

**Semestrální projekt MI-PAR 2010/2011:**

**Paralelní algoritmus pro řešení problému**

**Vladimír Kroupa**

**Jan Stadler**

**magisterské studium, FIT ČVUT, Kolejní 550/2, 160 00 Praha 6**

**November 7, 2012**

## **1 Definice problému a popis sekvenčního algoritmu**

Popište problém, který váš program řeší. Jako výchozí použijte text zadání, který rozšířte o přesné vymezení všech odchylek, které jste vůči zadání během implementace provedli (např. úpravy heuristické funkce, organizace zásobníku, apod.). Zmíňte 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

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  $i = 2, \dots, 32$  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 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. Vypuštěním závorek `figure` dostanete opět pouze rámeček v

Figure 1: Popis vašeho obrázku

textu 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.