Approfondimento Algoritmica

Liva Giovanni

Università di Udine

20 agosto 2014

Argomenti

Introduzione

Vari modelli di calcolo, si differenziano sul livello di astrazione dal modello reale

- *VLSI*: Attenzione ai limiti fisici dei processori
- *PRAM*: Modello teorico non implementabile che non considera i problemi di comunicazione

Principio di Brent

Abbiamo una computazione parallela eseguita in t passi.

- Supponiamo di avere x_i operazioni ad ogni passo i
- Il numero di processori necessario è $d = max\ x_i$
- Possediamo solo $p < m$ processori
- Possiamo simulare la computazione originaria con p processori in $\lceil x_i/p \rceil$ passi ad ogni i-esimo step
- In totale la simulazione viene eseguita in $\lceil \frac{\sum_i x_i}{p} \rceil + t$ passi

Parallel Random-Access Machine

- Si basa sul modello delle *RAM* introducendo una memoria globale (*registro accumulatore*) dove le varie *RAM* possono comunicare
- In tempo $\mathcal{O}(1)$ possiamo r/w una cella della memoria locale o globale oppure eseguire una operazione *RAM*

Un programma *PRAM* $\mathbb{P} = (\Pi_1, \ldots, \Pi_q)$ è fatto da q macchine *RAM* indipendenti dove q è una funzione $q(m, n)$ dove $m = |I|$ ed $n = \ell(I)$. Normalmente il numero di *RAM* richiesto dipende solo da m.

Parallel Random-Access Machine :: Tipologie e Ugualianze

In base a come gestiamo i conflitti di r/w sul registro accumulatore abbiamo 3 diverse tipologie di PRAM

- Exclusive-Read Exclusive-Write (*EREW*)
- Concurrent-Read Exclusive-Write (*CREW*)
- Concurrent-Read Concurrent-Write (*CRCW*)

Per risolvere i conflitti di scrittura abbiamo 3 metodi:

- *Common*: Tutti i processi che insistono in una stessa locazione devono scrivere lo stesso valore
- *Arbritary*: Tra tutti i processi che provano a scrivere, solo uno ha successo. L'algoritmo deve comunque funzionare a prescindere da chi vince
- *Priority*: Il processo l'identificativo più basso è quello che scrive

Parallel Random-Access Machine :: Tipologie e Ugualianze

L'ordine con il quale sono state presentate è anche l'ordine di potenza dei vari modelli.
Tra loro però sono correlati da un fattore logaritmico. Per cui il modello più potente, *CRCW Priority* può essere simulato da una *EREW* con lo stesso numero di processori e con un tempo parallelo aumentato di $\mathcal{O}(log\ P)$; dove P è il numero di processori.

- Dimostrazione a voce

Parallel Random-Access Machine :: Algoritmo Ottimale

Definizione

$polylog(n) = \bigcup_{k>0} \mathcal{O}(log^k n)$
Sia S un programma sequenziale che opera in tempo $T(n)$.

Diremo che il programma A di una *PRAM* per S che opera in tempo $t(n)$ con $p(n)$ processori è ottimale se:

- $t(n) = polylog(n)$
- $w(n) = p(n) \times t(n) = \mathcal{O}(T(n))$

Paragraphs of Text