Semestrální projekt MI-PPR.2 2014/2015
Paralelní algoritmus pro řešení problému

Karel Fiala
Michal Kučera

České vysoké učení technické v Praze
Fakulta informačních technologií
Thákurova 9, 160 00 Praha 6
Česká republika

30. listopadu 2014

Obsah

1 Definice problému a popis sekvenčního algoritmu

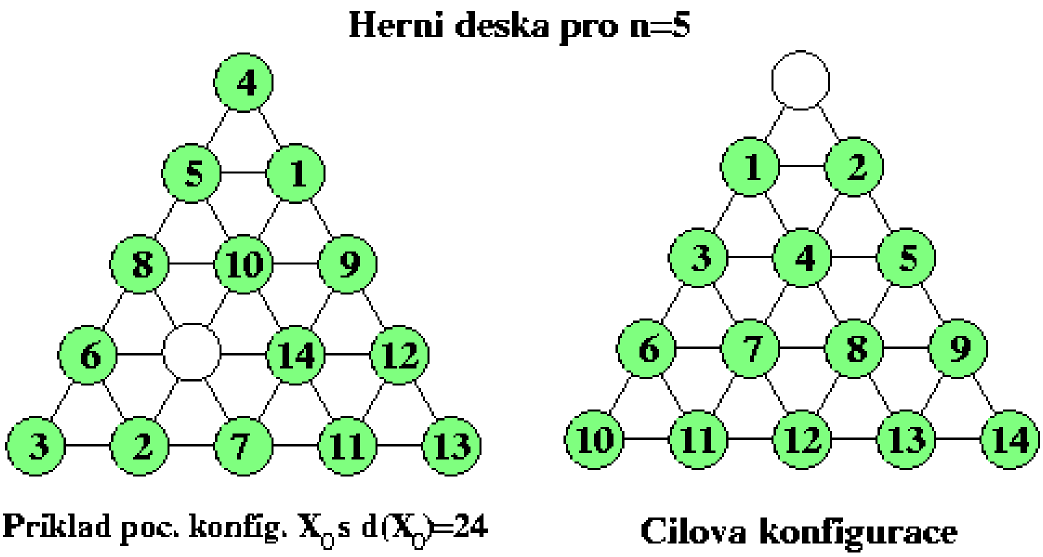
1.1 Úloha PEK: Permutace číselných koleček

1.1.1 Vstupní data

n = délka rovnostranného trojúhelníka, n¿=5 q = přirozené číslo, n na 2¿q X0 = počáteční konfigurace zkonstruovaná zpětným provedením q náhodných tahu z cílové konfigurace. Platí q¿=d(X0).

1.1.2 Pravidla a cíl hry

Herní deska má tvar rovnostranného trojúhelníka o délce strany n, kde v i-tem řádku je i políček, ležících na průsečících úseček, rovnoběžných se stranami trojúhelníka. V těchto políčkách jsou podle určité permutace rozmístěna kolečka s čísly 1,. . . ,M-1, kde M=n(n+1)/2. Jedno políčko zůstává volné, viz příklad na obrázku vlevo.



Obrázek 1: Hrací plocha

Tomuto rozmístění koleček budeme říkat počáteční konfigurace X0. Jeden tah je přesun kolečka na sousední volné políčko ve směru některé úsečky. Cílem hry je použitím minimálního počtu tahů převést počáteční konfiguraci X0 do cílové konfigurace C, ve které jsou kolečka seřazena vzestupně po řádcích tak, že políčko na horním vrcholu trojúhelníkové desky je volné, viz obrázek vpravo. Úloha má vždy řešení.

1.1.3 Definice

Je-li X konfigurace rozmístění všech koleček na herní desce, pak

t(X) je počet doposud provedených tahů, kterými jsme převedli počáteční konfiguraci X0 do konfigurace X. d(X) je spodní mez počtu tahů, kterými se lze dostat z konfigurace X do cílové konfigurace C. Tato spodní mez je rovna součtu vzdáleností koleček od jejich cílových políček. Vzdálenost 2 políček v této síti se počítá takto: Jsou-li obě políčka na úsečce rovnoběžné se stranou trojúhelníka, pak je vzdálenost rovna jejich lineární vzdálenosti po této úsečce. V opačném případě tvoří políčka vrcholy kosodélníka a vzdálenost se rovná součtu délek jeho dvou stran. Spodní mez počtu tahů nejlepšího možného řešení je tedy d(X0). Generování počátečního stavu:

X0 vygenerujeme nejprve q náhodně provedenými zpětnými tahy z cílové konfigurace C.

1.1.4 Výstup algoritmu

Výpis nejkratší posloupnosti tahů vedoucí z počáteční konfigurace do cílové konfigurace.

1.1.5 Sekvenční algoritmus

Sekvenční algoritmus je typu BB-DFS s neomezenou hloubkou stromu konfiguraci. Přípustný stav je cesta z počáteční do cílové konfigurace C. Cena, která se minimalizuje, je počet tahů takové cesty.

Horní mez počtu tahů je q.

Dolní mez je d(X0).

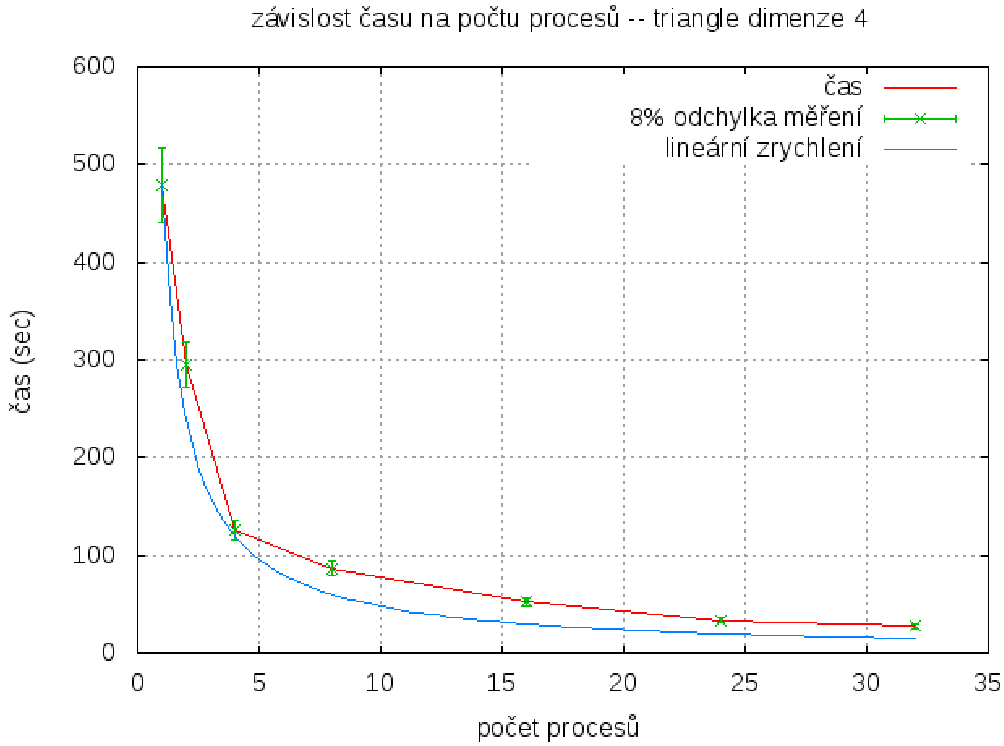
1.1.6 Paralelní algoritmus

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

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, ci 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

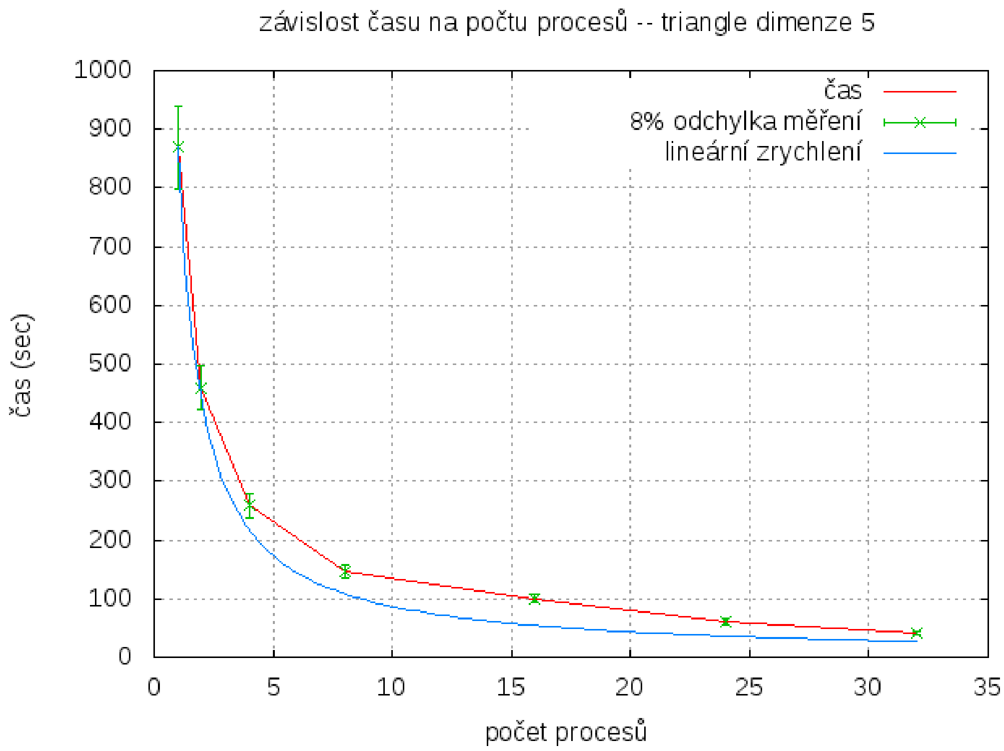


Obrázek 2: Měření pro trojúhelník dimenze 4

1. Zvolte tři instance problému s takovou velikostí vstupních dat, pro které má sekvenční algoritmus časovou složitost kolem 5, 10 a 15 minut. Pro meření 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 jakych podmínek došlo k superlineárnímu zrychlení a pokuste se je zdůvodnit.



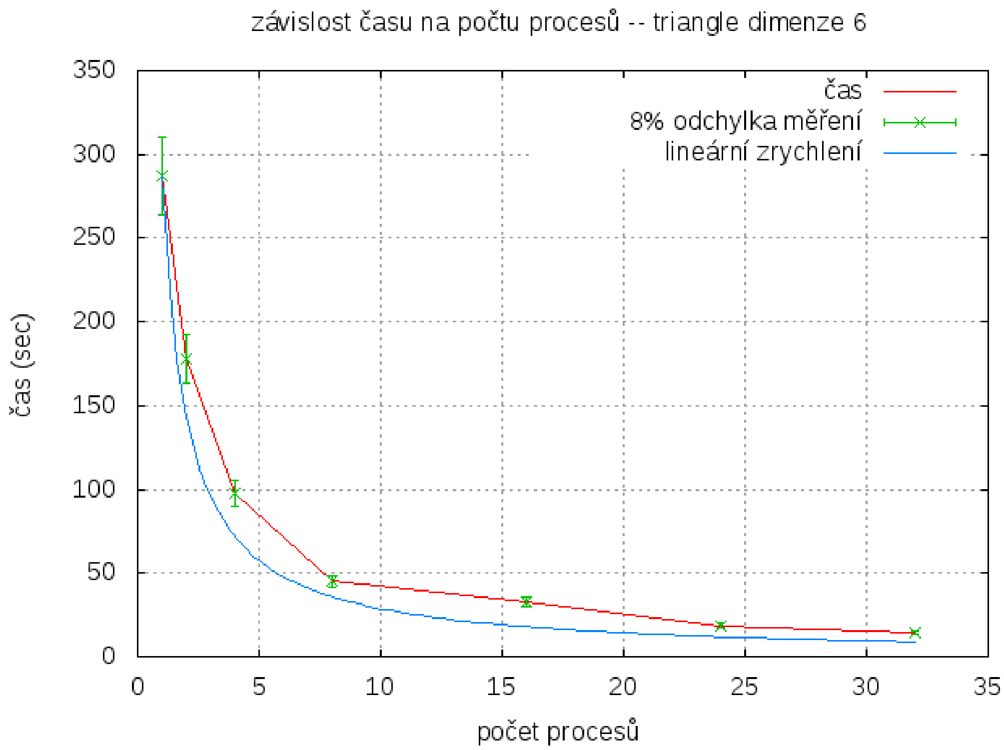
Obrázek 3: Měření pro trojúhelník dimenze 5

4. Vyhodnoďte komunikační složitost dynamického vyvažování zátěže a posuďte vhodnost vámi implementovaného algoritmu pro hledání dárce a dělení zásobníku pri 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 mezí, za kterými již není účinné rozkládat výpočet na menší procesy, protože by komunikační náklady prevážily urychlení paralelním výpočtem.

4 Závěr

Celkové zhodnocení semestrální práce a zkušenosti získaných během semestru.



Obrázek 4: Měření pro trojúhelník dimenze 6

5 Literatura