



Calcolo Parallelo e Distribuito

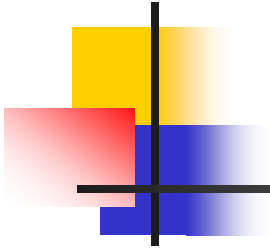
openMP:

Una prova d'esame molto facile

Docente: Prof. L. Marcellino

Tutor: Prof. P. De Luca

Esercitazione in aula



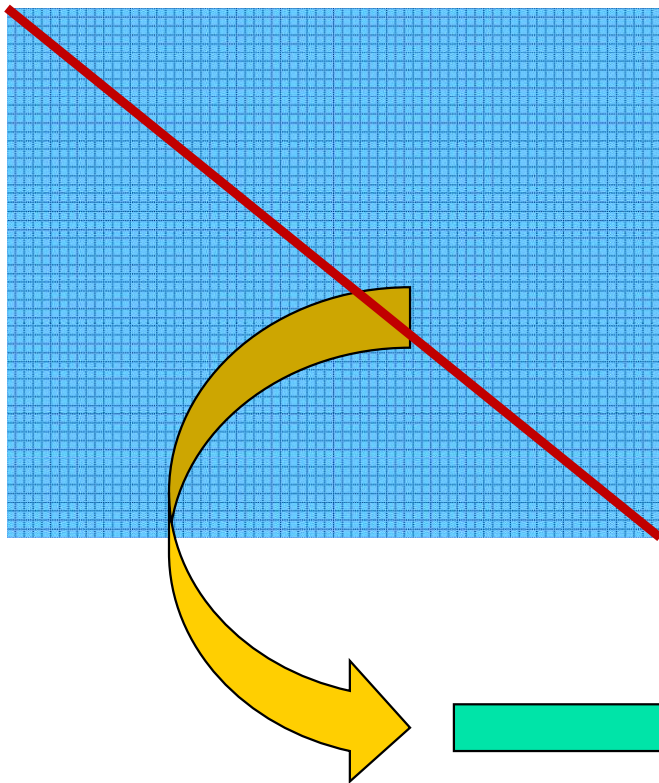
Assegnata una matrice di dimensione $N \times N$, estrarre in parallelo gli elementi della diagonale principale e ricopiarli in un vettore, quindi calcolarne il massimo, con una procedura parallela

Strategia di parallelizzazione - parte 1

Il primo passo è un full-parallel

Matrice A quadrata
 $\dim[A] = N \times N$

A



2 scelte:

1. parallel for (eventualmente con uno scheduling mio)
2. distribuzione indici a mano (come visto in somma vettori)

... quante sono le righe N
distribuisco il lavoro per blocchi riga

$\dim[\text{diag}] = N$



Strategia di parallelizzazione - parte 2

Il passo successivo (dipende da una buona riuscita del primo) ha una reduction



$\text{dim}[\text{diag}] = N$

Calcolare il massimo in parallelo

Calcolo il massimo localmente in sottovettori e
poi trovo il massimo tra i massimi

3 scelte:

1. parallel for + reduction (MAX)
2. distribuzione indici a mano (come visto in somma vettori) + reduction (MAX)
3. distribuzione indici a mano (come visto in somma vettori) + collezione sequenziale con CRITICAL

La dim del problema è N

Non devo scorrere tutti gli elementi della matrice



Aspettative

$$T_1(N) = N + N = 2N$$

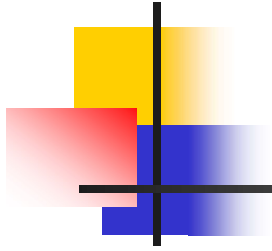
$$\text{dim}[\text{diag}] = N$$



Il primo passo è un full-parallel

Il secondo passo ha una reduction (1 - 2 strategia)

$$T_p(N) = N/p + N/p + \begin{matrix} \nearrow (p-1) \\ \searrow \log(p) \end{matrix}$$



Implementare il codice e
prendere i tempi
al variare di N e p

Proviamo a fare un check sul numero di core ed
adattare la 1 o 2 strategia
o magari combinarle?
considerare $p=8$ massimo