpo?tem.

4 Závěr

Celkové zhodnocení semestrální práce a zkušenosti získaných b?hem semestru.

5 Literatura