

# ELABORATO 2 - PRODOTTO MATRICE VETTORE PER COLONNE

LO BRUTTO FABIO / MAIONE PAOLO

## DEFINIZIONE DEL PROBLEMA

Si vuole progettare un algoritmo in MPI per risolvere il prodotto vettoriale tra una matrice, di dimensioni  $M \times N$ , e un vettore di reali, di dimensione  $N$ , su  $p$  processori.

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\_2.c*, sono:

- 1) Distribuzione per colonne della matrice in  $p$  processori e distribuzione del vettore: ognuno dei  $p$  processori eseguirà il prodotto vettoriale tra la porzione di matrice ricevuta dal processo *root* (cioè quello con rank 0), e la porzione del vettore;
- 2) Elaborazione del prodotto vettoriale in parallelo ;
- 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_Reduce()`).

Inoltre l'algoritmo progettato comprende anche il caso in cui la dimensione  $N$  (cioè il numero di colonne della matrice) non sia multipla del numero di processori  $p$  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`.
- **Output:** il vettore risultato del prodotto vettoriale tra la matrice e il vettore.



```
%un esempio per ciascuna condizione di errore
errori
```

**Errore: numero di processori non positivo**

[illegible]

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

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

```
M63000769@ui-studenti-elaborati/elaborato_2/prodotto_matrice_vettore_colonne
-----
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$ cat elaborato_2.out
-----
This job is allocated on 4 cpu(s)
Job is running on node(s):
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm273.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/elaborato_2/prodotto_matrice_vettore_colonne
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3923317.torque02.scope.unina.it
PBS: job name is elaborato_2
PBS: node file is /var/spool/pbs/aux//3923317.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/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/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
-----
Esegui: /usr/lib64/openmpi/1.4-gcc/bin/mpicc -o /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_2/prodotto_matrice_vettore_colonn
e/elaborato_2 /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_2/prodotto_matrice_vettore_colonne/elaborato_2.c
Esegui: /usr/lib64/openmpi/1.4-gcc/bin/mpixec -machinefile /var/spool/pbs/aux//3923317.torque02.scope.unina.it -np 4 /homes/DIS/CALCPA
R/2019/M63000769/elaborati/elaborato_2/prodotto_matrice_vettore_colonne/elaborato_2
Errore! La dimensione di colonna della matrice è inferiore al numero di processori.
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
```

## Errore: le dimensioni non sono coerenti

```
M63000769@ui-studenti-elaborati/elaborato_2/prodotto_matrice_vettore_colonne
-----
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$ cat elaborato_2.out
-----
This job is allocated on 4 cpu(s)
Job is running on node(s):
vm273.scope.unina.it
vm273.scope.unina.it
vm273.scope.unina.it
vm273.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/elaborato_2/prodotto_matrice_vettore_colonne
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3922635.torque02.scope.unina.it
PBS: job name is elaborato_2
PBS: node file is /var/spool/pbs/aux//3922635.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/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/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
-----
Esegui: /usr/lib64/openmpi/1.4-gcc/bin/mpicc -o /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_2/prodotto_matrice_vettore_colonn
e/elaborato_2 /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_2/prodotto_matrice_vettore_colonne/elaborato_2.c
Esegui: /usr/lib64/openmpi/1.4-gcc/bin/mpixec -machinefile /var/spool/pbs/aux//3922635.torque02.scope.unina.it -np 4 /homes/DIS/CALCPA
R/2019/M63000769/elaborati/elaborato_2/prodotto_matrice_vettore_colonne/elaborato_2
Errore! le dimensioni non sono coerenti
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
[M63000769@ui-studenti prodotto_matrice_vettore_colonne]$
```

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

## Tempo di esecuzione - T(p)

Si è scelto di misurare i tempi di esecuzione nel processo *root* usando la primitiva `MPI_Wtime()`. In particolare l'intervallo di tempo misurato è quello che comprende le fasi 2 e 3 dell'algoritmo prima citate.

Si è scelto inoltre di considerare tre casi diversi che rappresentano le tre possibilità per le dimensioni della matrice: il caso in cui il numero di righe è maggiore numero di colonne, il caso in cui il numero di righe è minore del numero di colonne ed, infine, il caso in cui la matrice è quadrata.

Per ciascuno di questi tre casi è stato considerato il minimo tra 3 esecuzioni ripetute, eseguite in momenti diversi ed in particolare:

- quando il numero di righe M è maggiore del numero di colonne N si fa variare M da 5000 a 50000 con N fissato a 1000
- quando il numero di righe M è minore del numero di colonne N si fa variare N da 5000 a 50000 con M fissato a 1000
- quando la matrice è quadrata si fa variare  $M=N$  da 100 a 10000

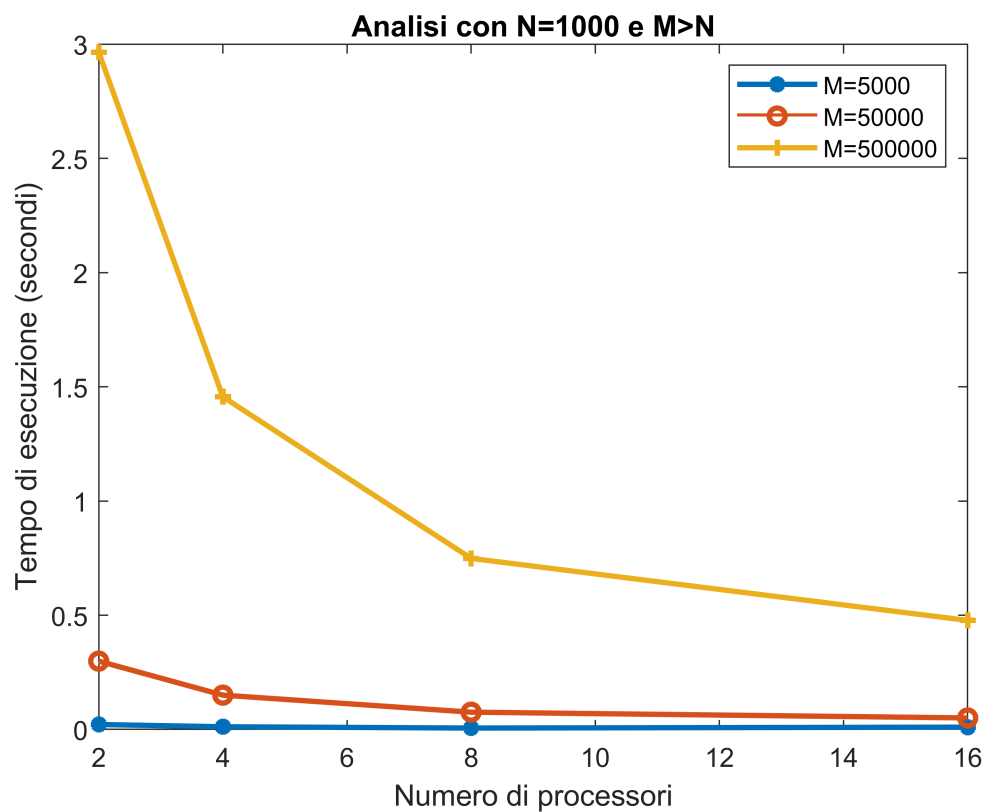
Si noti che nel caso di matrici quadrate si è scelta una diversa configurazione rispetto al caso di matrici sbilanciate data l'impossibilità di eseguire sul cluster il programma con dimensioni 50000 x 50000. Pertanto è parso ragionevole, solo nel caso di matrici quadrate, ridurre la dimensione massima. Inoltre, sebbene test con matrici 100x100 risultino poco significativi, si è scelto di utilizzare le stesse dimensioni dell'elaborato per righe, per facilitare il confronto.

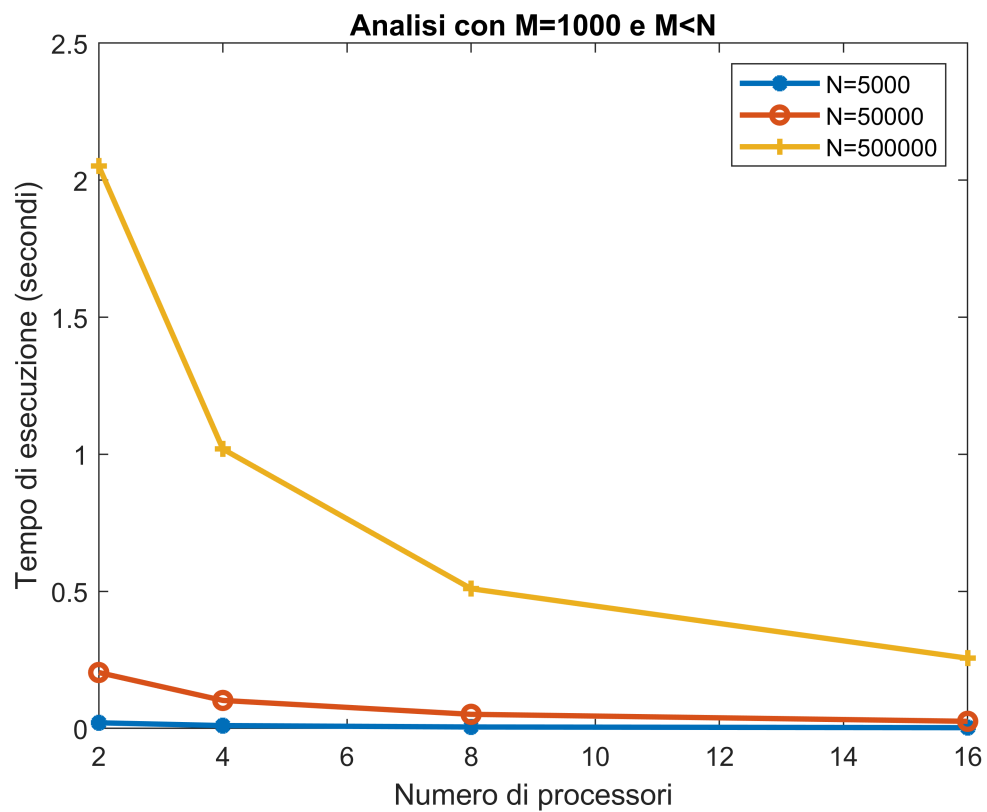
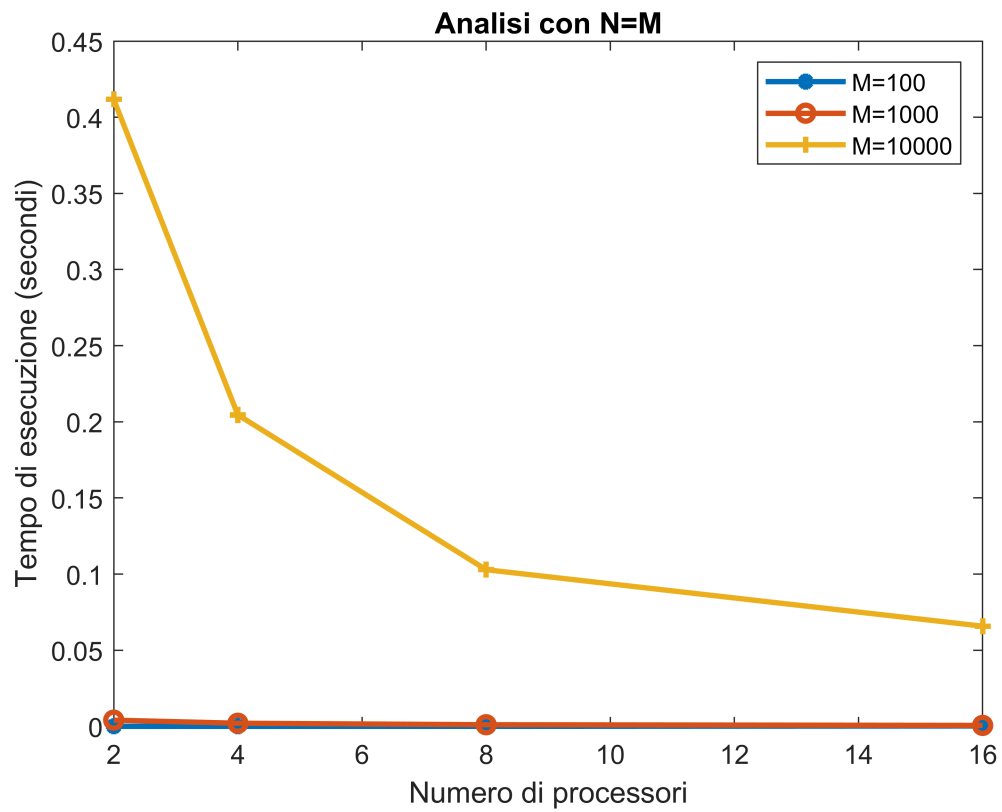
I grafici e le tabelle riassumono i risultati ottenuti per tutti i possibili valori di M,N e p.

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

### Tempi

PROVA M>N	N=1000		
	5000	50000	500000
2	0,0214601287841797	0,2989480590820310	2,964836835861200
4	0,0115590820312500	0,1500309028625480	1,456056118011470
8	0,0059680938720703	0,0754148960113525	0,748605966567993
16	0,0097928047180176	0,0502231121063232	0,477081060409545
PROVA M=N			
	100	1000	10000
2	0,0000560283660889	0,0041019916534424	0,411716938018798
4	0,0000460147857666	0,0020821094512939	0,204568147659301
8	0,0000338554382324	0,0010998249053955	0,102904081344604
16	0,0002830028533936	0,0006630420684814	0,065809976577759
PROVA M<N	M=1000		
	5000	50000	500000
2	0,0204608440399170	0,203956842422485	2,051769018173210
4	0,0103078914642334	0,102041006088256	1,019224882125850
8	0,0051660537719727	0,051406860351563	0,509707927703857
16	0,0026650428771973	0,025868146896362	0,255983829498291





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



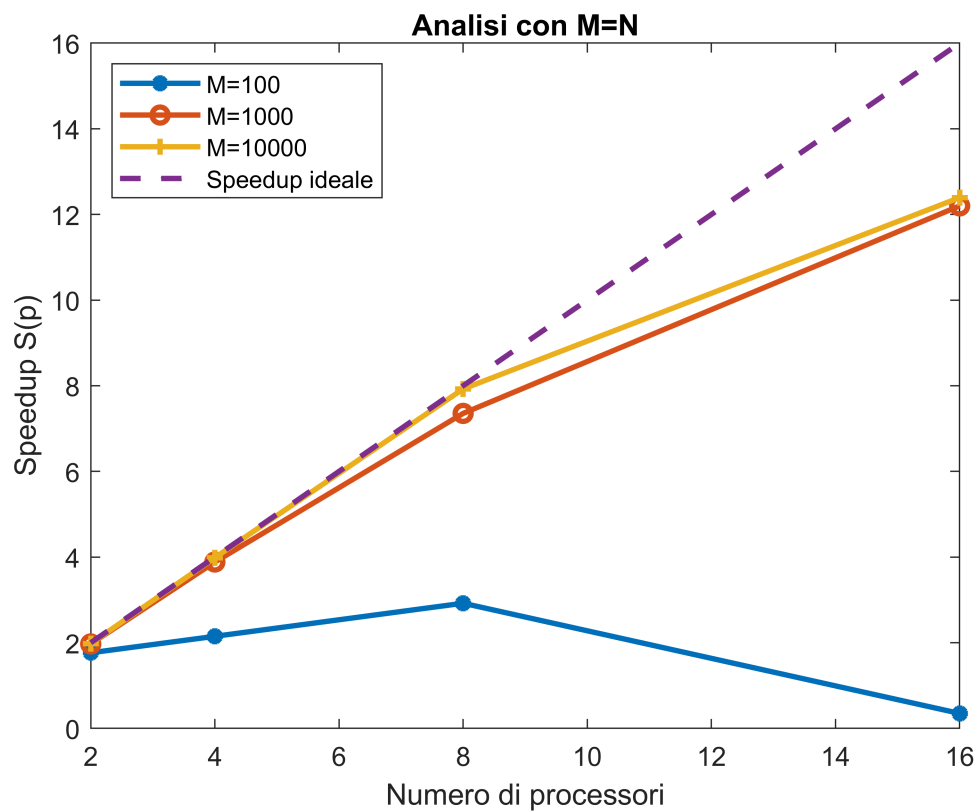
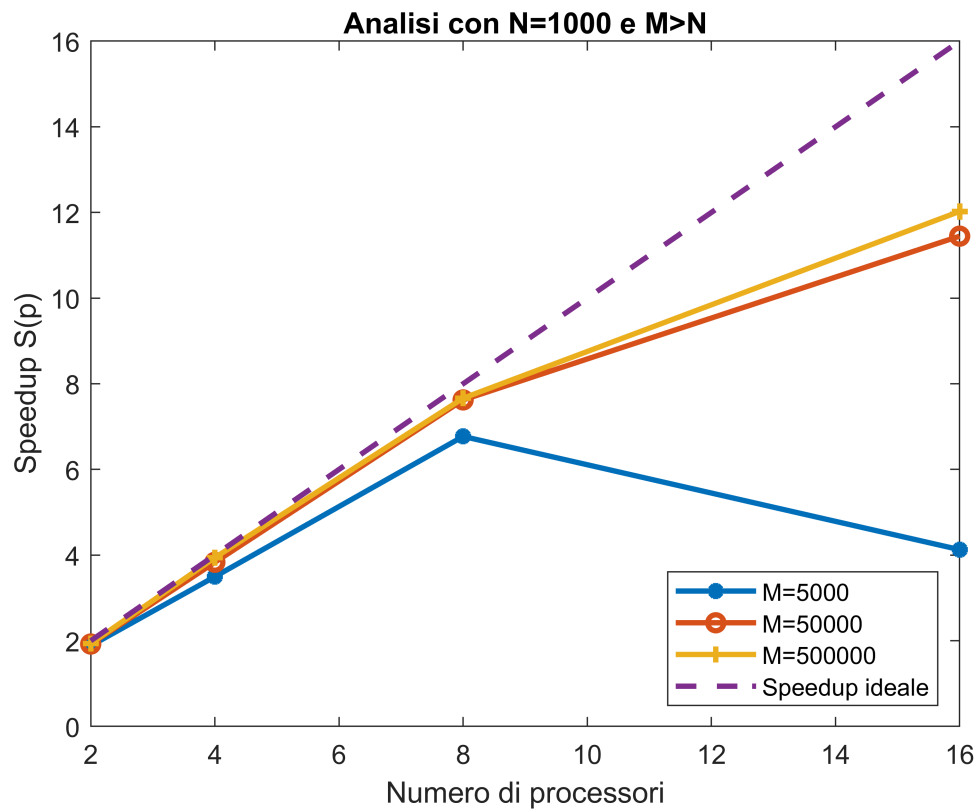
## Speed up ed Efficienza - $S(p)$ ed $E(p)$

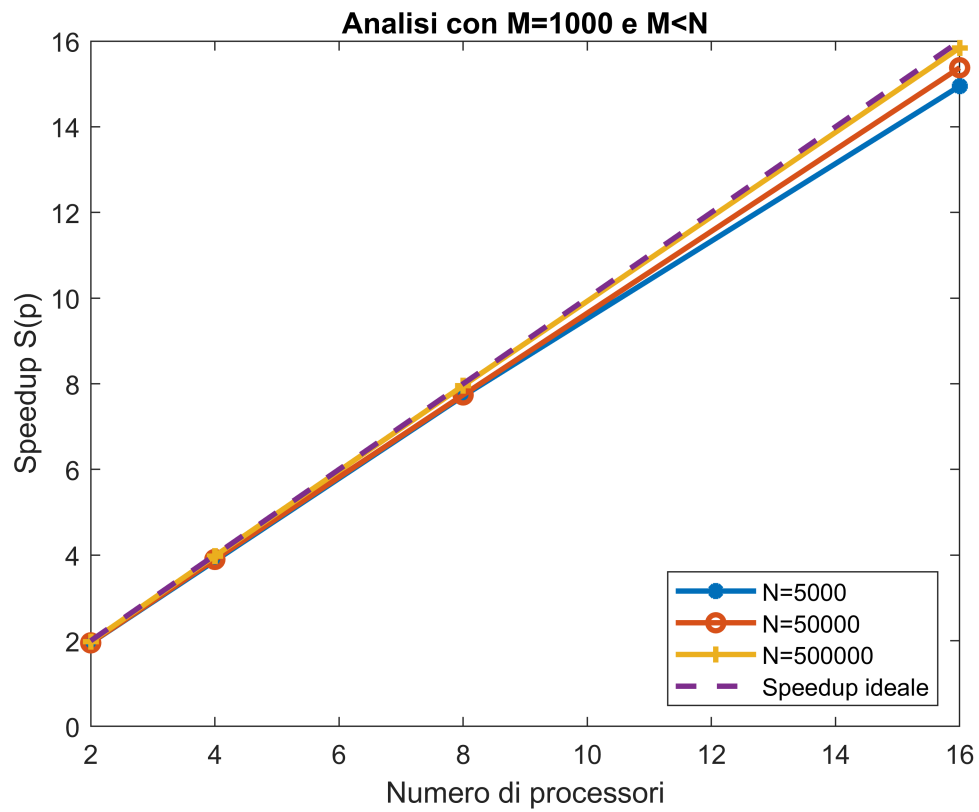
Si è calcolato, inoltre, il tempo di riferimento  $T(1)$  che corrisponde al tempo di esecuzione su un unico processore.

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

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

Speed up			
PROVA $M > N$	N=1000		
	5000	50000	500000
2	1,8829	1,9230	1,9348
4	3,4957	3,8316	3,9396
8	6,7705	7,6227	7,6627
16	4,1262	11,4462	12,0238
PROVA $M=N$			
	100	1000	10000
2	1,7660	1,9720	1,9809
4	2,1503	3,8850	3,9867
8	2,9225	7,3549	7,9254
16	0,3496	12,1999	12,3925
PROVA $M < N$	M=1000		
	5000	50000	500000
2	1,9471	1,9506	1,9766
4	3,8650	3,8988	3,9789
8	7,7119	7,7389	7,9564
16	14,9491	15,3793	15,8425



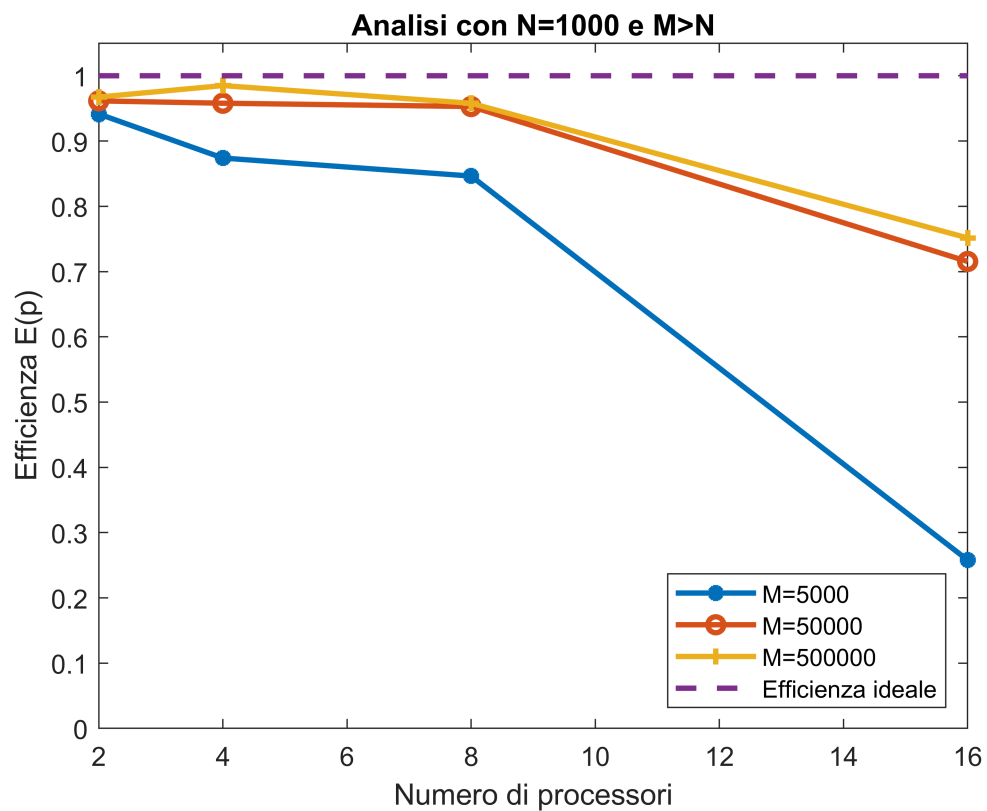


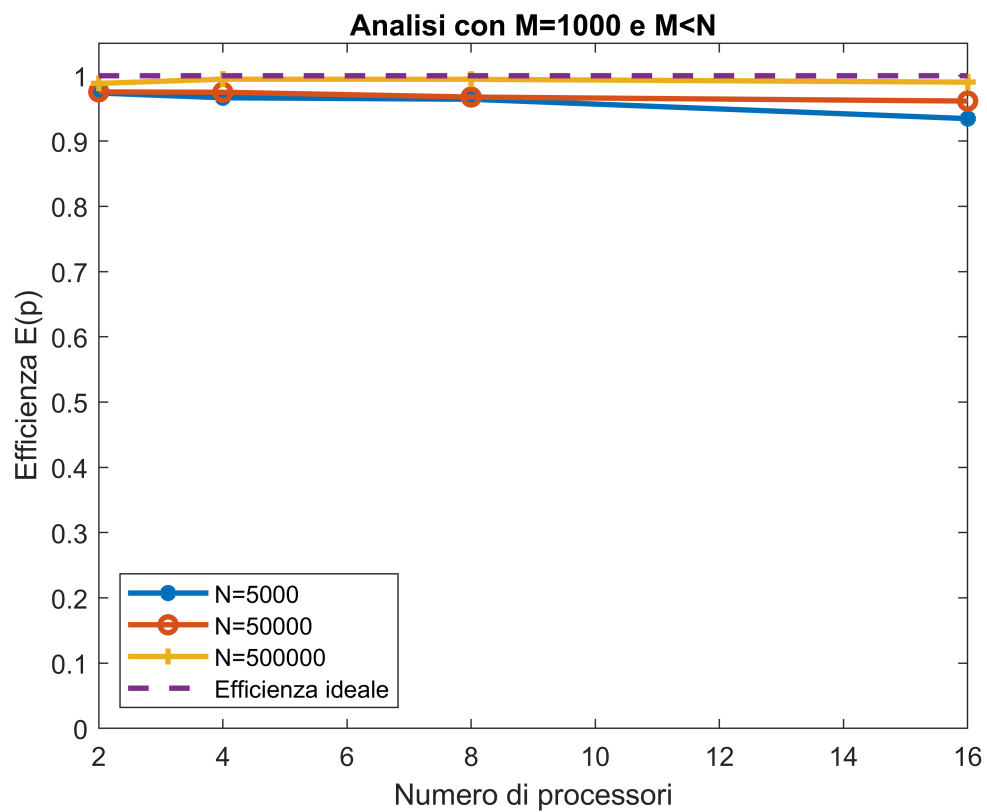
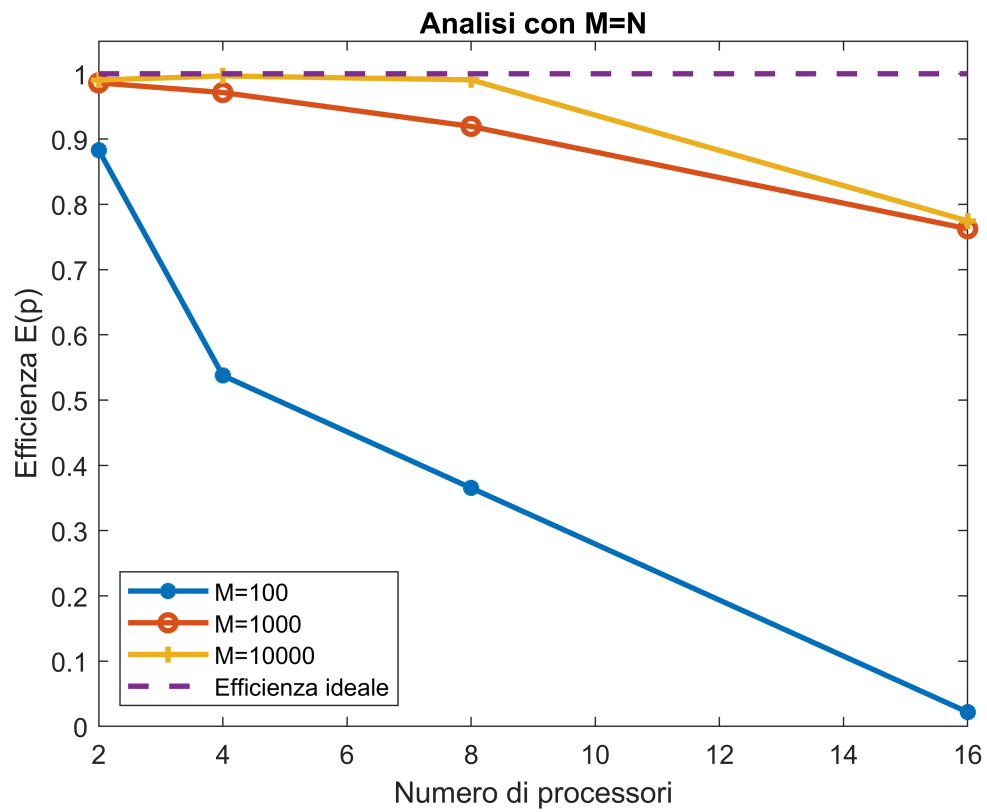
Infine si è calcolata l'efficienza rapportando lo speed-up  $S(p)$  al numero di processori  $p$ .

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

## Efficienza

PROVA M>N	N=1000		
	5000	50000	500000
2	0,9414	0,9615	0,9674
4	0,8739	0,9579	0,9849
8	0,8463	0,9528	0,9578
16	0,2579	0,7154	0,7515
PROVA M=N			
	100	1000	10000
2	0,8830	0,9860	0,9904
4	0,5376	0,9713	0,9967
8	0,3653	0,9194	0,9907
16	0,0219	0,7625	0,7745
PROVA M<N	M=1000		
	5000	50000	500000
2	0,9736	0,9753	0,9883
4	0,9662	0,9747	0,9947
8	0,9640	0,9674	0,9945
16	0,9343	0,9612	0,9902





## Conclusioni

Dai grafici e dalle tabelle appena presentate si possono trarre le seguenti conclusioni:

- se la matrice è sbilanciata, l'efficienza migliore si ha per 2 processori quando la maggiore tra le due dimensioni è 5000 oppure 50000. Invece il numero ottimo di processori risulta pari a 4 quando è 500000;
- l'efficienza risulta complessivamente maggiore nella configurazione in cui  $M < N$ ;
- anche nel caso di matrici quadrate si ha che l'efficienza migliore si ha per due processori fino a  $M=N=1000$  mentre è 4 per  $M=N=10000$ .

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

Analoghe considerazioni per i tempi e lo speedup.

## 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 4 il numero di processori con dimensione della matrice pari a  $7 \times 8$ .

In particolare, nel test seguente, sono stati usati un vettore e una matrice inizializzati nel modo seguente

$$\text{matrice}[i][j] = \pi * (i + 1)$$

$$\text{vettore}[i] = 1$$

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

```
risultato_scope = 7x1
    25.1327
    50.2655
    75.3982
   100.5310
   125.6637
   150.7964
   175.9292
risultato_matlab = 7x1
    25.1327
    50.2655
    75.3982
   100.5310
```

```
125.6637
150.7964
175.9292
errore_relativo = 3.2110e-15
```