Paralelné algoritmy, časť č. 2

František Mráz

Kabinet software a výuky informatiky, MFF UK, Praha

Paralelné algoritmy, 2011/2012

Obsah

Úvod

Ciele:

- vzájomné simulácie paralelných a sekvenčných modelov výpočtov
- použitie základných princípov návrhu paralelných algoritmov:

  rozdeľ a panuj – známe zo sekvenčných algoritmov; úloha sa rozdelí na jednoduchšie podúlohy, tie sa riešia paralelne a potom sa z riešení podúloh skladá riešenie celej úlohy.

  práca v tímoch – každej variante riešenia je pridelený jeden tím, ktorý overuje, či jeho riešenie odpovedá vstupom. Práve jeden tím zistí, že jeho riešenie zodpovedá vstupu a vydá svoje riešenie ako výsledok.

Dohody:

- obmedzená sada inštrukcií: $+$, $-$, div 2, $\lfloor x2^y \rfloor$ (aritmetické posuny)
- pre jednoduchosť budeme predpokladať, že vstupné čísla sú nezáporné celé čísla

Počítanie funkcií $\lfloor \log n \rfloor$ a $\lceil \log n \rceil$

- vstup: číslo n
- stačí $\lfloor \log n \rfloor$ procesorov
- jednočlenné tímy – i-ty tím (procesor) overuje, či $i = \lfloor \log n \rfloor$:

$$v := \left\lfloor \frac{n}{2^i} \right\rfloor$$
$$\text{if } (v > 0) \text{ and } \left(\left\lfloor \frac{v}{2} \right\rfloor = 0\right) \text{ then } \overline{r_1} := i$$

- pre výpočet $\lceil \log n \rceil$ pridáme: ak $n = 2^i$ tak výsledok i, inak $i + 1$.
-

Počet proc.	Čas	Šírka slova	Model
$\lfloor \log n \rfloor$	$O(1)$	$O(\log n)$	CREW

Počítanie maxima z n čísel

- Bez ujmy na všeobecnosti môžeme predpokladať, že n je mocnina 2 (viz. vyššie). Vstupy sú x_0 v $\bar{r}_1, \ldots, x_{n-1}$ v $\bar{r}_n$.
- Spočítame $\log n$ a n^2 (aritm. posunmi); výpočtu sa zúčastnia procesory s PID$< n^2$.
- Každý procesor rozdelí svoje PID na dve ($\log n$)-bitové čísla i, j. $i \in \{0, \ldots, n-1\}$ je číslo tímu, $j \in \{0, \ldots, n-1\}$ je číslo člena tímu.
- i-ty tím overuje, či je x_i maximum a bude komunikovať prostredníctvom registra $\bar{r}_{n+1+i}$; $(i, 0)$ je manažér
  1. if $j = 0$ then $\bar{r}_{n+1+i} := 0$
  2. if $(x_i < x_j)$ then $\bar{r}_{n+1+i} := 1$
  3. if $(j = 0)$ and $(\bar{r}_{n+1+i} = 0)$ then $\bar{r}_1 := \bar{r}_{i+1}$
-

Počet proc.	Čas	Šírka slova	Model
n^2	$O(1)$	min. $2 \log n$	COMMON

Sčítanie n b-bitových čísel

- predpokladajme, že n je mocnina 2 a vstupné čísla $x_0, \ldots, x_{n-1}$ sú kladné
- súčet n b-bitových čísel nebude mať viac než $O(b + \log n)$ bitov – toto číslo bude potrebovať každý procesor:

  $\log n$ viz alg. vyššie
  b spočítame max. a použijeme hornú celú časť logaritmu

- $b + \log n$ zaokrúhlime na najbližšiu vyššiu alebo rovnú mocninu 2
- $2^{n(b+\log n)}$ tímov po n procesoroch; procesor interpretuje svoje PID ako n čísel $y_0, \ldots, y_{n-1}$ s $b + \log n$ bitmi a číslo j s $\log n$ bitmi

y_{n-1}	y_{n-2}	$\cdots$	y_0	j

- každý procesor extrahuje z PID svoje číslo v tíme j, 0-tý člen tímu extrahuje y_0, j-ty ($j > 0$) člen tímu extrahuje y_j a y_{j-1}. Stačia na to posuny o $i(b + \log n)$ bitov. Veľkosť posunu je $i \times$ *mocnina* $2 \Rightarrow$ v konštantnom čase.
  1. 0-tý člen tímu kontroluje či $y_0 = x_0$
  2. j-ty člen tímu kontroluje, či $y_j = y_{j-1} + x_j$
- Procesory, ktoré zistia nerovnosť, to oznámia manažérovi (prostredníctvom komunikačného registra)

Sčítanie n b-bitových čísel (pokračovanie)

- práve jeden tím nezistí žiadnu nerovnosť – jeho manažér extrahuje y_{n-1} a vydá výsledok.
- to, že sa jedná o b-bitové čísla ovplyvňuje počet procesorov, ale nie dobu výpočtu
-

Počet proc.	Čas	Šírka slova	Model
$n\,2^{n(b+\log n)}$	$O(1)$	$n\,(b+\log n))+\log n$	COMMON

Násobenie b-bitových čísel v konštantnom čase

- školská metóda násobenia, predpokladáme, že x_1 a x_2 sú kladné
- spočítame b ($\log$ väčšieho z nich) a vložíme do $\overline{r}_0$
- i-ty procesor:
  - extrahuj i-ty bit čísla x_2 (zodpovedá rádu 2^i)
  - ak je bit 1, tak posuň x_1 o i bitov doľava (tj. násob 2^i) a výsledok zapíš do $\overline{r}_{i+1}$
  - inak do $\overline{r}_{i+1}$ zapíš 0
- súčet čísel $\overline{r}_1, \ldots, \overline{r}_b$ spočítame v konštantnom čase
-

Počet proc.	Čas	Šírka slova	Model
$b\,2^{b\,(2b+\log b)}$	$O(1)$	$O(b\,(b+\log b)) = O(b^2)$	COMMON

Odmocňovanie čísel v konštantnom čase

- nech $c > 1$, prirodzené číslo, n vstupné celé číslo. Chceme spočítať $\lceil n^{\frac{1}{c}} \rceil$
- n tímov procesorov
- tím i ($0 \leq i < n$) kontroluje, či $i = \lceil n^{\frac{1}{c}} \rceil$
  1. počíta i^c pomocou $c - 1$ násobení; jeden tím má $2^{O(\log^2 n)}$ procesorov
  2. if $(n \leq i^c)$ and $(i^{c-1} < n)$ then $\bar{r}_1 := i$
-

Počet proc.	Čas	Šírka slova	Model
$n\,2^{O(\log^2 n)}$	$O(1)$	$O(\log^2 n)$	COMMON

Sčítanie čísel v konštantnom čase s menšou šírkou slova

- budeme potrebovať sčítať čísla s konštantnou bitovou dĺžkou na COMMON PRAM s lineárnou dĺžkou slova
- Vstup: n kladných celých čísel s $n^{1-\frac{1}{c}}$ bitmi pre nejaké celé číslo $c \geq 2$
- Dohoda: nebudeme označovať horné a dolné celé časti
- súčet a každý čiastočný súčet bude mať maximálne $O(n^{1-\frac{1}{c}})$ bitov
- Algoritmus:

	spočítame $m = n^{\frac{1}{c+1}}$
Fáza 1:	rozdeľ vstupné čísla do skupín po m čísel a urob súčet každej skupiny
	opakuj c-krát
	dostaneme $\frac{n}{m^c}$ čísel
Fáza 2:	sčítaj $\frac{n}{m^c}$ čiastočných súčtov

Sčítanie čísel v konštantnom čase s menšou šírkou slova

- predvýpočet: čas $O(1)$, malá šírka slova pre veľké n
- Fázy 1 a 2 sa urobia starým postupom v konšt. čase
- šírka slova na Fázu 1:

$$m \cdot n^{1-\frac{1}{c}} = n^{1-\frac{1}{c^2+c}}$$

šírka slova na Fázu 2:

$$\frac{n}{m^c} \cdot n^{1-\frac{1}{c}} = n^{1-\frac{1}{c^2+c}}$$

Sčítanie čísel v konštantnom čase s menšou šírkou slova

Veta

n $(n^{1-\frac{1}{c}})$*-bitových čísel sa dá sčítať v konštantnom čase na PRAMe so šírkou slova* $O\left(n^{1-\frac{1}{c^2+c}}\right)$.

Ekvivalentne: $\forall\lambda, 0 < \lambda \leq \frac{1}{2}$ $\quad\exists\mu > 0$: *súčet* n *čísel s* $n^{1-\lambda}$ *bitmi sa dá spočítať v čase* $O(1)$ *na PRAMe COMMON so šírkou slova* $n^{1-\mu}$

- obecná technika: použitím jedného algoritmu na menšie skupiny sa veľkosť úlohy zredukuje a na zmenšenú úlohu sa aplikuje pôvodný algoritmus.
  Zisk: zníženie potrebnej šírky slova z
  $n(n^{1-\frac{1}{c}} + \log n) + \log n \approx n^{2-\frac{1}{c}} > n^{\frac{3}{2}}$

Počítanie funkcií prev a last

- Nech $x_i \in \mathbb{N}, 1 \leq x_i \leq n$, $1 \leq i \leq n$
  - prev : $\mathbb{Z}^n \rightarrow \mathbb{Z}^n$ a prev$(x_1, \ldots, x_n) =< y_1, \ldots, y_n >$
    kde $y_i = \begin{cases} j & \text{ak } x_j = x_i, j < i \text{ a } x_k \neq x_i \text{ pre } j < k < i \\ 0 & \text{ak také } j \text{ neexistuje} \end{cases}$
  - last : $\mathbb{Z}^n \rightarrow \mathbb{Z}^n$ a last$(x_1, \ldots, x_n) =< y_1, \ldots, y_n >$
    kde $y_i = \begin{cases} j & \text{ak } x_j = x_i \text{ a } x_k \neq x_i \text{ pre } j < k \leq n \\ 0 & \text{ak také } j \text{ neexistuje} \end{cases}$
- prev a last sa dajú spočítať v konštantnom čase na PRAMe so šírkou slova $O(\log n)$

Počítanie funkcií prev a last

- prev:
  - n^2 tímov – tím zodpovedá dvojici $< i, j >$, $1 \leq i, j \leq n$
  - každý tím má komunikačný register v spoločnej pamäti; ten manažér tímu inicializuje na 0
  - k-ty procesor, $j < k < i$ (ostatné nepracujú) overí, či $x_k \neq x_i$, ak zistí rovnosť, tak do komunikačného registra zapíše 1
  - manažér skontroluje komunikačný register tímu, ak je tam 0 a $x_i = x_j$ tak do $\overline{r}_i$ zapíše j
-

Počet proc.	Čas	Šírka slova	Model
n^3	$O(1)$	$O(\log n)$	COMMON

Počítanie súčtu n čísel

- Označenie: Nech $T(n, P(n))$ je čas potrebný na výpočet súčtu n čísel (konečnej alebo nekonečnej) pologrupy pomocou $P(n)$ procesorov na modele COMMON PRAM. [Namiesto $T(n, n)$ budeme písať $T(n)$.]

Definícia

Hovoríme, že $f : \mathbb{Z} \to \mathbb{Z}$ je konštruovateľná v konštantnom čase, ak COMMON PRAM s n procesormi môže spočítať $f(n)$ zo vstupu n v konštantnom čase.

- s rozšírenou sadou aritmetických inštrukcií sú polynómy, $\log n$, odmocniny, ...konštruovateľné v konštantnom čase.

Počítanie súčtu n čísel

Lemma

Nech $f : \mathbb{N} \to \mathbb{N}$, $f(n) \leq n$, pre všetky $n \geq 0$, je konštruovateľná v konštantnom čase. Potom

$$T(n) \leq T\left(\left\lceil \frac{n}{f(n)} \right\rceil\right) + T(f(n), n) + O(1) \ .$$

- **Dôkaz:** vstup n čísel, n procesorov
  spočítame $f(n)$; procesory rozdelíme do $f(n)$ tímov po maximálne $\lceil \frac{n}{f(n)} \rceil$ procesoroch
  každý tím sčíta svoje vstupy v čase $T(\lceil \frac{n}{f(n)} \rceil)$
  Predpokladáme, že $T(n)$ je monotónna neklesajúca funkcia. Dostaneme $f(n)$ čiastočných súčtov, ktoré sa dajú sčítať v čase $T(f(n), n)$
- pre $f(n) = O(\log n)$ sa súčet n čísel v *konečnej* pologrupe dá spočítať v čase $O(\frac{\log n}{\log \log n})$ s použitím n procesorov.

Počítanie maxima

- n^2 procesorov – čas $O(1)$
- n procesorov – čas $O(\log n)$
- pre $f(n) = \sqrt{n}$

$$T(n) \leq T(\sqrt{n}) + T(\sqrt{n}, n) + O(1) = T(\sqrt{n}) + O(1) \leq O(\log \log n)$$

- s $n^{1+\frac{1}{k}}$ procesormi, pre nejakú konštantu $k \geq 1$, a $f(n) = \sqrt{n}$ sa hĺbka rekurzie obmedzí na $\log k$ – konštantný čas