

ELABORATO IBRIDO

LO BRUTTO FABIO / MAIONE PAOLO

DEFINIZIONE DEL PROBLEMA

Si vuole progettare un algoritmo ibrido (MPI + OpenMP) per risolvere il prodotto vettoriale tra una matrice, di dimensioni $M \times N$, e un vettore di reali, di dimensione N , su p processi e t thread valutando le diverse configurazioni di p e t .

In particolare si utilizza l'infrastruttura S.C.o.P.E. per permettere l'esecuzione del software in un ambiente parallelo.

DESCRIZIONE DELL'ALGORITMO

In particolare le fasi dell'algoritmo, implementato nel file *eLaborato_ibrido.c*, sono:

- 1) Distribuzione per righe della matrice in p processi e distribuzione del vettore: ognuno dei p processi eseguirà il prodotto vettoriale tra la porzione di matrice ricevuta dal processo *root* (cioè quello con rank 0), e tutto il vettore;
- 2) Elaborazione del prodotto vettoriale in parallelo in t thread su ogni processo mediante le direttive e le clausole fornite da OpenMP (`#pragma omp parallel for`);
- 3) Aggregazione del vettore calcolato nel processo *root* che determinerà il risultato finale.

A tal proposito sono state utilizzate le primitive fornite da MPI (rispettivamente per la prima fase `MPI_Scatterv()` e per la terza `MPI_Gatherv()`). Per maggiori dettagli sulle fase 1 e 3 si consulti l'elaborato 2 di MPI riguardo alla distribuzione per righe mentre per ulteriori dettagli sulla fase 2 si consulti l'elaborato 2 di OpenMP riguardo il prodotto matrice vettore.

Inoltre l'algoritmo progettato comprende anche il caso in cui la dimensione M (cioè il numero di righe della matrice) non sia multipla del numero di processi p a disposizione.

Si è scelto di misurare i tempi di esecuzione nel processo di rank 0 usando la primitiva `MPI_Wtime()` tra la fase 2 e la fase 3 scegliendo il minimo tra 3 misurazioni ripetute.

Infine, si osservi che i controlli di robustezza del software sono stati interamente delegati al processo *root*.

INPUT, OUTPUT E CONDIZIONI DI ERRORE

- **Input:** la matrice e il vettore di cui effettuare il prodotto vettoriale, le loro dimensioni M , N e dim_vett e il numero di core numcore da utilizzare.
- **Output:** il vettore risultato del prodotto vettoriale tra la matrice e il vettore.
- **Condizioni di errore:** la dimensione delle righe della matrice deve essere uguale al numero di colonne del vettore e devono essere interi positivi. Il numero di righe della matrice non deve essere minore del numero di processi. Il prodotto tra il numero di processi p e il numero di thread t deve essere pari a numcore .

ESEMPIO DI FUNZIONAMENTO

Nell'immagine seguente vi è un esempio di funzionamento, con 2 processi su due nodi MPI diversi con 2 thread, matrice di dimensione 7×8 e dimensione del vettore pari a 8.

```
%esempio di funzionamento
funzionamento
```

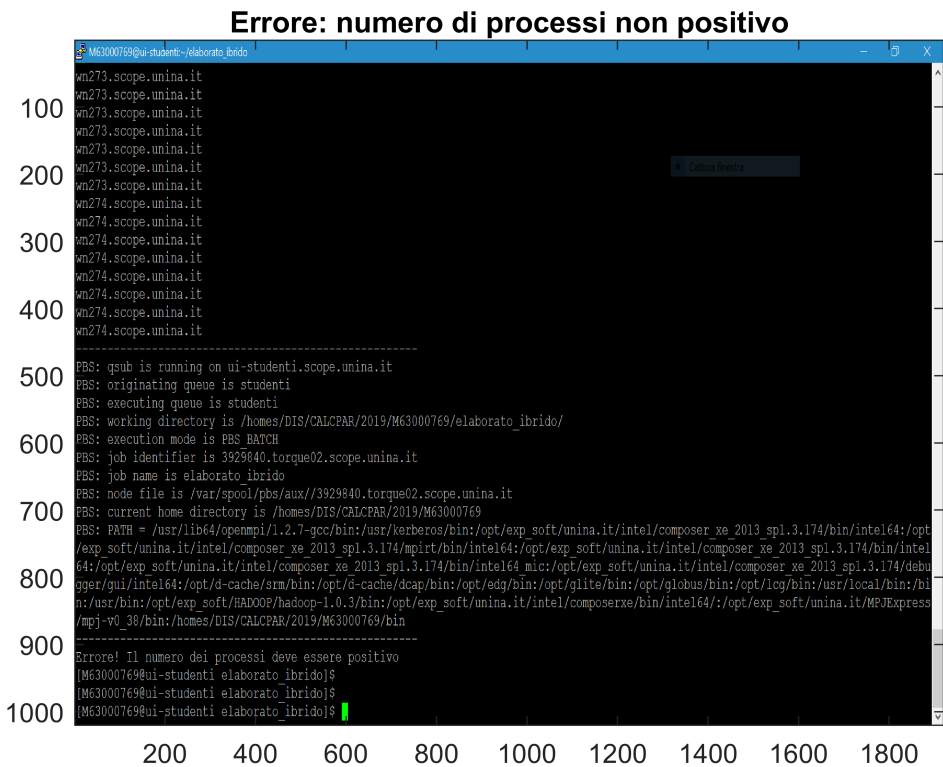
Esempio di funzionamento con 2 processi 2 thread e dimensione della matrice 7

```
M63000769@ui-studenti:~/elaborato_ibrido
-----
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/elaborato_ibrido/
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3929835.torque02.scope.unina.it
PBS: job name is elaborato_ibrido
PBS: node file is /var/spool/pbs/aux//3929835.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/mpirt/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/gul/intel64:/opt/d-cache/srm/bin:/opt/d-cache/dcapi/bin:/opt/edg/bin:/opt/glibe/bin:/opt/globus/bin:/opt/icc/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/mpj-v0_38/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
-----
Ciao sono il processo 0 e mi chiamo wn273.scope.unina.it
(Thread 0) Il programma sarà eseguito con 2 threads per processo
Ciao sono il processo 1 e mi chiamo wn274.scope.unina.it
(Thread 0) Il programma sarà eseguito con 2 threads per processo
Sono il processo 0. Il vettore risultato di 7 righe è :
Stampa array del processo 0:
25.1327412287183449
50.2654824574366899
75.3982236861550206
100.5309649148733797
125.6637061435917389
150.7964473723100411
175.9291886010284429
Sono il processo 0: tempo di esecuzione totale di 0.0015048980712891 secondi.
(M63000769@ui-studenti elaborato_ibrido)$
(M63000769@ui-studenti elaborato_ibrido)$
(M63000769@ui-studenti elaborato_ibrido)$
(M63000769@ui-studenti elaborato_ibrido)$
```

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



100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

Errore: la dimensione di riga della matrice è inferiore al numero di processi

```

MC63000769@ui-studenti-elaborato_ibrido
Job is running on node(s):
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
vm274.scope.unina.it
-----
PBS: qsub is running on ui-studenti.scope.unina.it
PBS: originating queue is student1
PBS: executing queue is student1
PBS: working directory is /homes/DIS/CALCPAR/2019/M63000769/elaborato_ibrido/
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3929859.torque02.scope.unina.it
PBS: job name is elaborato_ibrido
PBS: node file is /var/spool/pbs/aux/3929859.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_sp1.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/mpirt/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/dbugger/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:/bin:/usr/bin:/opt/exp_soft/HADOOP/hadoop-1.0.3/bin:/opt/exp_soft/unina.it/intel/composere/bin/intel64:/opt/exp_soft/unina.it/WP3Express/mpj-v0_38/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
-----
Errore! La dimensione di riga della matrice è inferiore al numero di processori.
[M63000769@ui-studenti elaborato_ibrido]$

```

Errore: le dimensioni non sono coerenti

```

M63000769@ui-studenti:~elaborato_ibrido
Job is running on node(s):
100  xn273.scope.unina.it
    xn273.scope.unina.it
    xn273.scope.unina.it
    xn273.scope.unina.it
200  xn273.scope.unina.it
    xn273.scope.unina.it
    xn273.scope.unina.it
    xn273.scope.unina.it
300  xn274.scope.unina.it
    xn274.scope.unina.it
    xn274.scope.unina.it
    xn274.scope.unina.it
400  xn274.scope.unina.it
    xn274.scope.unina.it
    xn274.scope.unina.it
    xn274.scope.unina.it
500  xn274.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/elaborato_ibrido/
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3929862.torque02.scope.unina.it
700  PBS: job name is elaborato_ibrido
PBS: node file is /var/spool/pbs/aux//3929862.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_sp1.3.174/bin/intel64:/opt
800  /exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/mpirt/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/bin/intel
64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/bin/intel64_mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/dsbu
gger/gui/intel64:/opt/d-cache/stm/bin:/opt/d-cache/dcap/bin:/opt/edg/bin:/opt/glite/bin:/opt/globus/bin:/opt/lcg/bin:/usr/local/bin:/bi
900  n:/usr/bin:/opt/exp_soft/HADOOP/hadoop-1.0.3/bin:/opt/exp_soft/unina.it/intel/composer_xe/bin/intel64:/opt/exp_soft/unina.it/MPJExpress
/mpj-v0_38/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
-----
Errore! Le dimensioni non sono coerenti
[M63000769@ui-studenti elaborato_ibrido]$
1000

```

100
200
300
400
500
600
700
800
900
1000

200 400 600 800 1000 1200 1400 1600 1800

100
200
300
400
500
600
700
800
900
1000

200 400 600 800 1000 1200 1400 1600 1800

ANALISI DELLE DIVERSE CONFIGURAZIONI

Lo scopo di questo elaborato è individuare la migliore configurazione del numero di processi e thread: supponiamo che nel nostro caso l'architettura sia composta da 2 nodi MPI ciascuno formato da 2 cpu-socket di 4 core (pertanto si ha a disposizione un totale di 16 core) e si cerca di trovare la configurazione migliore per risolvere il prodotto matrice vettore su una matrice di dimensione 20000×15000 .

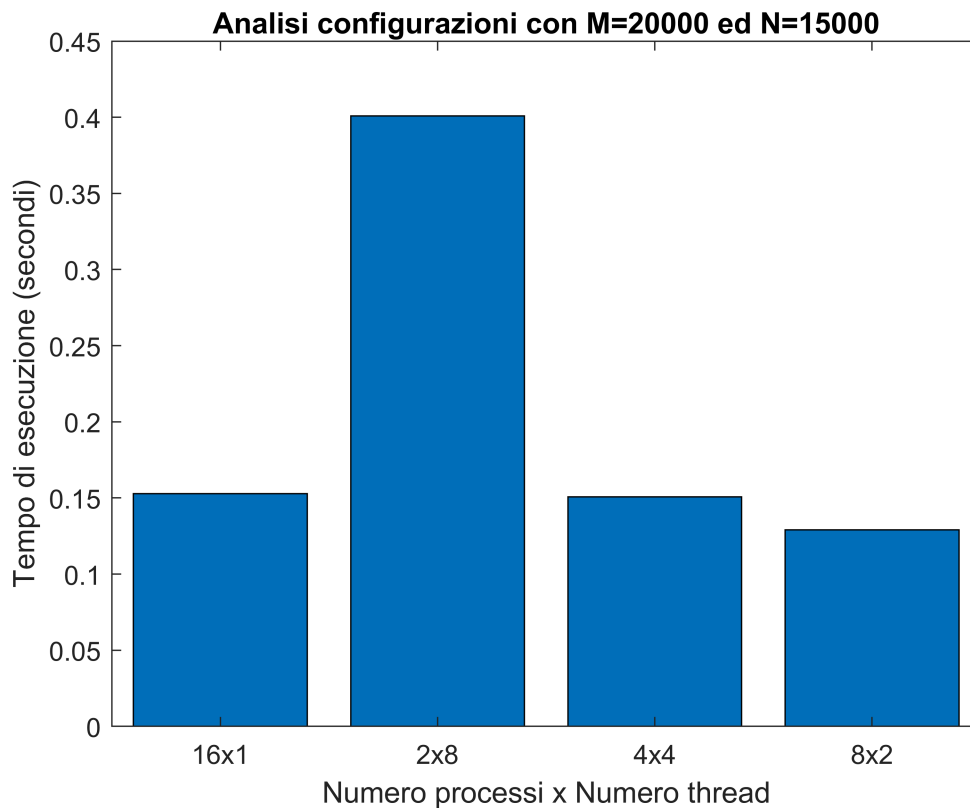
A differenza degli elaborati precedenti, lo scopo non è quindi quello di studiare la scalabilità del problema (analizzare le caratteristiche dell'algoritmo quali efficienza, speedup ecc al variare della dimensione del problema e del numero di processi), ma quello di trovare la configurazione ottimale una volta fissate la dimensione del problema ed il numero di processi e thread.

Per configurazione ottimale si intende quella che minimizza l'overhead dovuto alle comunicazioni MPI e quello di gestione dei thread OpenMP rispetto al tempo di calcolo parallelo.

Le quattro alternative, con il numero di nodi fissato a 2, che abbiamo scelto di considerare sono:

- 16 processi single-thread;
- 8 processi con 2 thread;
- 4 processi con 4 thread;
- 2 processi con 8 thread.

```
%script per il grafico  
prestazioni
```



Analizzando l'istogramma ottenuto la configurazione che ha i tempi peggiori è quella composta da 2 processi con 8 thread mentre la migliore è quella composta da 8 processi con 2 thread. Le altre due configurazioni ottengono risultati intermedi.

Una possibile spiegazione del risultato è che, essendo la matrice distribuita per righe, conviene sfruttare al massimo la divisione per righe (e cioè il numero di processi) con un numero limitato di thread per non far pesare troppo l'overhead di creazione e gestione dei thread.

ANALISI DELL' ACCURATEZZA

Confrontando il risultato ottenuto sul cluster Scope e quello ottenuto su MATLAB si ottiene il seguente errore relativo (tramite il comando `norm`), fissando a 2 il numero di processi con 2 thread con dimensione della matrice pari a 7x8.

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

```
risultato_scope = 7x1
102 ×
```



```

0.251327412287183
0.502654824574366
0.753982236861550
1.005309649148730
1.256637061435910
1.507964473723100
1.759291886010280
risultato_matlab = 7×1
102 ×
0.251327412287183
0.502654824574367
0.753982236861550
1.005309649148734
1.256637061435917
1.507964473723100
1.759291886010284
errore_relativo =
3.210969507161735e-15

```