

ELABORATO 1 - INTEGRALE

LO BRUTTO FABIO / MAIONE PAOLO

DEFINIZIONE DEL PROBLEMA

Si vuole progettare un algoritmo in OpenMP per risolvere l'integrale definito tra a e b di una funzione $y = f(x)$ su p thread.

In particolare si utilizza l'infrastruttura S.C.o.P.E. per permettere l'esecuzione del software in un ambiente parallelo. Tuttavia, a differenza degli elaborati MPI, il software è eseguibile anche su una macchina multicore locale.

DESCRIZIONE DELL'ALGORITMO

In particolare l'algoritmo implementato nel file *eLaborato_1.c*, consiste nell'elaborazione dell'integrale parziale in parallelo con la **formula trapezoidale composta** sul sotto intervallo di ampiezza

$$h = \frac{(b - a)}{\text{num_intervalli}}$$

$$T_{n+1}(f) = \frac{h}{2} \left(f(a) + f(b) + 2 \sum_{i=1}^n f(a + ih) \right)$$

Questa formula è realizzata attraverso un ciclo for parallelizzato su p thread utilizzando le direttive e le clausole fornite da OpenMP.

Inoltre l'algoritmo progettato comprende anche il caso in cui il numero degli intervalli non è multiplo del numero di thread p a disposizione.

Si è scelto di misurare i tempi di esecuzione nel thread 0 (master) usando la primitiva `omp_get_wtime()` prima e dopo il calcolo parallelo scegliendo il minimo tra 3 misurazioni ripetute.

Infine, si osservi che i controlli di robustezza del software sono stati interamente delegati al master thread.

INPUT, OUTPUT E CONDIZIONI DI ERRORE

- **Input:** la funzione f da integrare, i due estremi dell'intervallo a e b , il numero di intervalli `num_intervalli`.
- **Output:** l'approssimazione dell'integrale definito tra a e b di f .

- **Condizioni di errore:** l'estremo b non deve essere minore di a, il numero di intervalli deve essere un intero positivo non minore del numero di thread.

ESEMPIO DI FUNZIONAMENTO

Nell'immagine seguente vi è un esempio di funzionamento con $p=8$ thread, $f(x) = \frac{x}{x^2 + 5}$,

con $a=0$ e $b=1$ e con `num_intervalli = 100000`.

%esempio di funzionamento
funzionamento

Esempio di funzionamento con 8 processori, in [0,1] con 100000 intervalli

```

[M63000769@ui-studenti-elaborati_open_mp/elaborato_1]
[M63000769@ui-studenti-elaborato_1]$ cat elaborato_1.err
[M63000769@ui-studenti-elaborato_1]$ cat elaborato_1.out
-----
This job is allocated on 8 cpu(s)
Job is running on node(s):
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
-----
PBS: qsub is running on ui-studenti.scope.unina.it
PBS: originating queue is studenti
PBS: executing queue is studenti
PBS: working directory is /homes/DIS/CALCPAR/2019/M63000769/elaborati_open_mp/elaborato_1
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3929375.torque02.scope.unina.it
PBS: job name is elaborato_1
PBS: node file is /var/spool/pbs/aux//3929375.torque02.scope.unina.it
PBS: current home directory is /homes/DIS/CALCPAR/2019/M63000769
PBS: PATH = /usr/lib64/openmpi/1.2.7-gcc/bin:/usr/kerberos/bin:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64:/opt
/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/mpi/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel
64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64_mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/debu
gger/gui/intel64:/opt/d-cache/srm/bin:/opt/d-cache/dcap/bin:/opt/edg/bin:/opt/glite/bin:/opt/globus/bin:/opt/lcg/bin:/usr/local/bin:/bi
n:/usr/bin:/opt/exp_soft/HADOOP/hadoop-1.0.3/bin:/opt/exp_soft/unina.it/intel/composerxe/bin/intel64:/opt/exp_soft/unina.it/MPJExpress
/mpj-v0_36/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
-----
[Thread 0] Il programma sarà eseguito con 8 threads
[Thread 0] L'integrale è pari a 0.0911607783962367.
Il tempo di esecuzione totale è stato di 0.0091370000009192 secondi.
[M63000769@ui-studenti-elaborato_1]$
[M63000769@ui-studenti-elaborato_1]$
[M63000769@ui-studenti-elaborato_1]$

```

ESEMPI DI ERRORE

Nelle successive immagini, invece, sono mostrati i messaggi di errore al verificarsi delle condizioni sopra citate.

%un esempio per ciascuna condizione di errore
errori

Error: numero di thread non positivo

```
M63000769@ui-studenti:~/elaborati_open_mp/elaborato_1
[M63000769@ui-studenti elaborato_1]$ cat elaborato_1.err
libgomp: Invalid value for environment variable OMP_NUM_THREADS
[M63000769@ui-studenti elaborato_1]$ cat elaborato_1.out

This job is allocated on 8 cpu(s)
Job is running on node(s):
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
-----
PBS: qsub is running on ui-studenti.scope.unina.it
PBS: originating queue is studenti
PBS: executing queue is studenti
PBS: working directory is /homes/DIS/CALCPAR/2019/M63000769/elaborati_open_mp/elaborato_1
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3929376.torque02.scope.unina.it
PBS: job name is elaborato_1
PBS: node file is /var/spool/pbs/aux//3929376.torque02.scope.unina.it
PBS: current home directory is /homes/DIS/CALCPAR/2019/M63000769
PBS: PATH = /usr/lib64/openmpi/1.2.7-gcc/bin:/usr/kerberos/bin:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/mpiirt/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64_mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/debugger/gui/intel64:/opt/d-cache/srm/bin:/opt/d-cache/dcap/bin:/opt/edg/bin:/opt/glite/bin:/opt/globus/bin:/opt/icq/bin:/usr/local/bin:/bin:/usr/bin:/opt/exp_soft/HADOOP/hadoop-1.0.3/bin:/opt/exp_soft/unina.it/intel/composerxe/bin/intel64:/opt/exp_soft/unina.it/MPJExpress/mpi-v0_38/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
-----
Errore! Il numero dei threads deve essere maggiore di zero
[M63000769@ui-studenti elaborato_1]$
[M63000769@ui-studenti elaborato_1]$
[M63000769@ui-studenti elaborato_1]$
```

100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

ANALISI DELLE PRESTAZIONI (T(p), S(p), E(p))

Di seguito per brevità si indicherà con p il numero di thread e con N il numero di intervalli.

Tempo di esecuzione - T(p)

Si è scelto di misurare i tempi di esecuzione usando la primitiva `omp_get_wtime()` prima e dopo il calcolo parallelo.

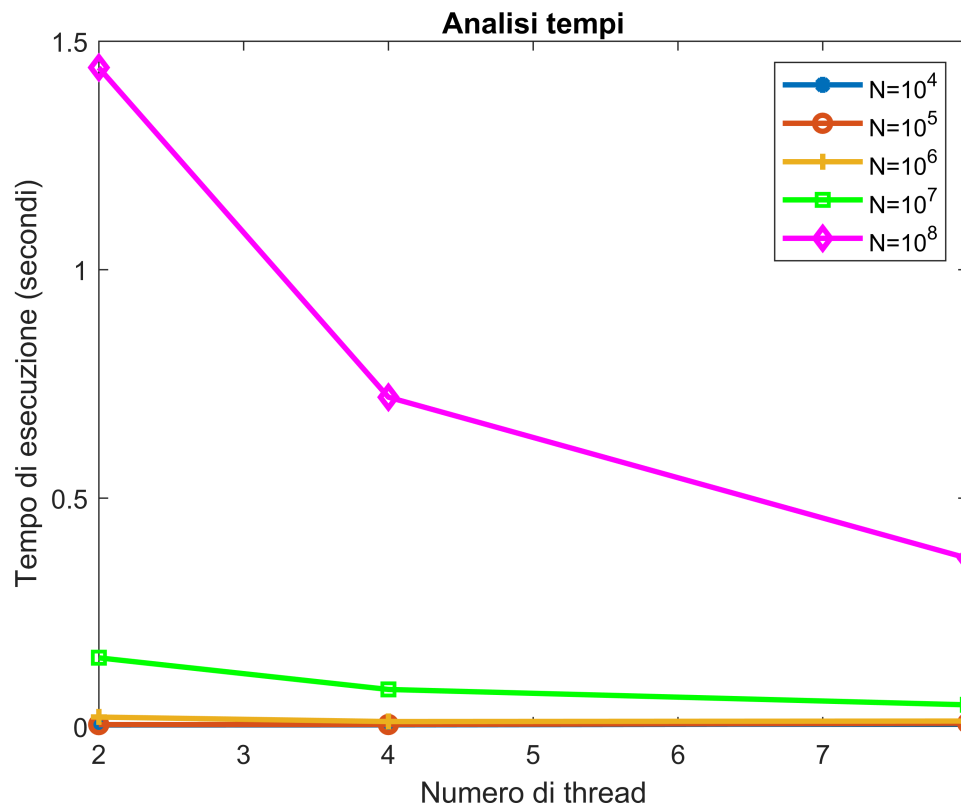
Per ciascuna misurazione (al variare di N da 10k a 100M e al variare di p da 2 a 8) è stato considerato il minimo tra 3 esecuzioni ripetute, eseguite in momenti diversi.

Di seguito si riportano i risultati in forma di tabelle e grafici.

```
%esecuzione script per tabelle e grafici
tempi
```

TEMPI	Tempi				
	10000	100000	1000000	10000000	100000000
2	0,0029639999993378	0,003940000002331	0,02041799999279	0,1501430000062100	1,442007000004230
4	0,0032320000027539	0,004165999998804	0,01060400000279	0,0808800000013434	0,720618000000831
8	0,0056699999986449	0,007055999997142	0,01172599999356	0,0471789999937755	0,367989999995350

Warning: Image is too big to fit on screen; displaying at 67%



Per considerazioni più di dettaglio su questi risultati si rimanda alla sezione Conclusioni.

Speed up ed Efficienza - $S(p)$ ed $E(p)$

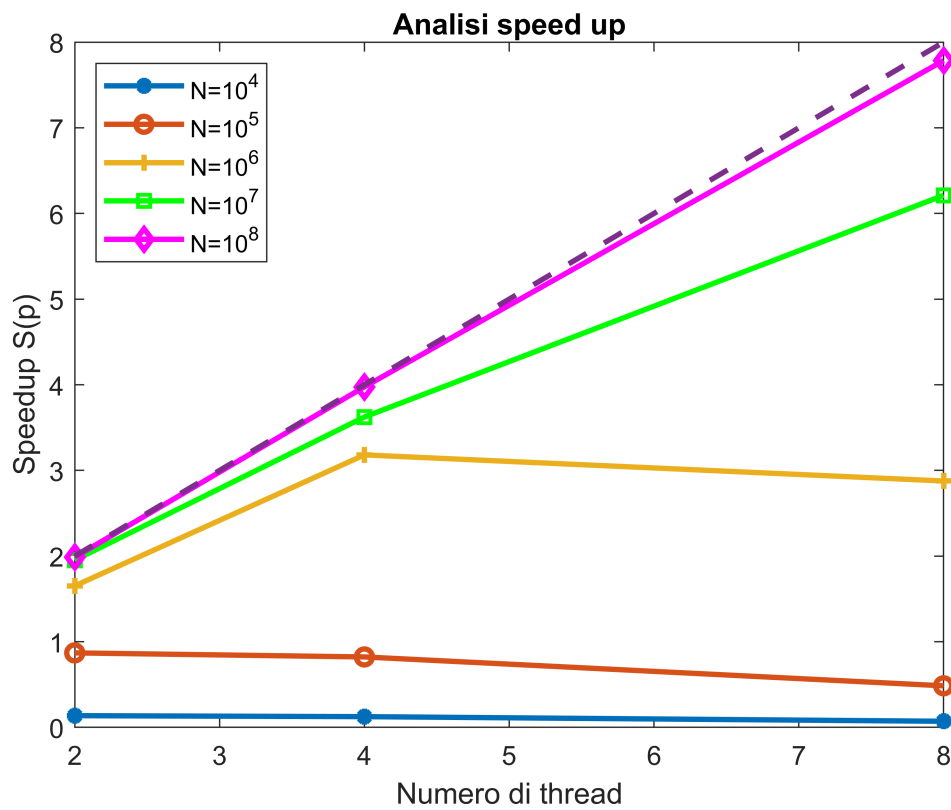
Si è calcolato, inoltre, il tempo di riferimento $T(1)$ che corrisponde al tempo di esecuzione su un processo single-thread.

A partire dai tempi misurati nella sezione precedente e da $T(1)$ è stato calcolato lo speed-up al variare di N e p .

%esecuzione script per tabelle e grafici
speedup

SPEEDUP	Speedup				
	10000	100000	1000000	10000000	100000000
2	0,1370	0,8698	1,6521	1,9518	1,9867
4	0,1256	0,8226	3,1811	3,6233	3,9754
8	0,0716	0,4857	2,8767	6,2115	7,7849

Warning: Image is too big to fit on screen; displaying at 67%

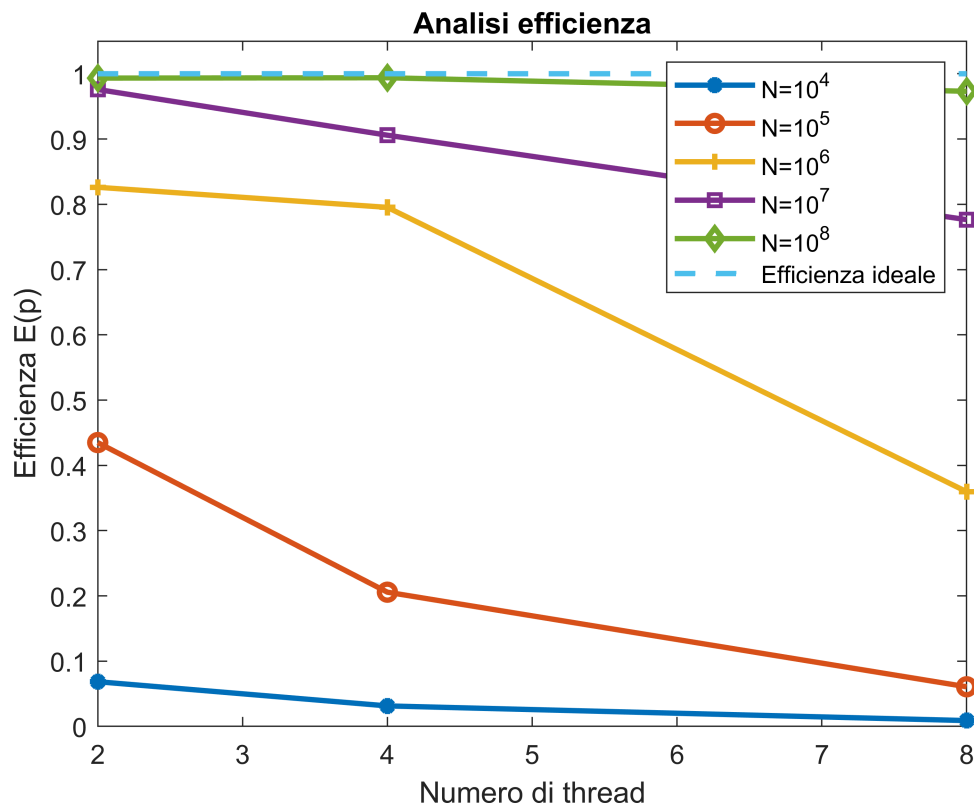


Infine si è calcolata l'efficienza rapportando lo speed up $S(p)$ al numero di thread p .

%esecuzione script per tabelle e grafici
efficienza

Efficienza					
EFFICIENZA	10000	100000	1000000	10000000	100000000
2	0,0685	0,4349	0,8260	0,9759	0,9933
4	0,0314	0,2057	0,7953	0,9058	0,9939
8	0,0090	0,0607	0,3596	0,7764	0,9731

Warning: Image is too big to fit on screen; displaying at 67%



Conclusioni

Dai grafici appena presentati si possono trarre alcune considerazioni.

Analizzando l'efficienza si nota come da $N=10^4$ a $N=10^7$ l'efficienza ottimale si ha per 2 thread. Nel caso $N=10^4$ l'efficienza è estremamente bassa, probabilmente a causa del fatto che la creazione e la terminazione dei thread genera un enorme overhead rispetto alla dimensione del problema.

Invece nel caso di $N=10^8$ l'ottimo è in corrispondenza di 4 thread.

Si possono fare ulteriori considerazioni notando che, per N fissato, l'efficienza peggiora dopo un certo valore di p (ciò verifica sperimentalmente la legge di Amdahl), e che, in generale, all'aumentare sia di N che di p , l'efficienza migliora (verificando la legge di Gustafson).

Analoghe considerazioni per i tempi e lo speedup.

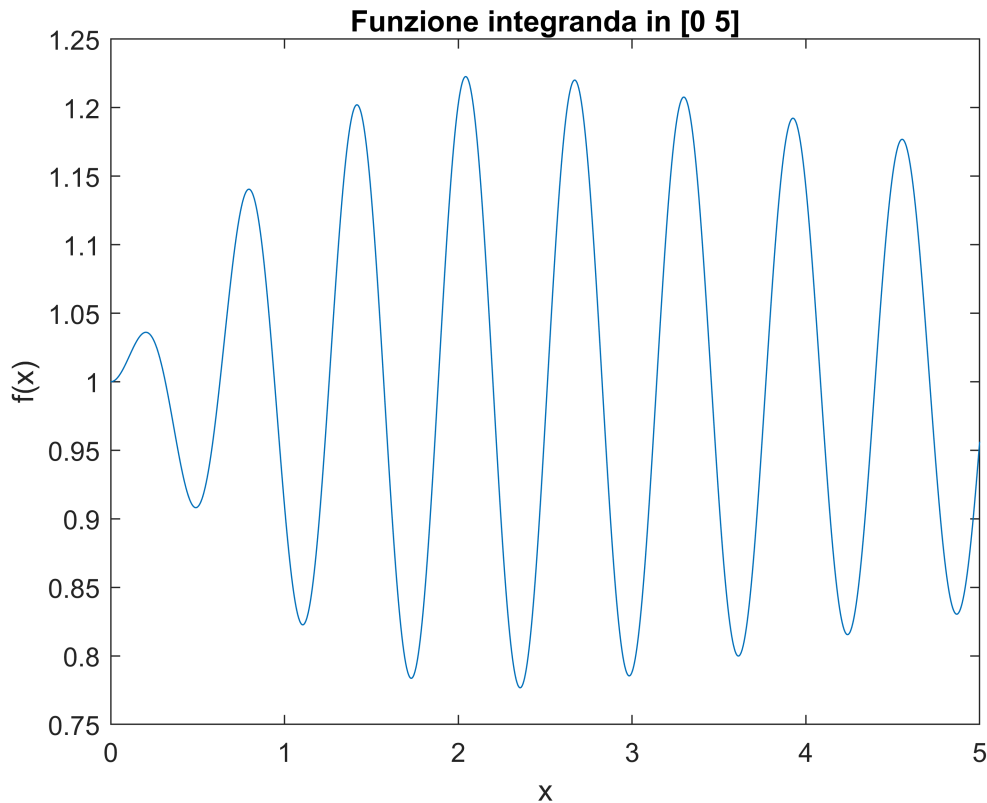
ANALISI DELL' ACCURATEZZA

Confrontando i risultati ottenuti sul cluster Scope e quelli ottenuti su MATLAB si ottengono i seguenti errori relativi. I livelli di accuratezza sono stati misurati fissando a 8 il numero di thread, e facendo variare la dimensione del problema num_intervalli. Gli estremi di integrazione scelti sono $a=0$, $b=5$.

Per i test è stata usata la seguente funzione integranda:

$$f(x) = 1 + \sin(10x) * \frac{x}{x^2 + 5}$$

```
%grafico della funzione integranda  
grafico
```



```
%esecuzione script per i test di accuratezza  
accuratezza
```

```
F = function_handle with value:  
  @(x)1+sin(10.*x).*(x./(x.^2+5))  
risultato_matlab =  
  4.983979125364205  
err_rel_10k_intervalli =  
  1.259469664462556e-08  
err_rel_100k_intervalli =  
  5.915098092282934e-09  
err_rel_1M_intervalli =  
  5.848306046909409e-09
```

```
err_rel_10M_intervalli =  
    5.847643830851800e-09  
err_rel_100M_intervalli =  
    5.847607833100552e-09
```