

Progetto APSD : Risoluzione di un sistema lineare

Algoritmo di Gauss-Jordan

Alberto Puritano 191303

L'algoritmo risolve sistemi di equazioni lineari in due fasi:

- 1) Nella prima, "dimezza" la matrice attraverso operazioni di riduzione eliminando la parte sotto la diagonale
- 2) Nella seconda, effettua operazioni di sostituzione partendo dal basso e trovando la soluzione per tutte le variabili

Essendo un'algoritmo molto pesante dal punto di vista computativo, la parallelizzazione aiuta molto nella velocità di esecuzione.

Le stampe della soluzione vengono eseguite su standard std, ma l'algoritmo dispone anche di una semplice interfaccia scritta con la libreria Allegro5, la cui parte grafica è gestita interamente dal master thread. La parte grafica è solo a fini dimostrativi e appesantisce il programma (oltre a fare una pausa da circa 1 secondo ogni ciclo di calcolo), quindi per un corretto svolgimento dei vari test è necessario disattivarla commentando le varie linee in cui viene richiamata la funzione visualizza()

COMPILAZIONE (ovviamente bisogna avere le librerie di allegro):

```
mpic++ Progetto_Paralleli.cpp -o NOMEFILE $(pkg-config allegro-5 allegro_ttf-5 allegro_video-5 allegro_image-5 allegro_acodec-5 allegro_main-5 allegro_audio-5 --libs --cflags)
```

AVVIO (usuale modo di avvio di un programma MPI):

```
mpirun -n numthread --oversubscribe NOMEFILE
```

N= grandezza matrice

Qui una tabella di alcuni test eseguiti sul mio laptop con i3-8130(2 core fisici e 2 virtuali->4 thread totali):

N thread	1	2	4	6	100
N					
10	0s	0s			
1000	0s	0s	0s	0s	5s
1500	2s	1s	0s		
3000	23s	13s	12s	16s	
5000	107s	60s	54s	76s	

Tabella Speedup:

N thread	1	2	4	6	100
N					
10		0			
1000		0	0	0s	0
1500		2	0		
3000		1.7	1.9	1.4	
5000		1.8	1.9	1.4	