

CALCOLO PARALLELO

a.a. 2006/2007

Dense Matrix-Matrix multiplication in parallel



Gruppo: M³

Boldrin Mattia (809585)

Bortolozzo Matteo (810408)

Lionello Marco (810079)



STRUTTURA DEL SOFTWARE

ALGORITMI & TEST

Serial Multiplication

Matrix-Matrix Multiplication

Cannon Algorithm

Processor Farm

AUTOMATIZZAZIONE

PROBLEMI RISCONTRATI

GRAFICI



Diapositiva 3

MB2

Io toglierei questa SLIDE. Non offre nessuna informazione utile.

Mattia Boldrin; 10/10/2007

SERIAL MULTIPLICATION

Algoritmo per la moltiplicazione seriale di due matrici.

Complessità algoritmica $O(n^3)$

$$(A \cdot B)_{ij} = \sum_{k=1}^p a_{ik} \cdot b_{kj},$$

$$1 \leq i \leq n \quad , \quad 1 \leq j \leq m \quad .$$

Il tempo di computazione di un prodotto sequenziale viene usato nel calcolo dello SpeedUp.



SERIAL MULTIPLICATION

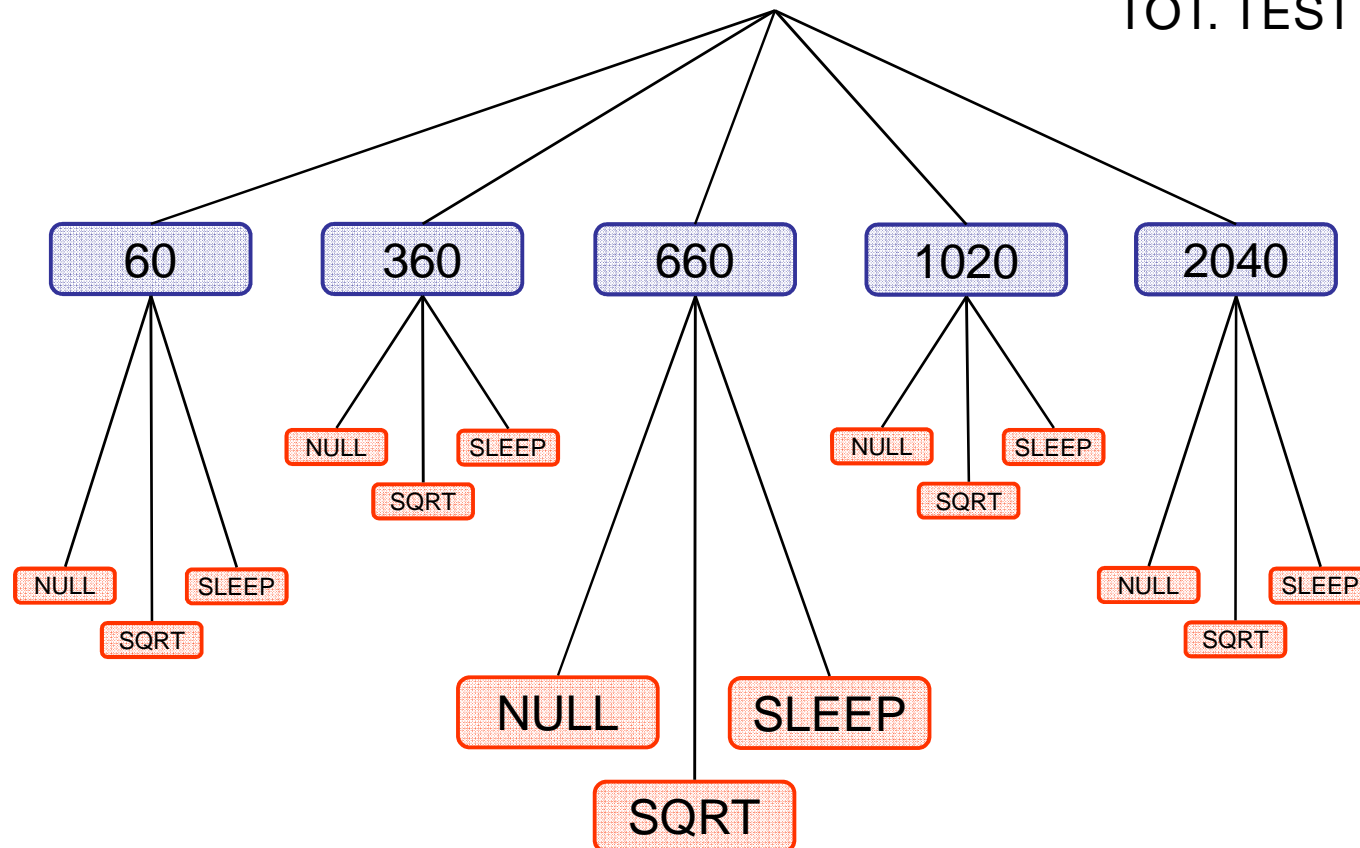
DIMENSIONE DELLE MATRICI

AUMENTO DEL
CALCOLO COMPUTAZIONALE

TEST MOLTIPLICAZIONE SERIALE

(METODO 0)

TOT. TEST = 15



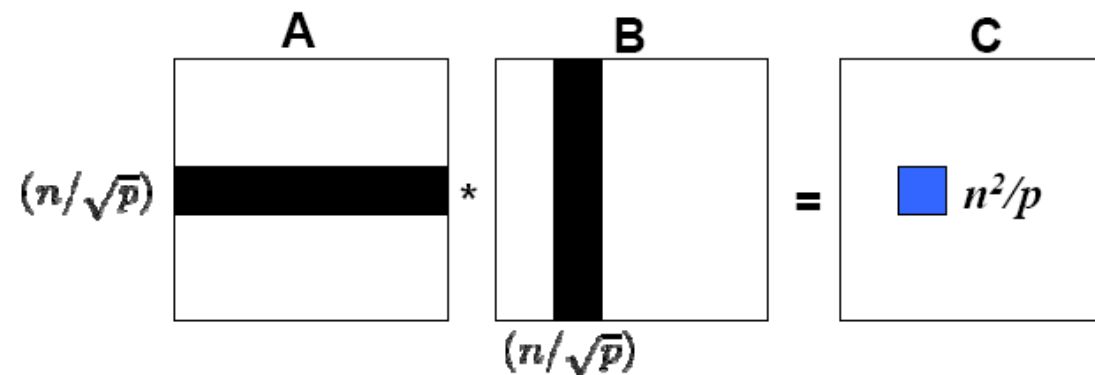
CALCOLO PARALLELO

Dense Matrix-Matrix multiplication in parallel

MATRIX-MATRIX MULT.

PREMESSE

- Si calcola il numero di righe (e colonne) che compongono ciascun TASK (n/\sqrt{p});
- Si invia un TASK-riga e un TASK-colonna ad ogni Worker;
- Anche il MASTER esegue una computazione locale;



MATRIX-MATRIX MULT.

Funzionamento dell'algoritmo

MASTER

- Determina il numero di processi avviati e la dimensione dei task.
- Crea e spedisce il vettore di configurazione .
- Invia a tutti i Worker i dati necessari alla computazione (task-riga, task-colonna).
- Computa una parte del lavoro localmente.
- Riceve la sottomatrice C computata da ciascun Worker.
- Inserisce ciascuna sotto-matrice C computata nella matrice prodotto finale

WORKER

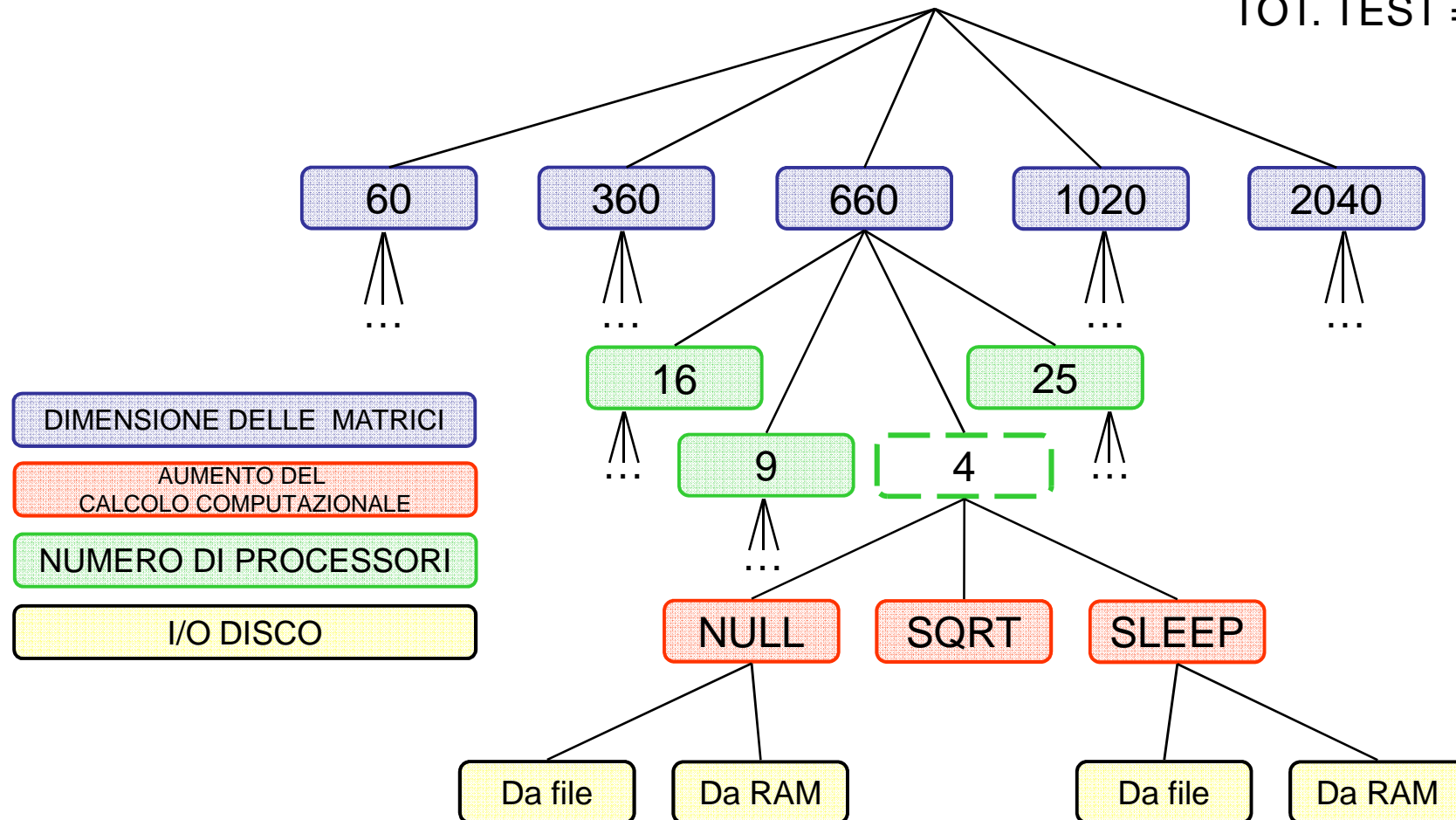
- Ricezione del vettore di configurazione.
- Ricezione del task-riga e task-colonna.
- Computazione del sottoproblema.
- Invio sottomatrice C computata.

MATRIX-MATRIX MULT.

TEST MATRIX-MATRIX MULTIPLICATION

(METODO 1)

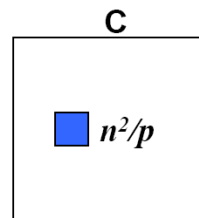
TOT. TEST = 70



CANNON ALGORITHM

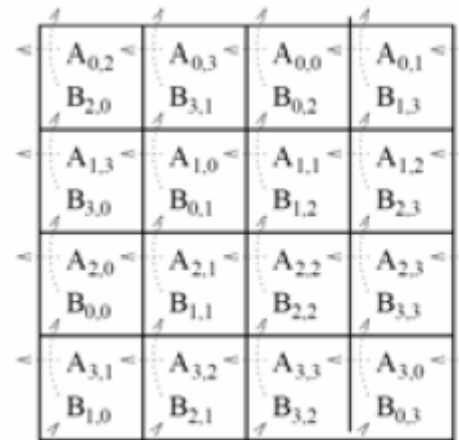
PREMESSE

- Algoritmo basato su permutazioni e somme parziali
- Il Master è parte integrante del calcolo distribuito
- Il prodotto totale è ottenuto in modo incrementale dopo ogni permutazione

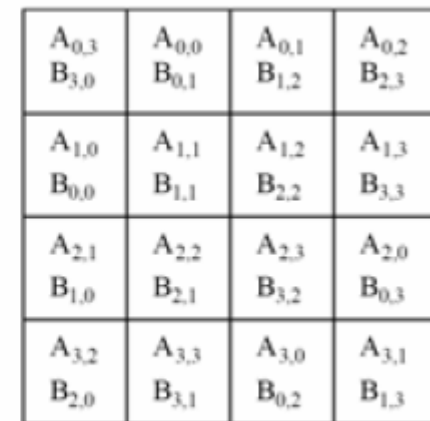


Dimensione Task

Esempio di funzionamento



(e) Submatrix locations after second shift



(f) Submatrix locations after third shift

CANNON ALGORITHM

Funzionamento dell'algoritmo

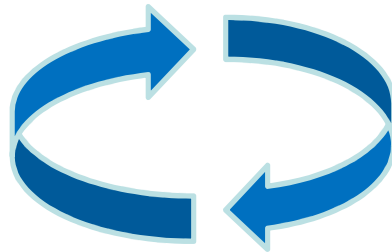
MASTER

- Determina il numero di processi avviati e la dimensione di ciascun task
- Crea ed invia il vettore di configurazione
- Esegue la prima permutazione
- Invia a tutti i worker i dati necessari alla computazione

WORKER

- Ricezione del vettore di configurazione

- Ricezione delle sottomatrici A e B



- Computa un calcolo parziale della sottomatrice C
- Invia e riceve le sottomatrici A e B secondo permutazioni ben definite

- Ricrea la matrice C completa

- Spedisce la sottomatrice C computata

CANNON ALGORITHM

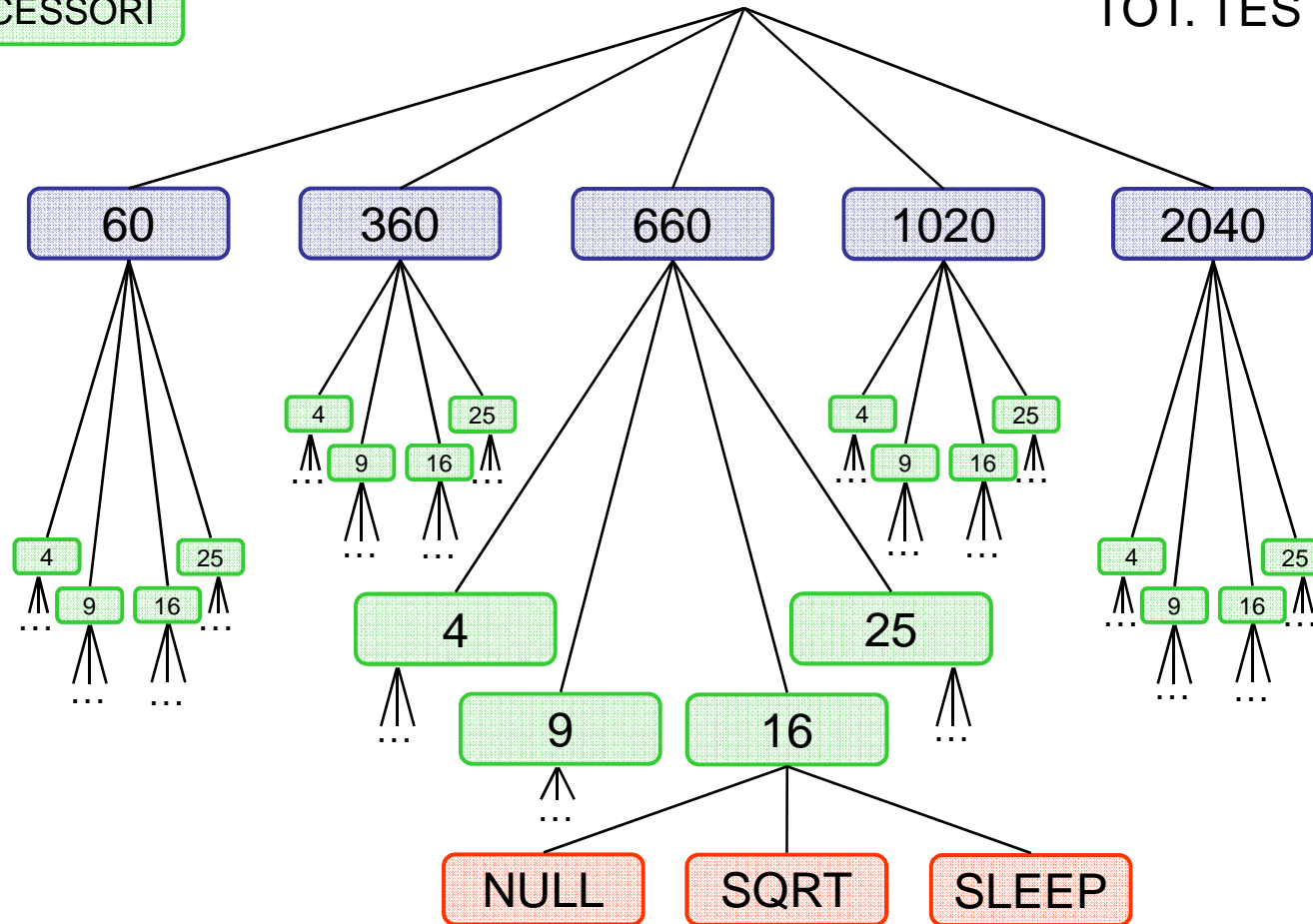
DIMENSIONE DELLE MATRICI

AUMENTO DEL
CALCOLO COMPUTAZIONALE

NUMERO DI PROCESSORI

TEST CANNON ALGORITHM (METODO 2)

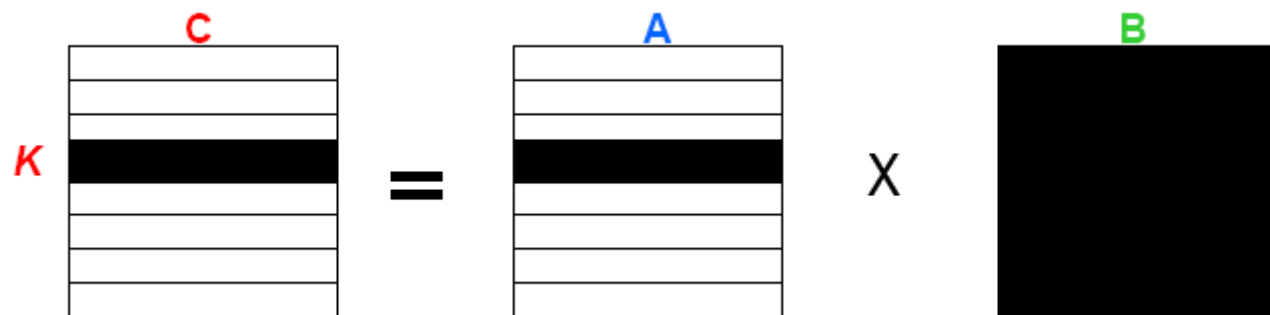
TOT. TEST = 60



PROCESSOR FARM

PREMESSE

- Si replica la matrice B su ogni Worker
- La dimensione del Task è definita come numero di righe di A da spedire a ciascun Worker ($K = n^2/v_p$)
- Il MASTER non computa
- La granularità può essere calcolata ad hoc dopo una serie di test



PROCESSOR FARM

Funzionamento dell'algoritmo

MASTER

- Determina il numero di processi avviati
- Crea ed invia il vettore di configurazione contenente la granularità impostata dall'utente

- Invia a tutti i client la matrice B ed un task-A

▪ Finché ci sono task da computare

- Riceve una sottomatrice C computata
 - Invia un altro task, se ce ne sono
-
- Matrice C ricreata completamente

WORKER

- Ricezione del vettore di configurazione

- Riceve la matrice B ed un task-A

- Il Worker i-esimo termina la computazione del suo task

- Il Worker i-esimo riceve un altro task

PROCESSOR FARM

DIMENSIONE DELLE MATRICI

AUMENTO DEL
CALCOLO COMPUTAZIONALE

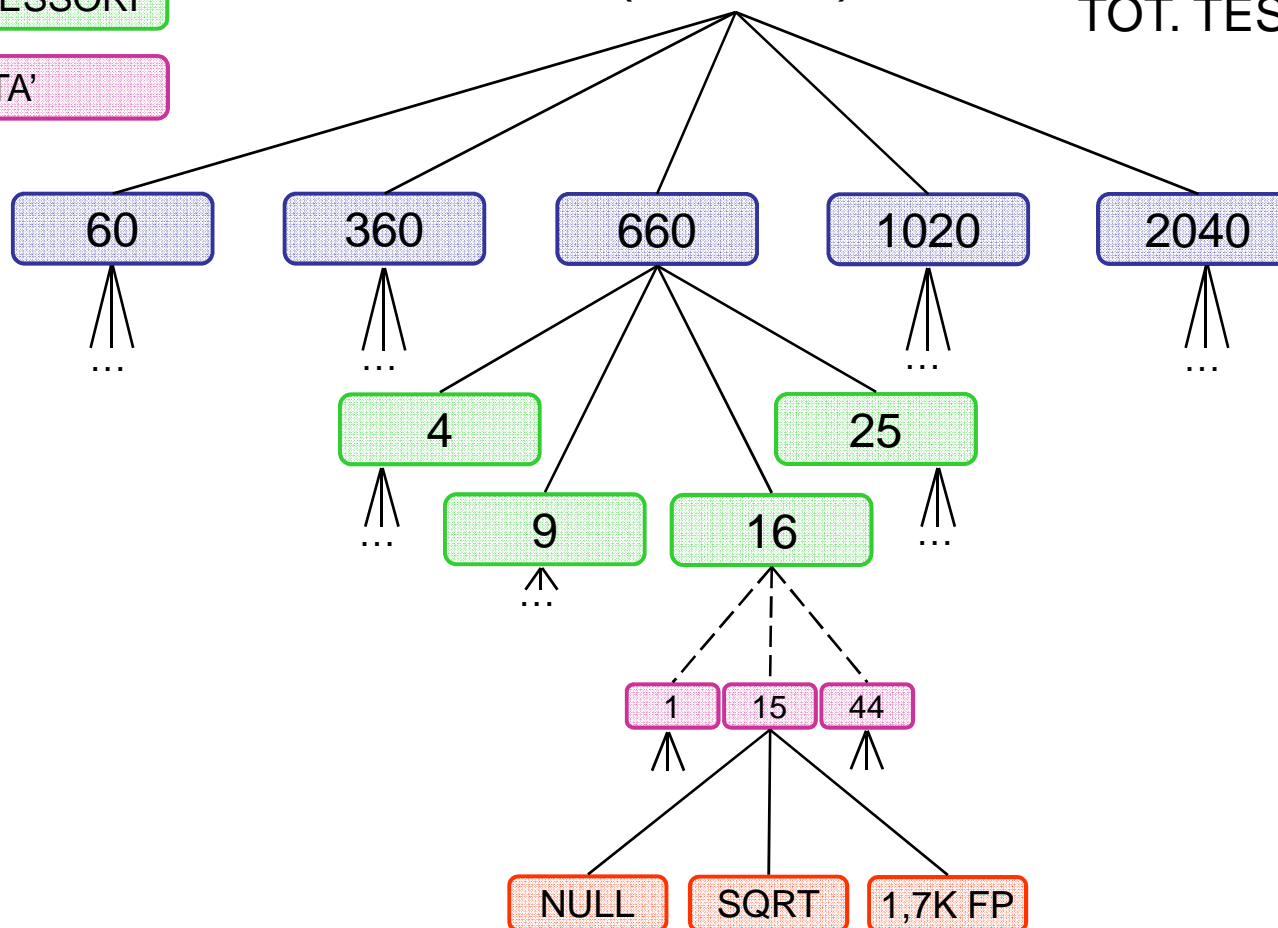
NUMERO DI PROCESSORI

GRANULARITA'

TEST PROCESSOR FARM

(METODO 3)

TOT. TEST = 180



AUTOMATIZZAZIONE



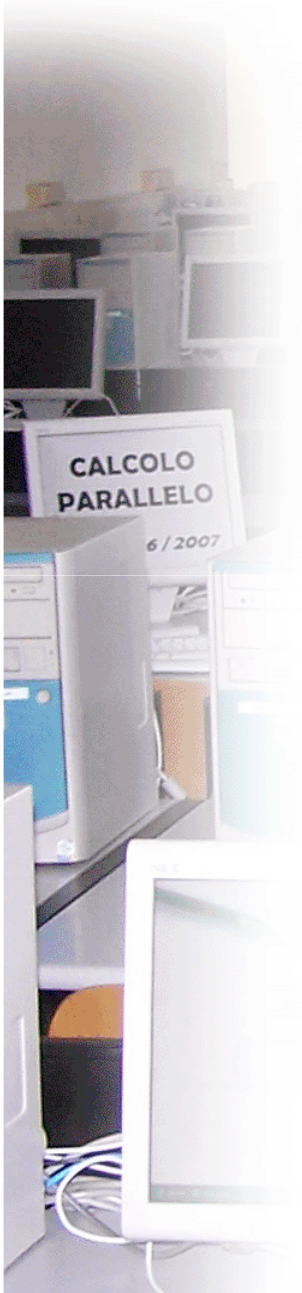
SCRIPT BASH

```
mpirun -np 4 -machinefile machinefile a.out 1 60 0 15 0 >> log.csv
```

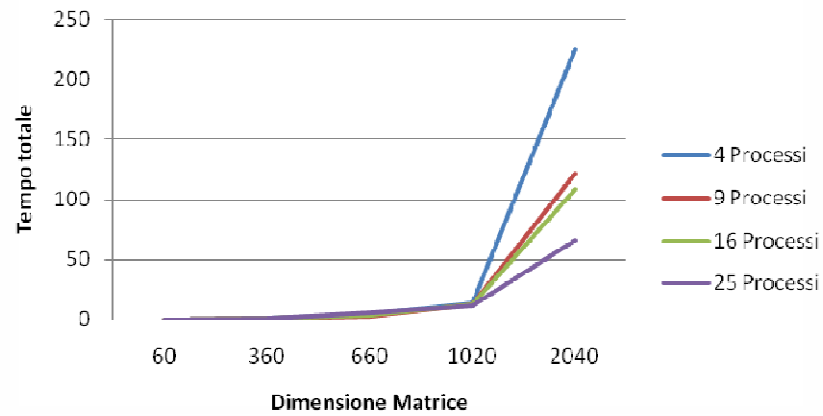


PROBLEMI RISCONTRATI

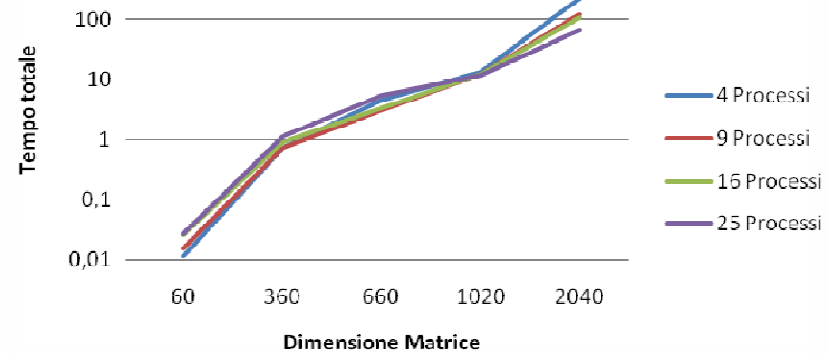
- Ordine nelle operazioni in FLOATING POINT
- Difficoltà nella pianificazione dei test
- Aria condizionata del laboratorio troppo alta



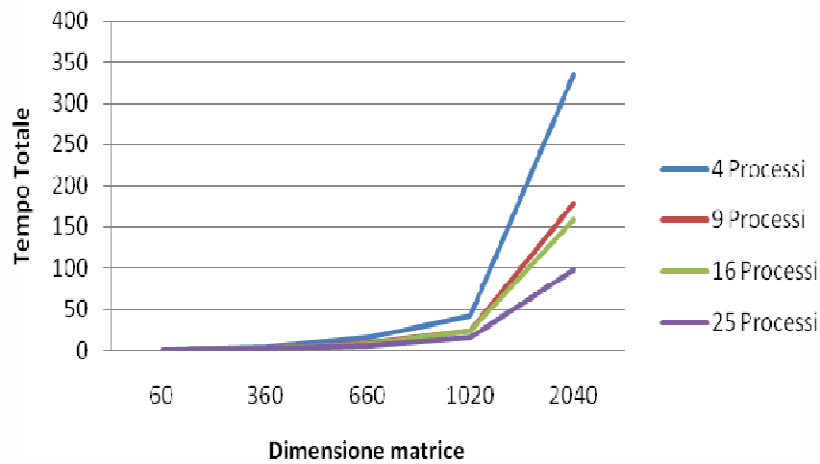
Metodo 1, Peso Computazionale 0



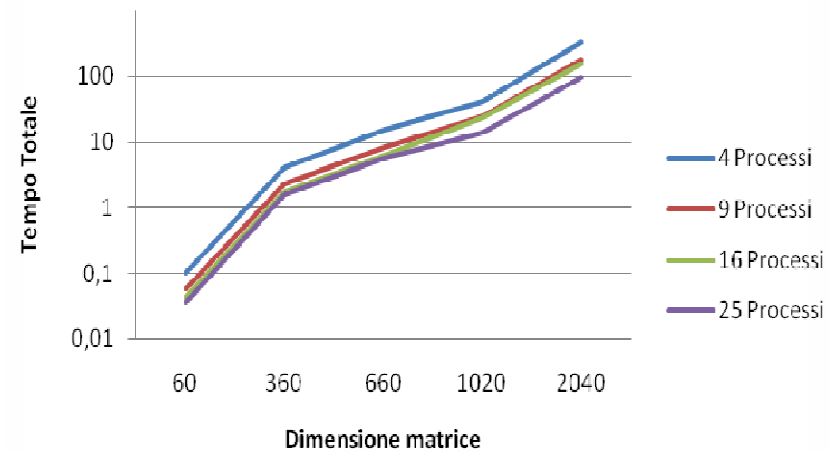
Metodo 1, Peso Computazionale 0



Metodo 1, Peso Computazionale 6



Metodo 1, Peso Computazionale 6



Diapositiva 17

MB3

9, 10, 27, 28

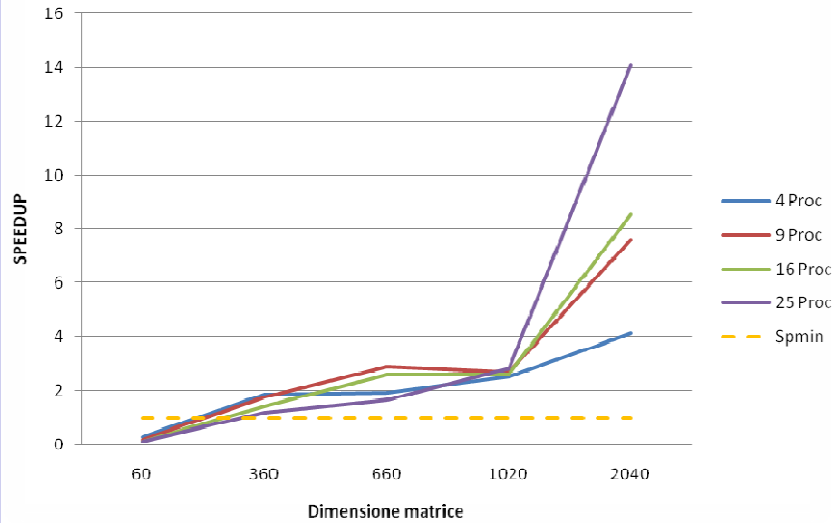
Mattia Boldrin; 13/10/2007

SpeedUp/Efficienza Metodo 1

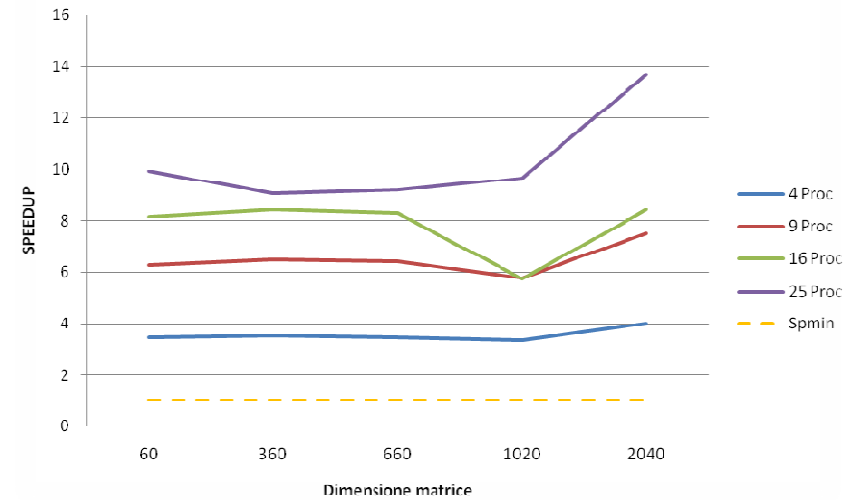
SpeedUp

Efficienza

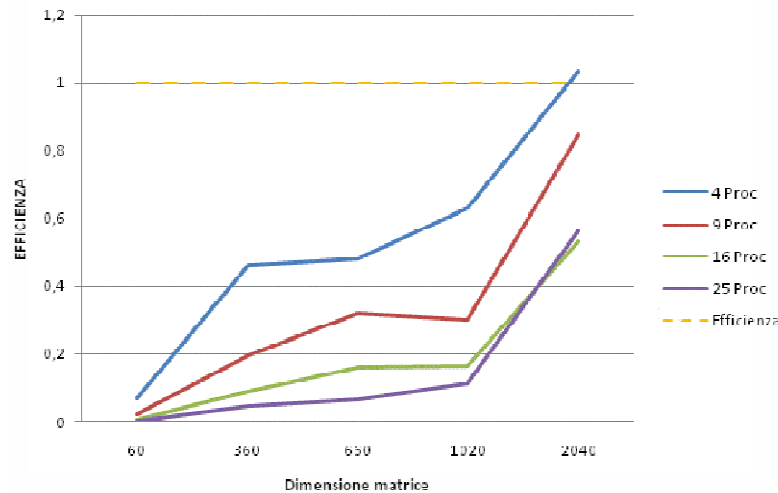
Metodo 1 Peso 0



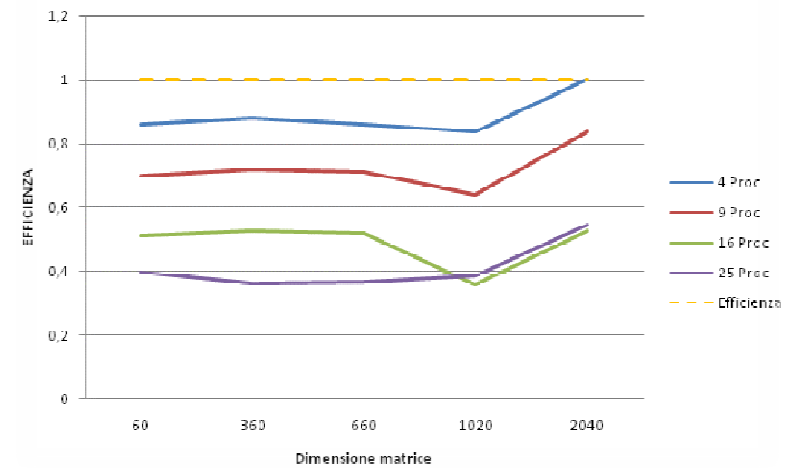
Metodo 1 Peso 6



Metodo 1 Peso 0



Metodo 1 Peso 6



Diapositiva 18

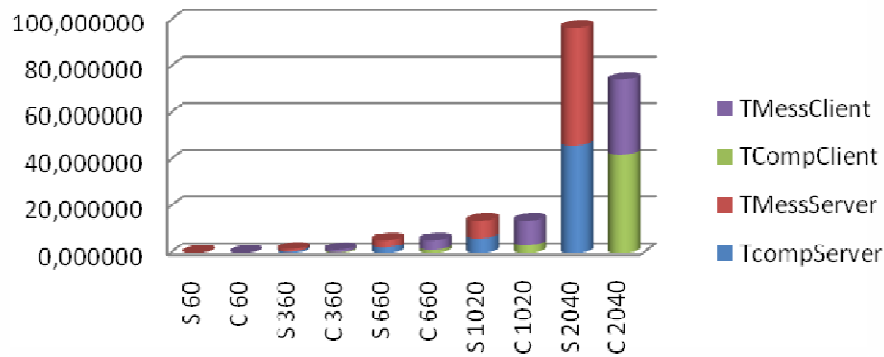
MB10

55, 56

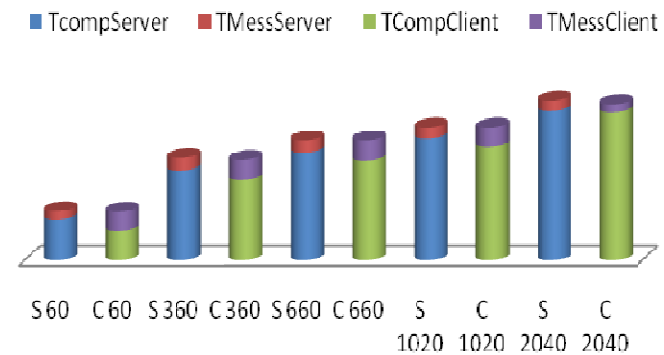
Mattia Boldrin; 13/10/2007

Metodo 1: ripartizione tempi

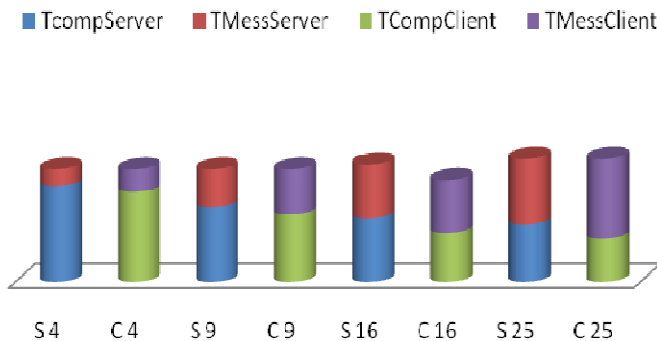
**Ripartizione dei tempi metodo1
proc=25 peso=6**



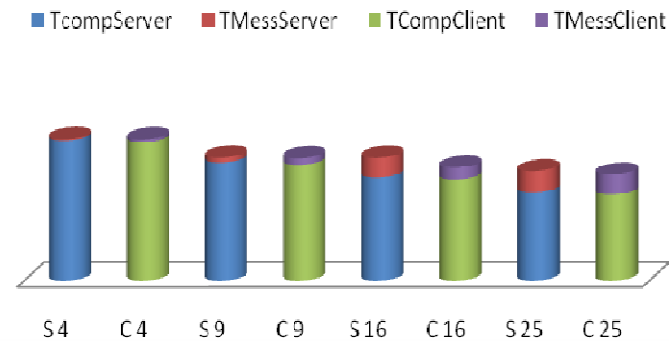
**Ripartizione dei tempi metodo1
proc=25 peso=6**



**Ripartizione dei tempi metodo1
dim=360 peso=2**



**Ripartizione dei tempi metodo1
dim2040 peso2**



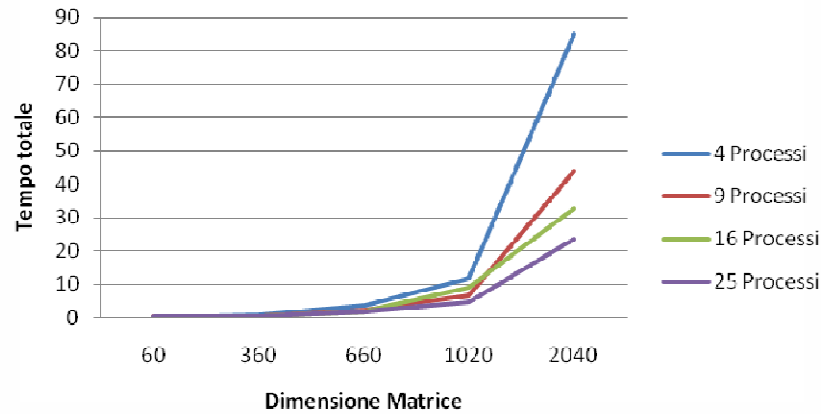
Diapositiva 19

MB34

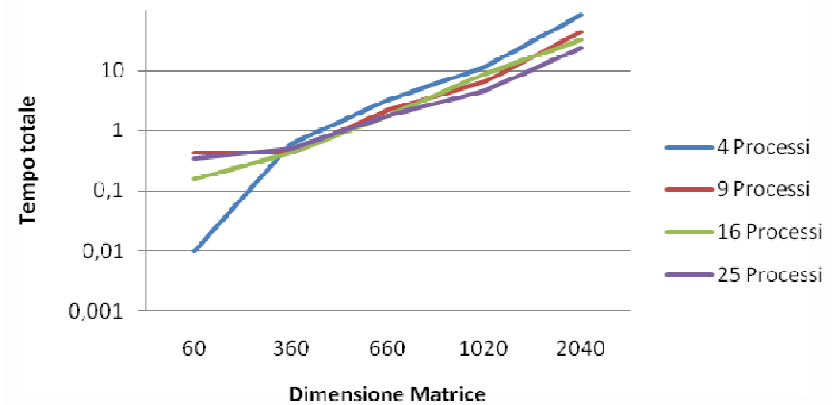
134, 137, 138

Mattia Boldrin; 15/10/2007

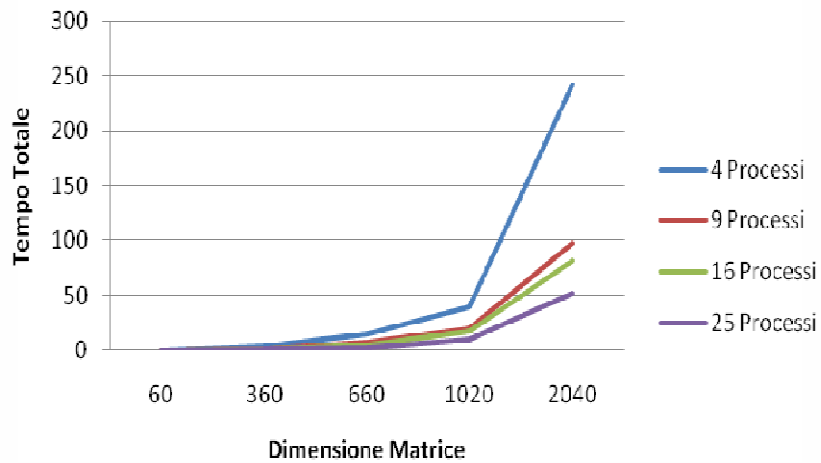
Metodo 2, Peso Computazionale 0



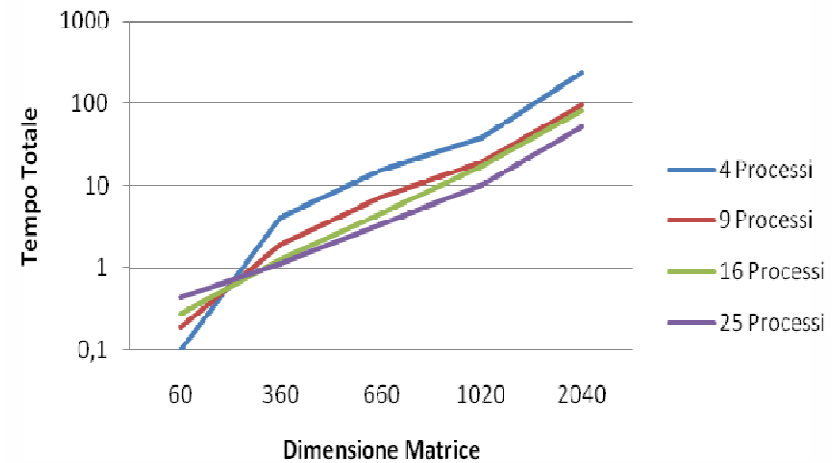
Metodo 2, Peso Computazionale 0



Metodo 2, Peso Computazionale 6



Metodo 2, Peso Computazionale 6



Diapositiva 20

MB5

11, 12, 29, 30

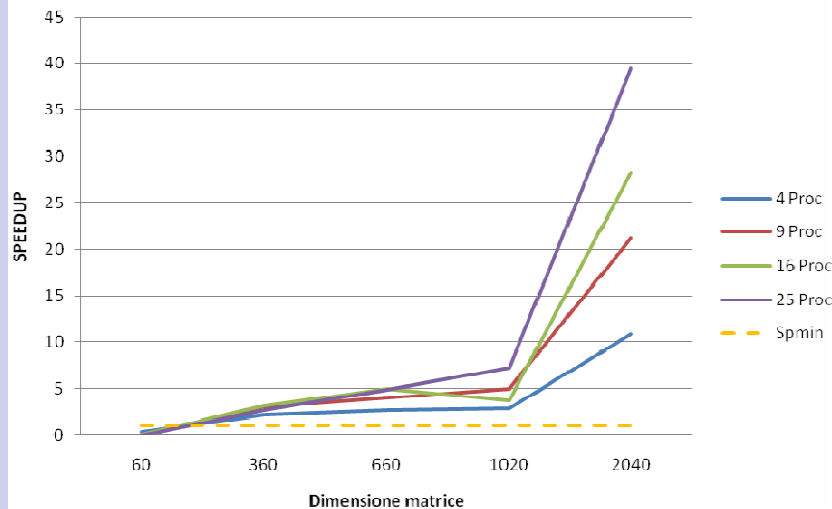
Mattia Boldrin; 13/10/2007

SpeedUp/Efficienza Metodo 2

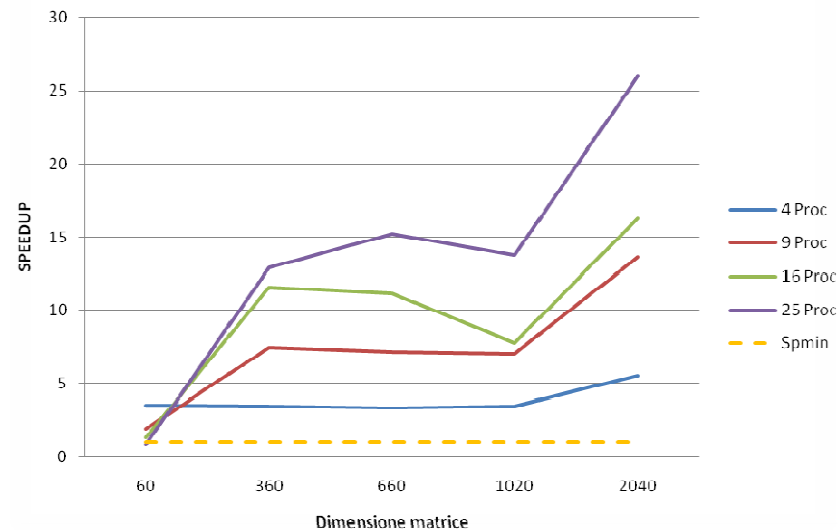
SpeedUp

Efficienza

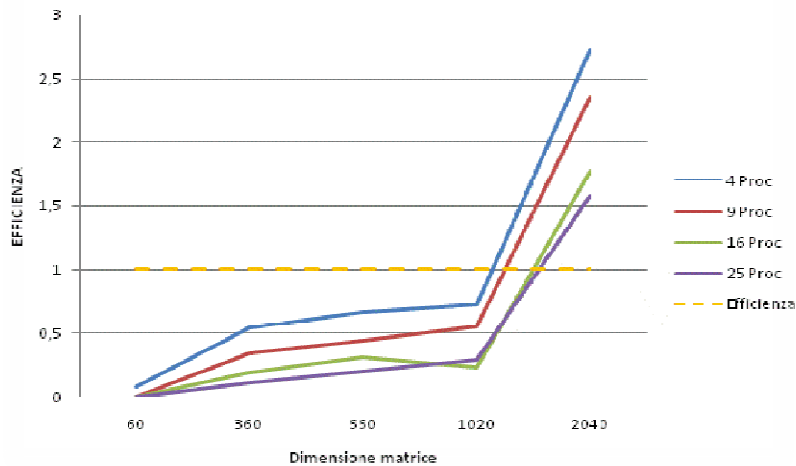
Metodo 2 Peso 0



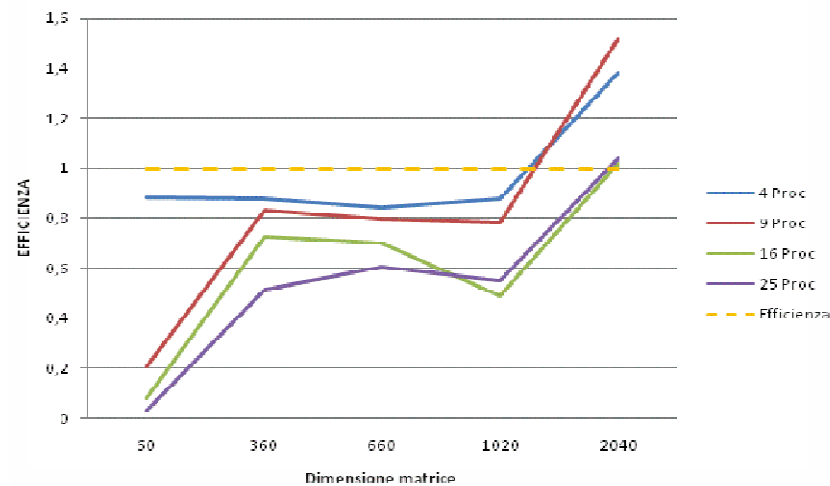
Metodo 2 Peso 6



Metodo 2 Peso 0



Metodo 2 Peso 6



Diapositiva 21

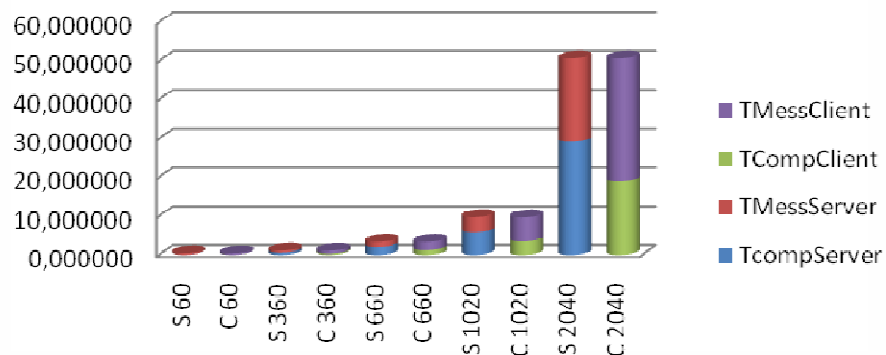
MB11

57, 58, 213, 214

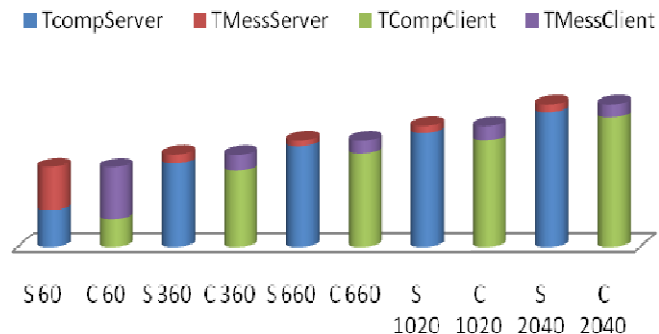
Mattia Boldrin; 15/10/2007

Metodo 2: ripartizione dei tempi

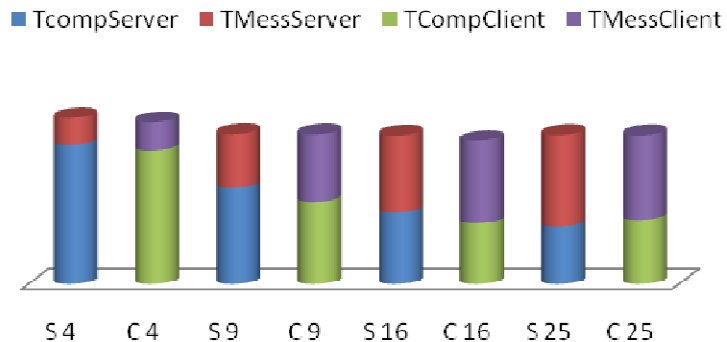
**Ripartizione dei tempi metodo2
proc=25 peso=6**



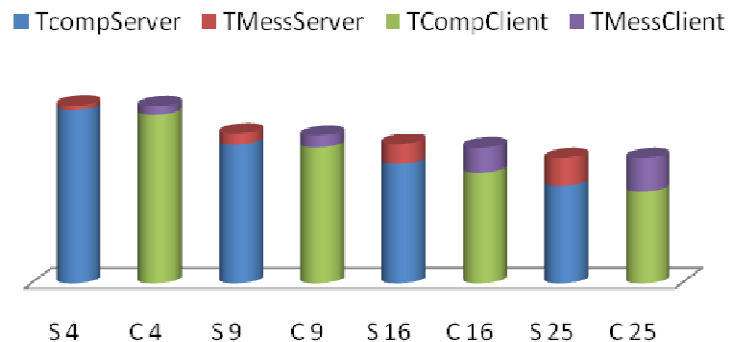
**Ripartizione dei tempi metodo2
proc=25 peso=6**



**Ripartizione dei tempi metodo2
dim=360 peso=2**



**Ripartizione dei tempi metodo2
dim2040 peso2**



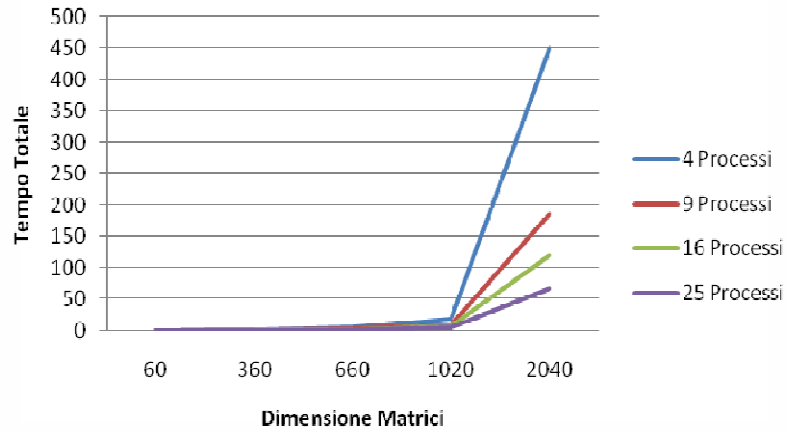
Diapositiva 22

MB35

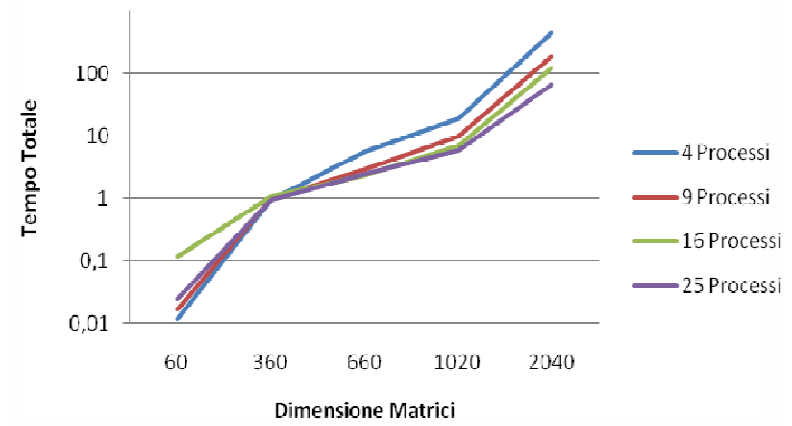
145, 147, 148

Mattia Boldrin; 15/10/2007

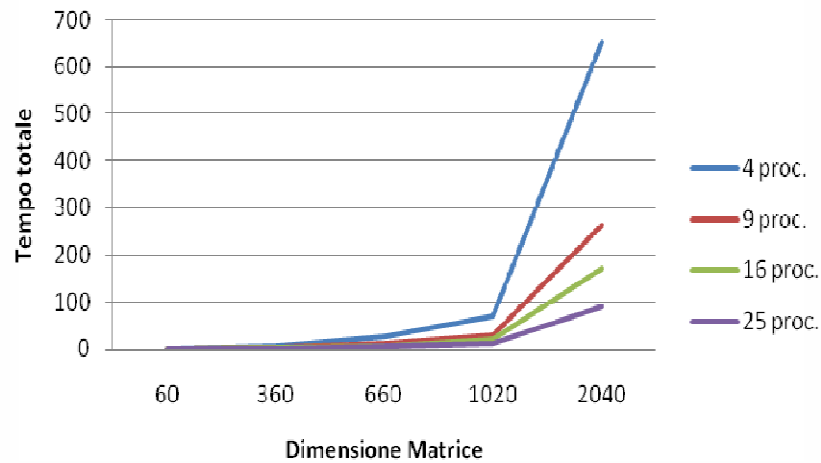
Metodo 3, Peso Computazionale 0



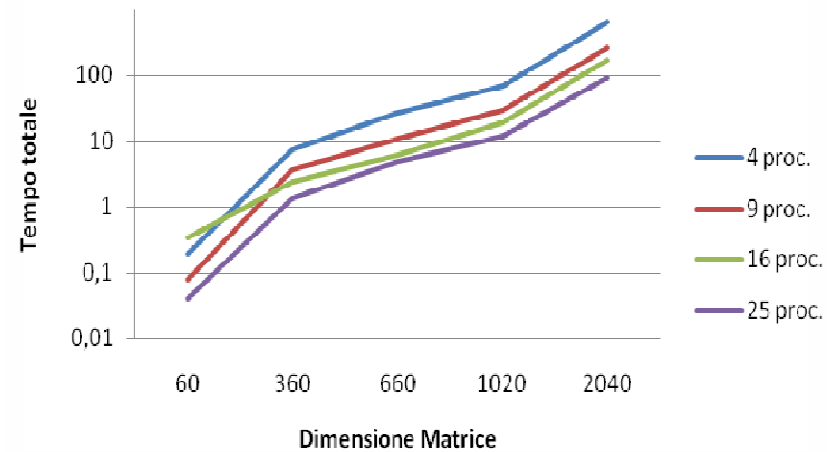
Metodo 3, Peso Computazionale 0



Metodo 3, Peso Computazionale 6



Metodo 3, Peso Computazionale 6



Diapositiva 23

MB6

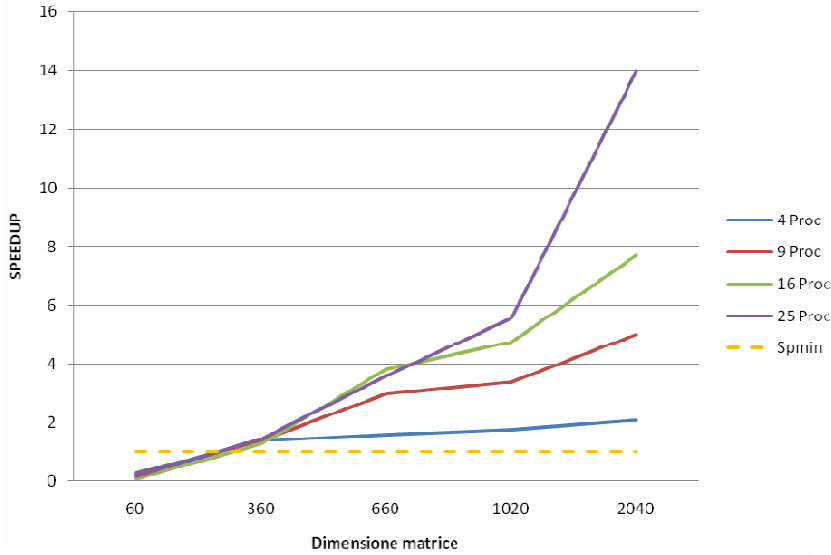
13, 14, 31, 32

Mattia Boldrin; 13/10/2007

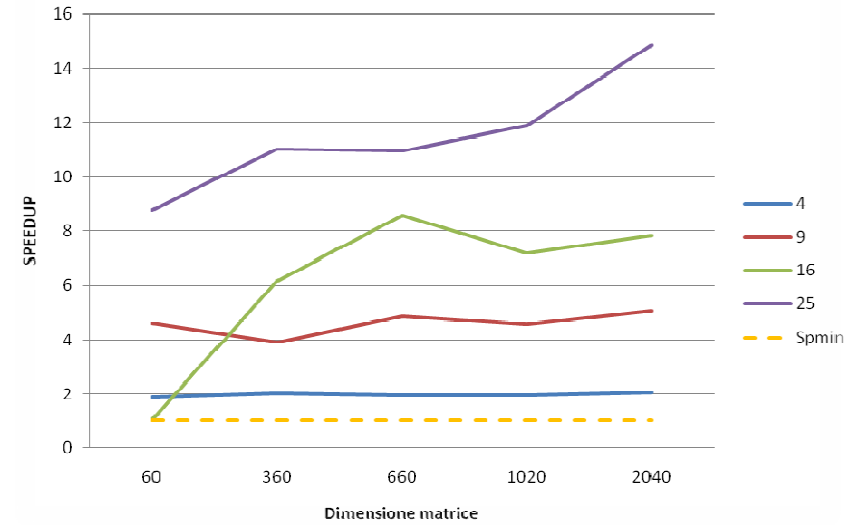
SpeedUp/Efficienza Metodo 3

SpeedUp

Metodo 3 Peso 0

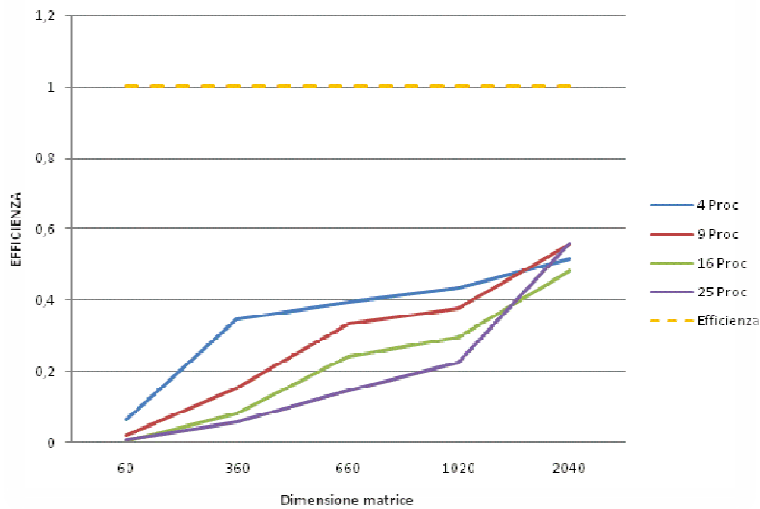


Metodo 3 Peso 6

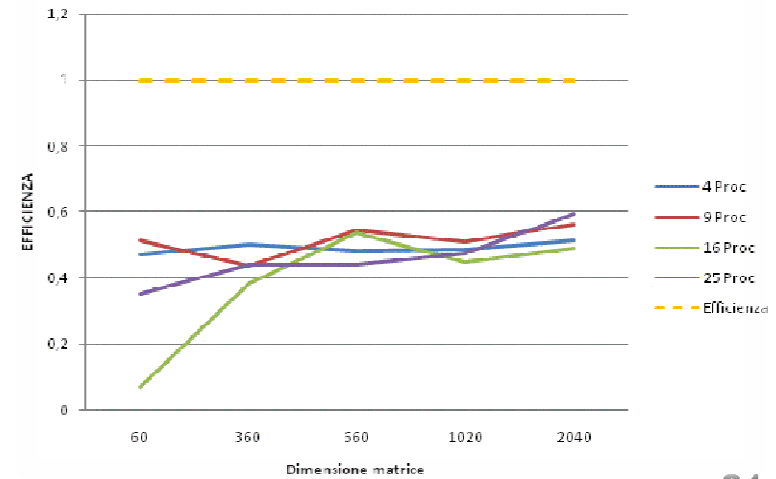


Efficienza

Metodo 3 Peso 0



Metodo 3 Peso 6



Diapositiva 24

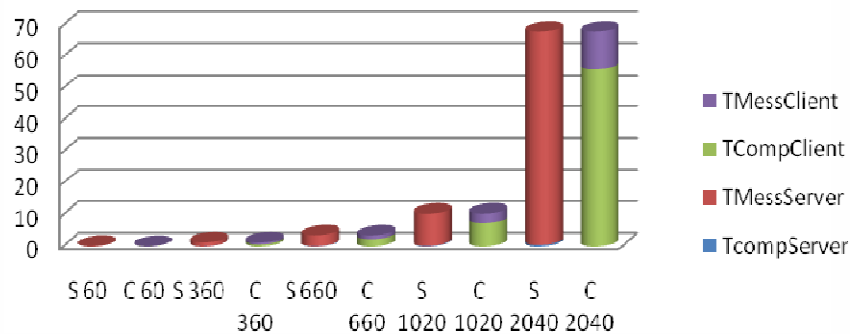
MB12

59, 60, 215, 216

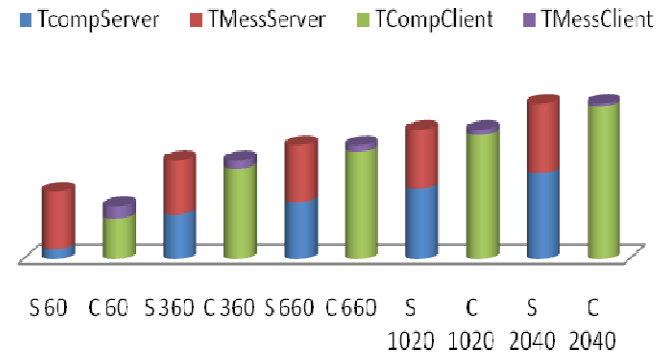
Mattia Boldrin; 15/10/2007

Metodo 3: ripartizione dei tempi

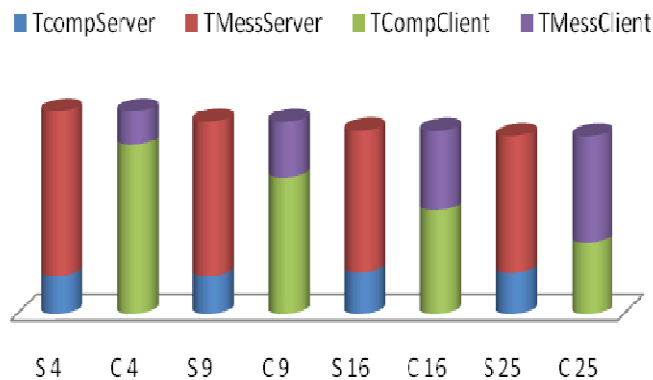
Ripartizione dei tempi metodo 3min
proc=25 peso=6



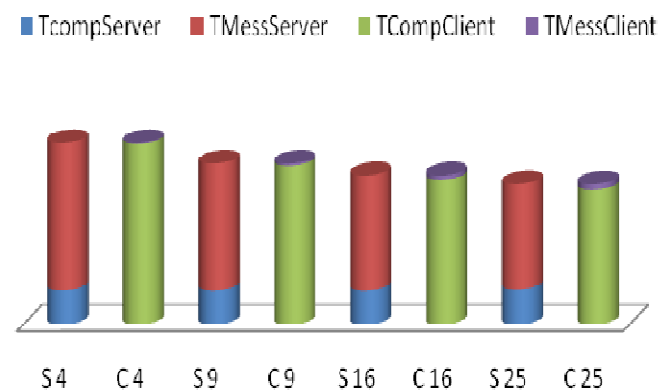
Ripartizione dei tempi metodo
3min proc=25 peso=6



Ripartizione dei tempi metodo
3min dim=360 peso=2



Ripartizione dei tempi metodo
3min dim2040 peso2



Diapositiva 25

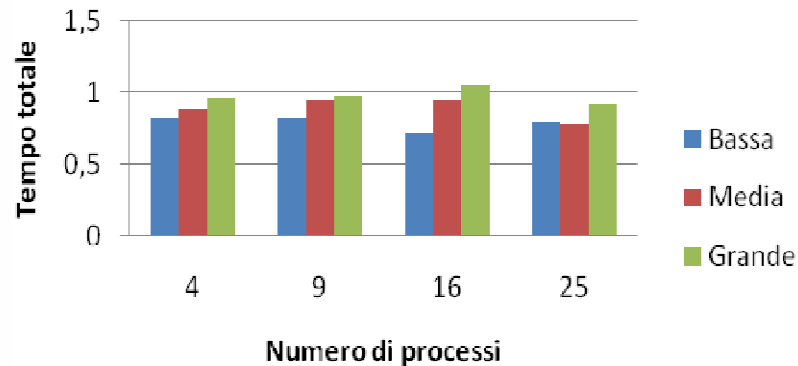
MB36

155, 157, 158

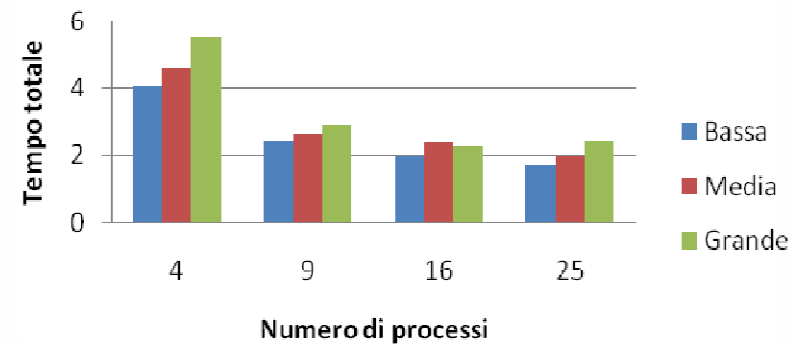
Mattia Boldrin; 15/10/2007

Metodo 3: Granularità

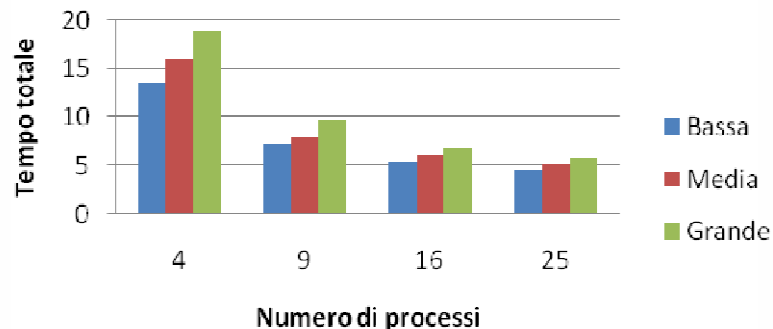
Dim Matrice 360, Peso 0



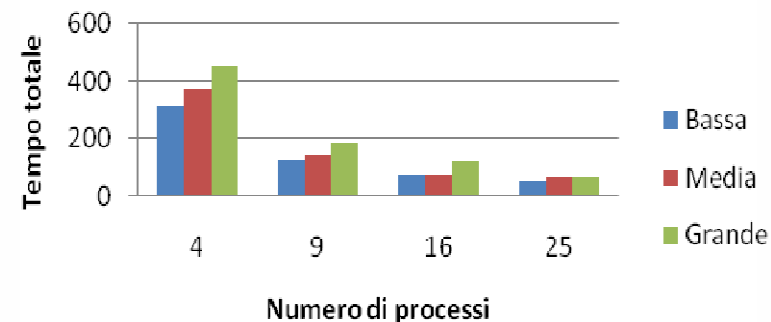
Dim Matrice 660, Peso 0



Dim Matrice 1020, Peso 0



Dim Matrice 2040, Peso 0



Nella relazione è possibile osservare gli stessi test con peso computazionale 6 (G48 – G51).

26

Diapositiva 26

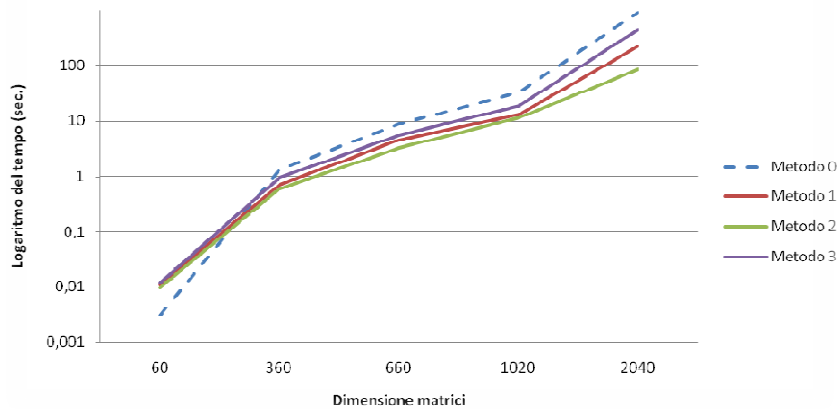
MB7

44, 45, 46, 47

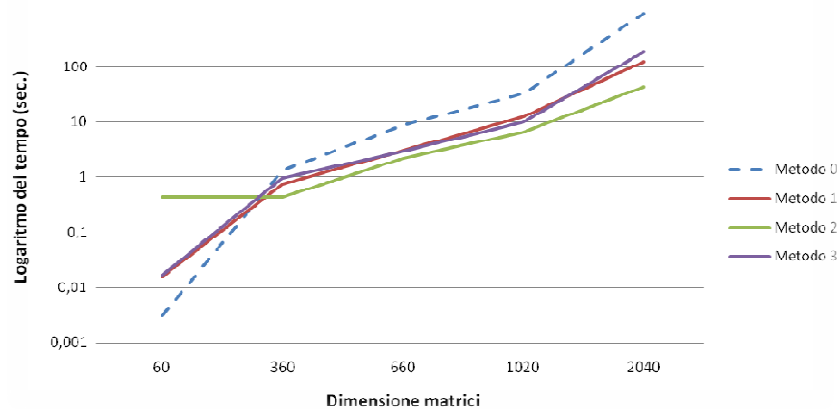
Mattia Boldrin; 13/10/2007

Confronto tempi al variare della dimensione della matrice

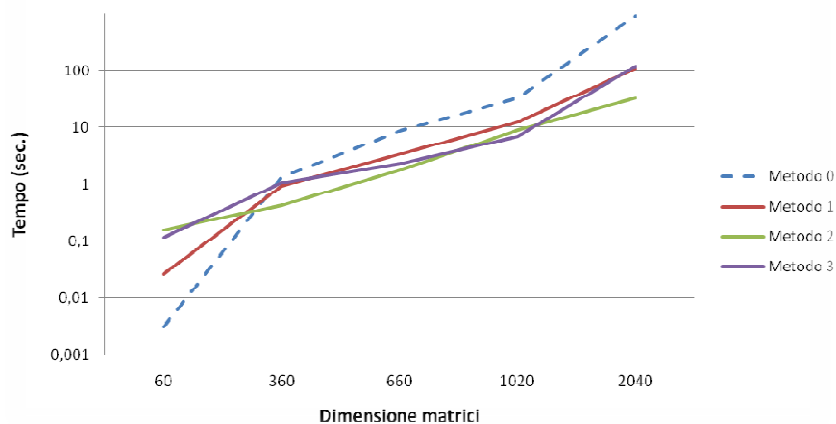
Confronto dei tempi tra il Metodo Sequenziale e i Metodi paralleli con 4 Processi



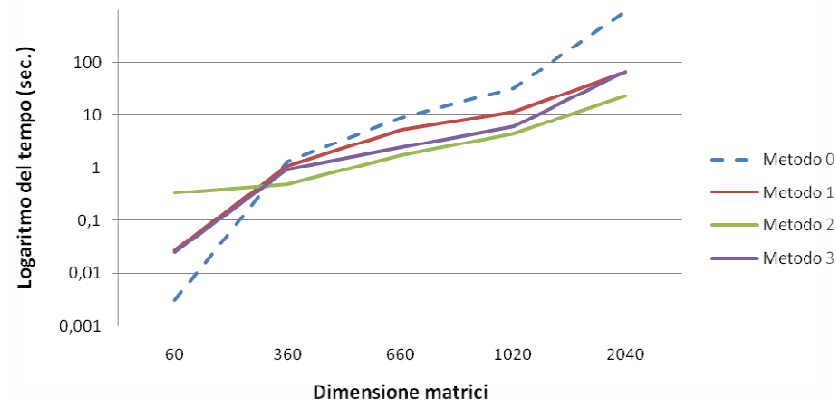
Confronto dei tempi tra il Metodo Sequenziale e i Metodi paralleli con 9 Processi



Confronto dei tempi tra il Metodo Sequenziale e i Metodi paralleli con 16 Processi



Confronto dei tempi tra il Metodo Sequenziale e i Metodi paralleli con 25 Processi



Diapositiva 27

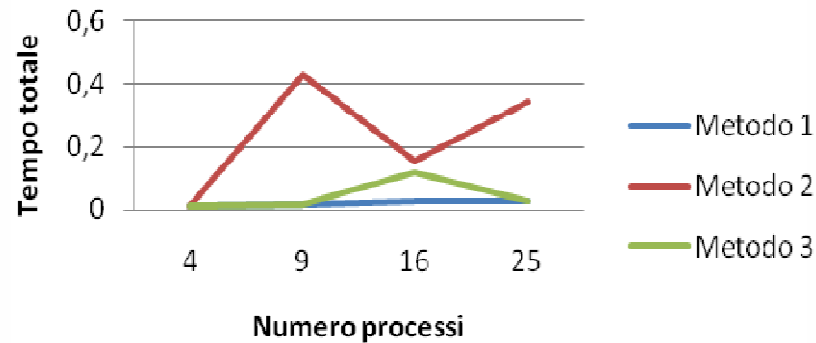
MB14

80, 82, 84, 86

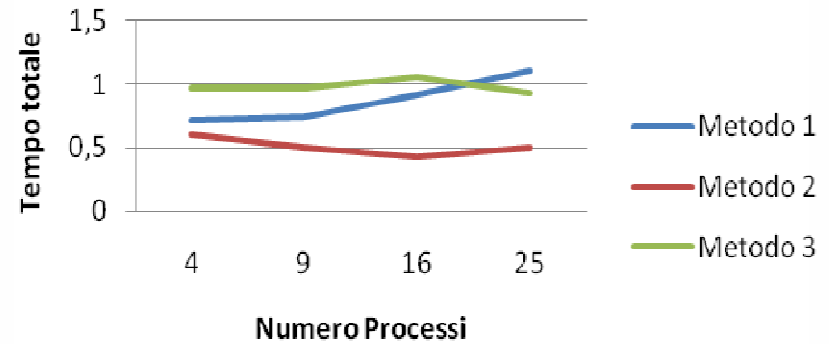
Mattia Boldrin; 13/10/2007

Confronto tempi al variare dei processori

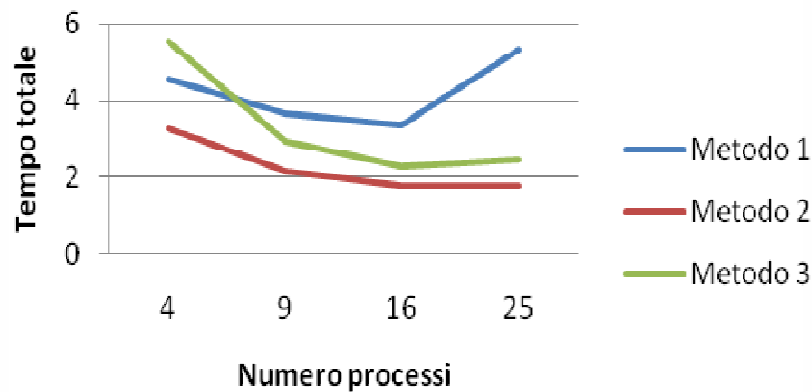
Matrice 60x60, Peso 0



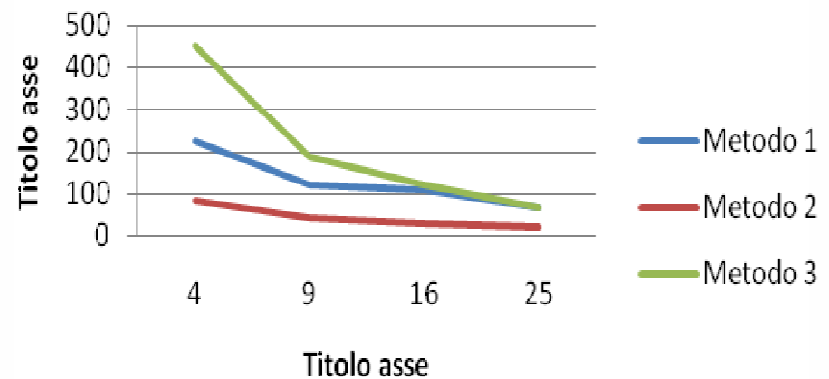
Dim Matrice 360x360, Peso 0



Dim Matrice 660, Peso 0



Dim Matrice 2040, Peso 0



Diapositiva 28

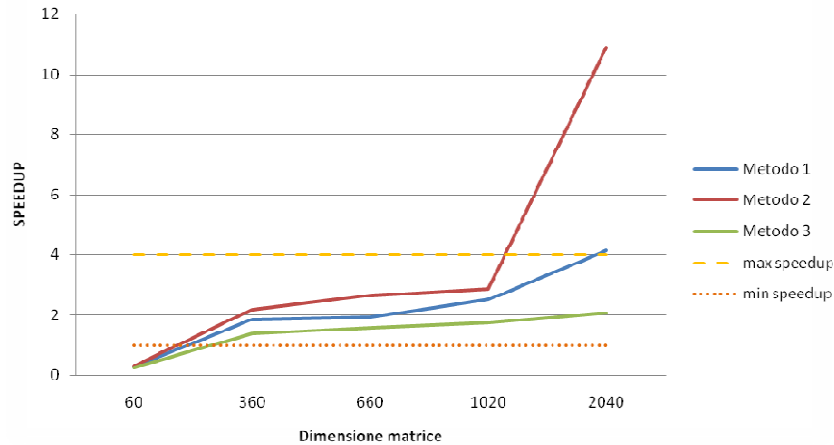
MB13

33, 34, 35, 36

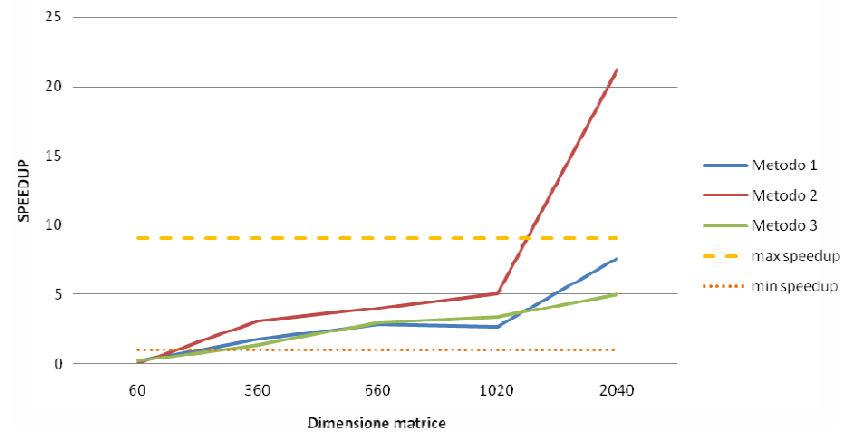
Mattia Boldrin; 13/10/2007

Confronto degli SeedUp

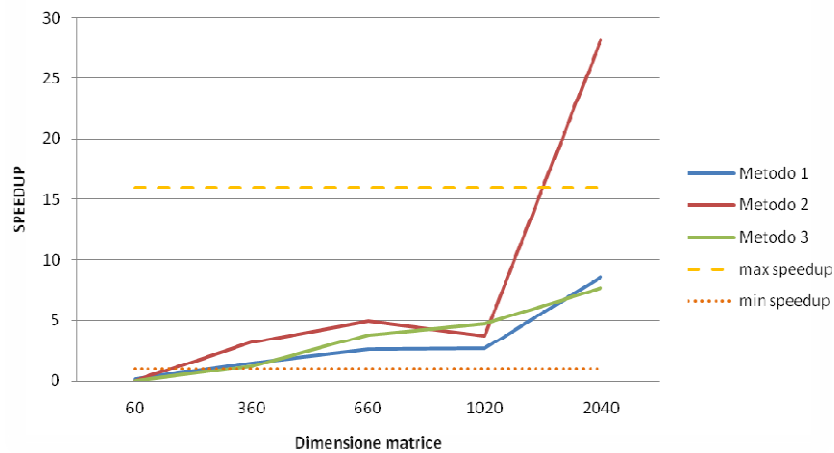
Numero Processi 4 Peso 0



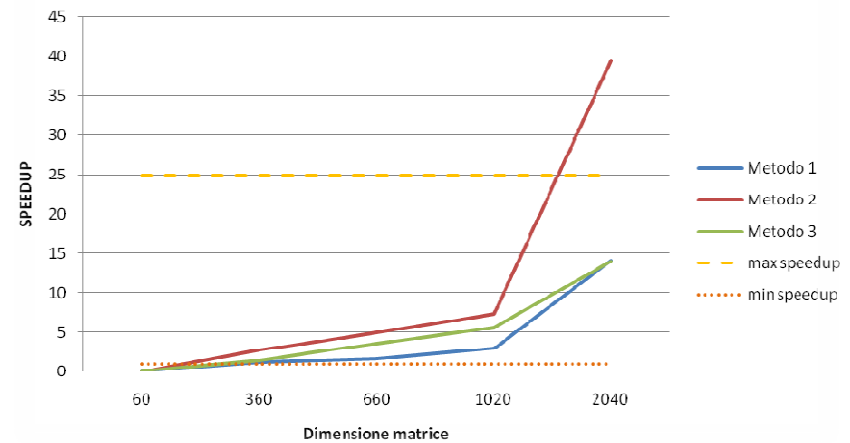
Numero Processi 9 Peso 0



Numero Processi 16 Peso 0



Numero Processi 25 Peso 0



Diapositiva 29

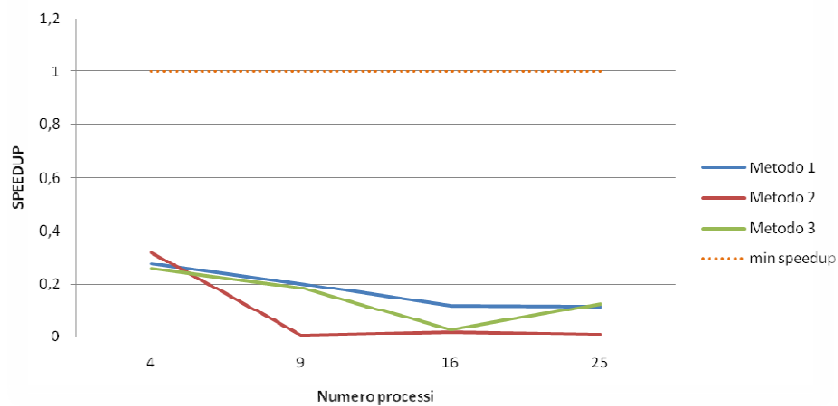
MB15

61, 63, 65, 67

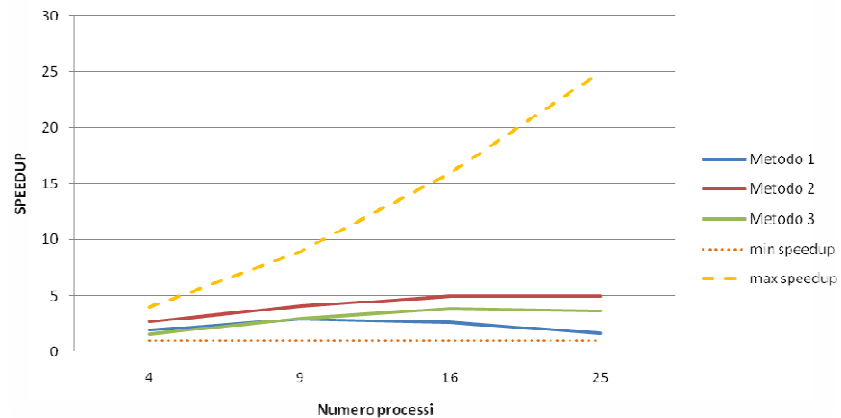
Mattia Boldrin; 13/10/2007

Confronto degli SpeedUp

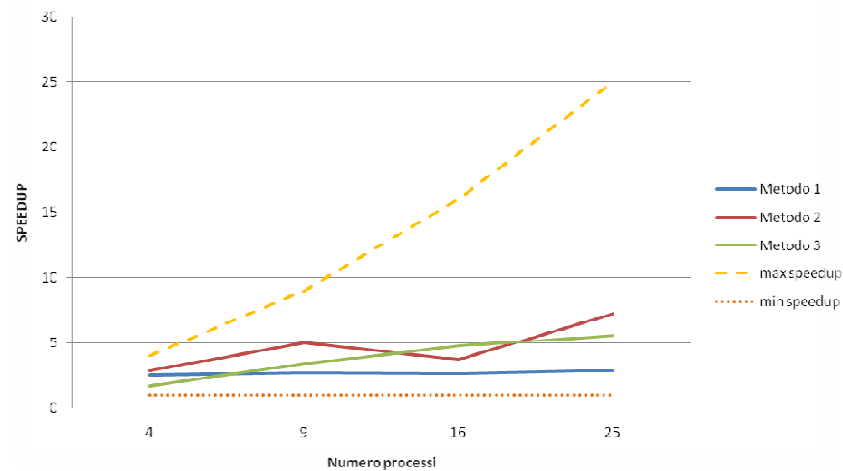
Dimensione matrice 60 Peso 0



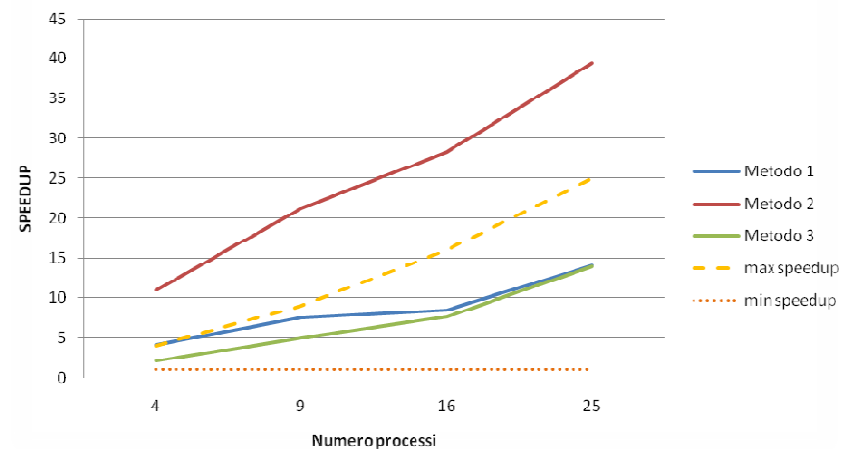
Dimensione matrice 660 Peso 0



Dimensione matrice 1020 Peso 0



Dimensione matrice 2040 Peso 0



Diapositiva 30

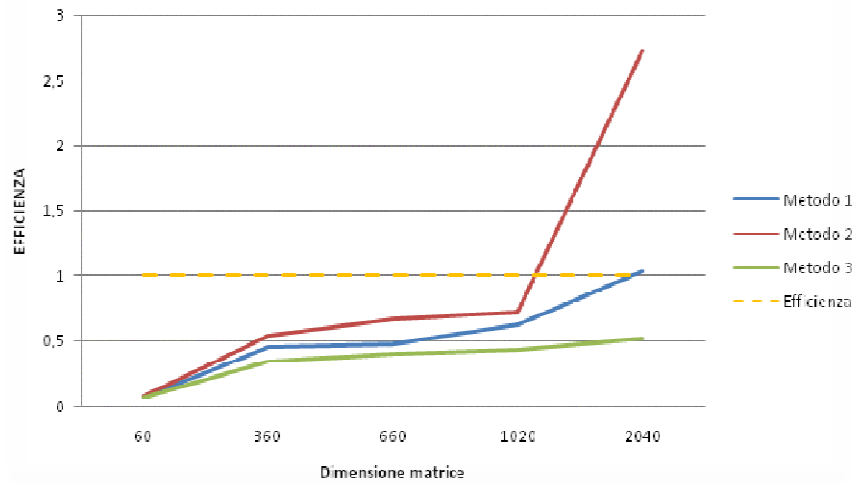
MB16

69, 73, 75, 77

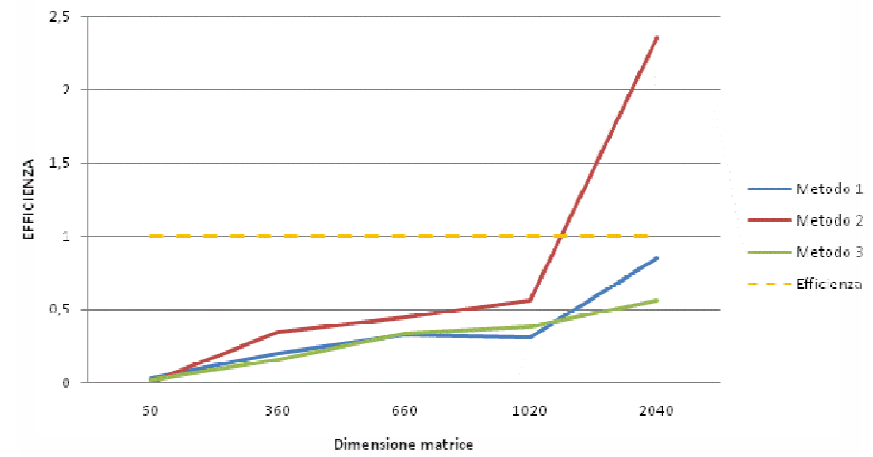
Mattia Boldrin; 13/10/2007

Confronto delle efficienze

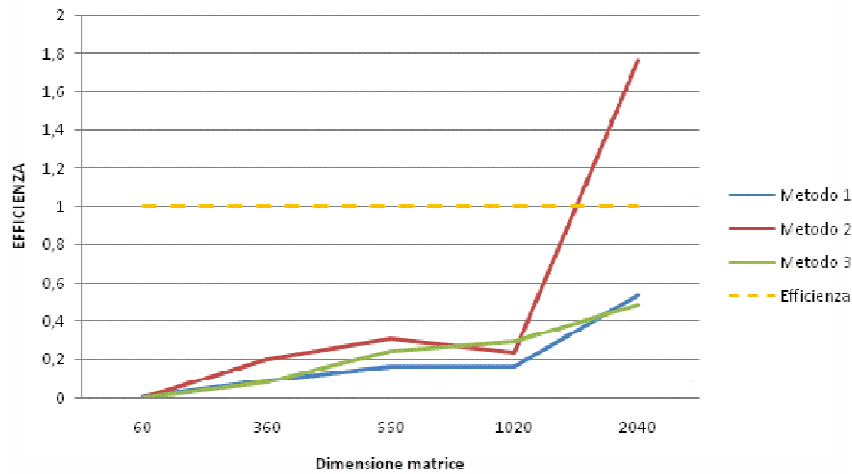
Numero Processi 4 Peso 0



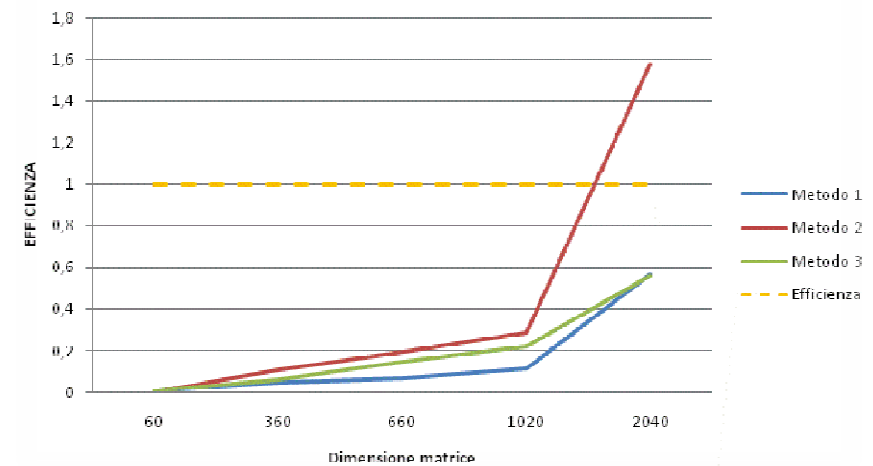
Numero Processi 9 Peso 0



Numero Processi 16 Peso 0



Numero Processi 25 Peso 0



Diapositiva 31

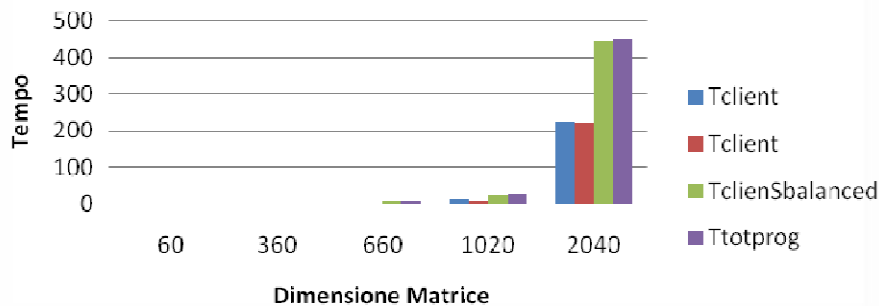
MB30

217, 218, 219, 220

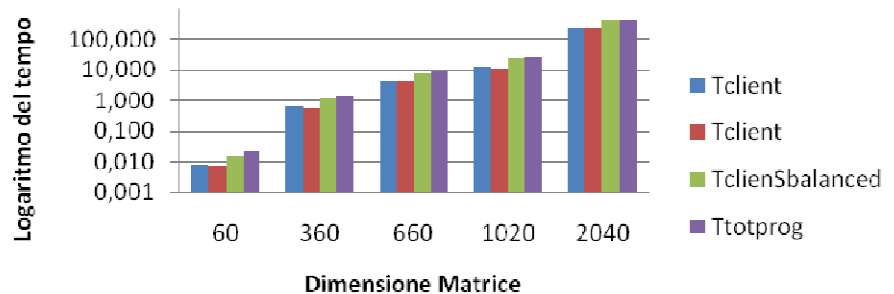
Mattia Boldrin; 13/10/2007

Metodo 1: sbilanciamento del carico [1/2]

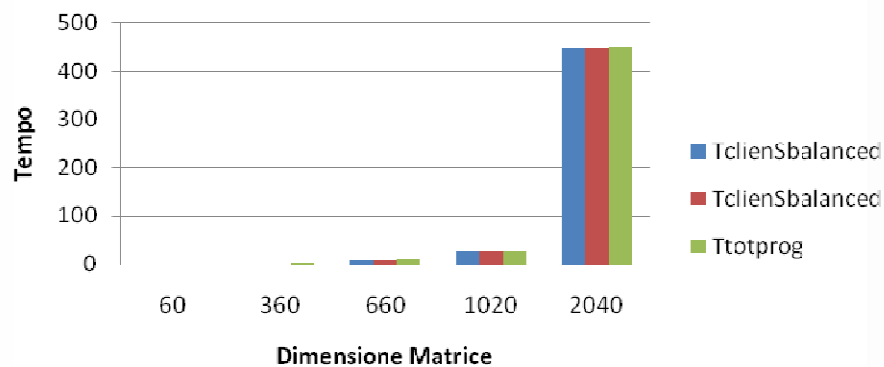
Tempi metodo 1: 3 processori con 2 processi su stesso processore peso comp 0



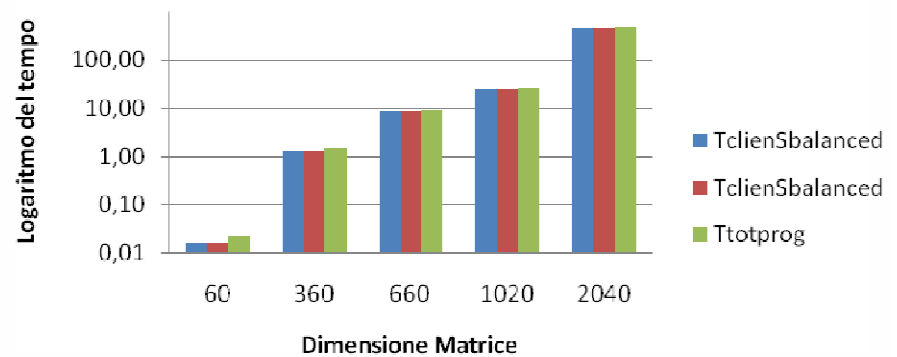
Tempi metodo 1: 3 processori con 2 processi su stesso processore peso comp 0



Tempi metodo 1: 4 processi allocati in 2 processori peso 0



Tempi metodo 1: 4 processi allocati in 2 processori peso 0



Diapositiva 32

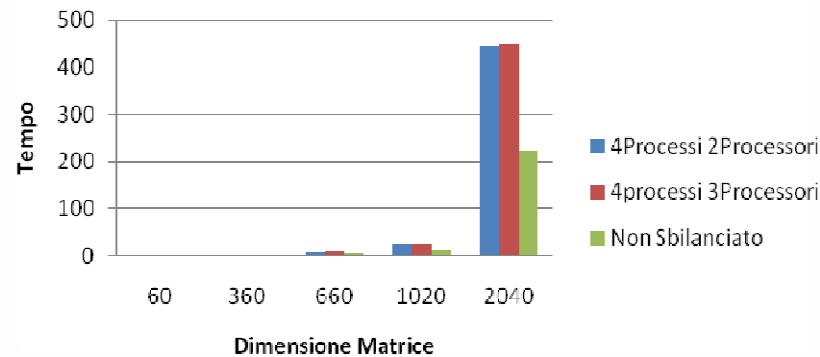
MB21

167, 168, 169, 170

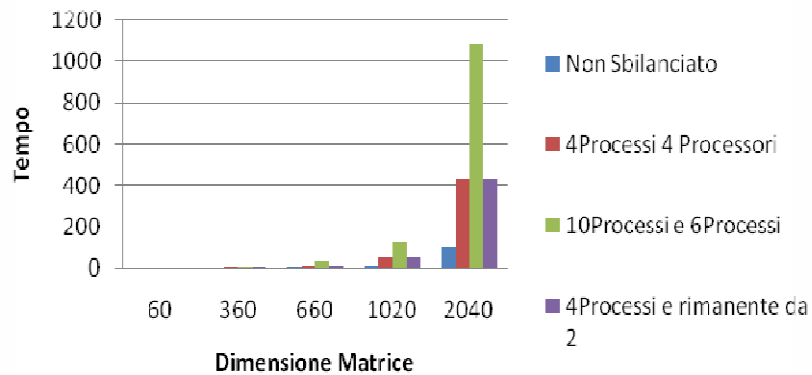
Mattia Boldrin; 13/10/2007

Metodo 1: sbilanciamento del carico [2/2]

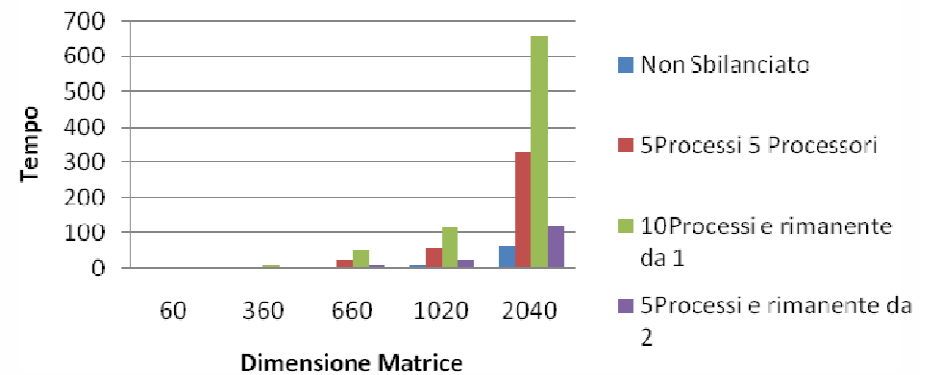
Confronto tempo metodo 1: con sbilanciamento



Metodo 1: confronto tempi peso 0 16 processi



Metodo 1: confronto tempi peso 0 25 processi



Diapositiva 33

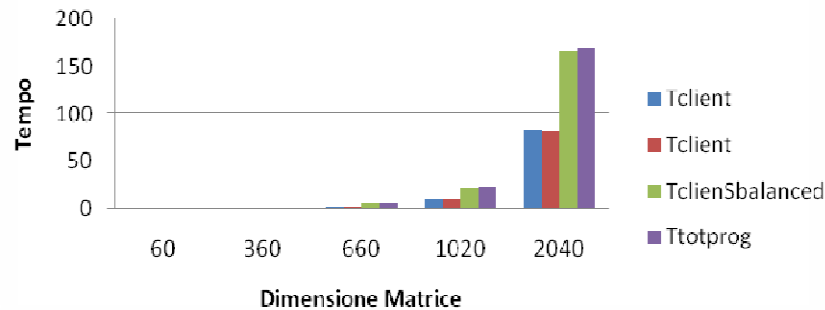
MB22

176, 178, 180

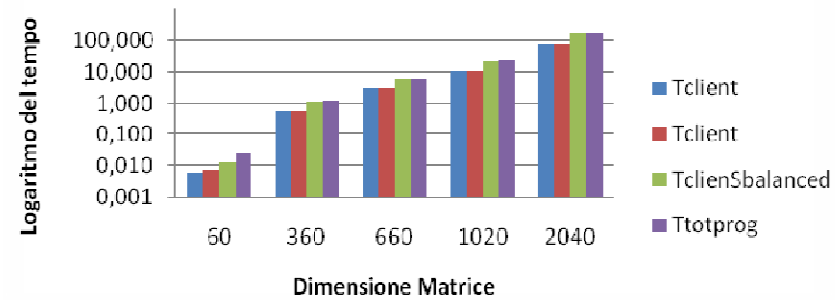
Mattia Boldrin; 13/10/2007

Metodo 2: sbilanciamento del carico [1/2]

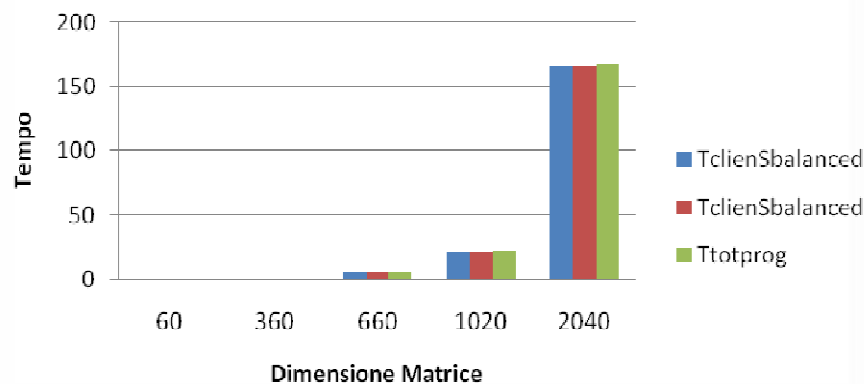
Tempi metodo 2: 3 processori con 2 processi su stesso processore peso comp 0



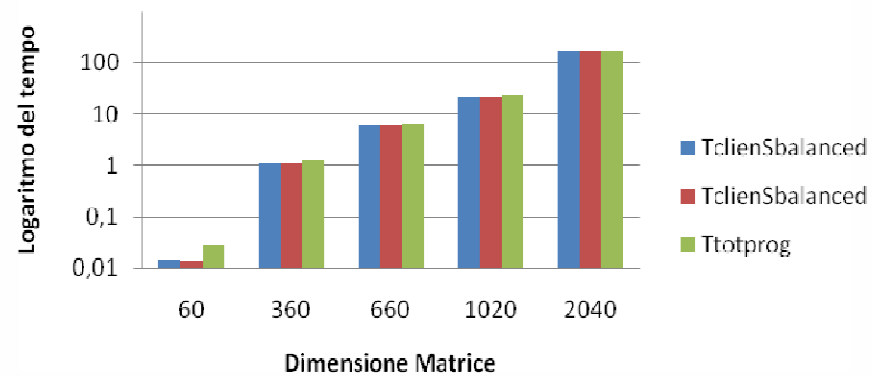
Tempi metodo 2: 3 processori con 2 processi su stesso processore peso comp 0



Tempi metodo 2: 4 processi allocati in 2 processori peso 0



Tempi metodo 2: 4 processi allocati in 2 processori peso 0



Diapositiva 34

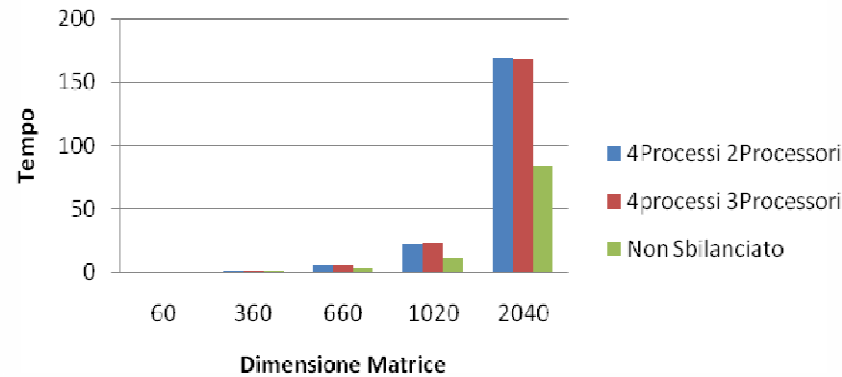
MB24

184, 185, 186, 187

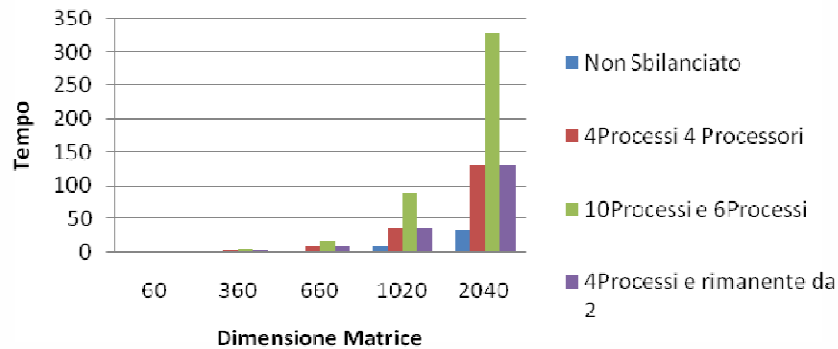
Mattia Boldrin; 13/10/2007

Metodo 2: sbilanciamento del carico [2/2]

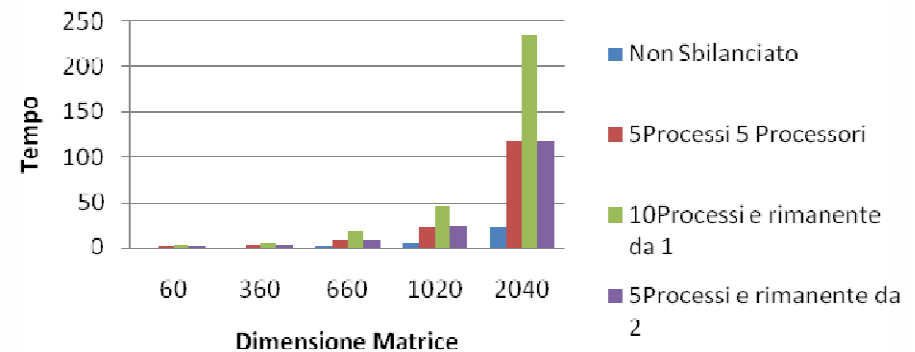
Confronto tempi metodo 2: con sbilanciamenti e senza peso 0



Metodo 2: confronto tempi peso 0 16 processi



Metodo 2: confronto tempi peso 0 25 processi



Diapositiva 35

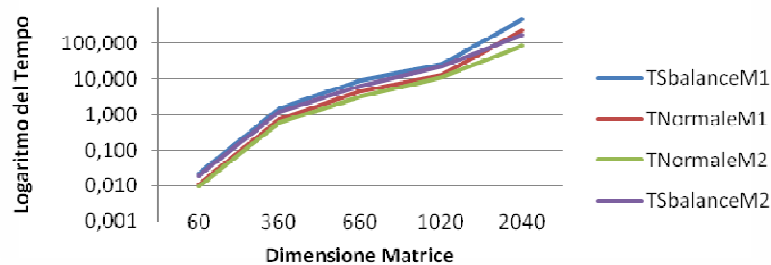
MB25

197, 195, 193

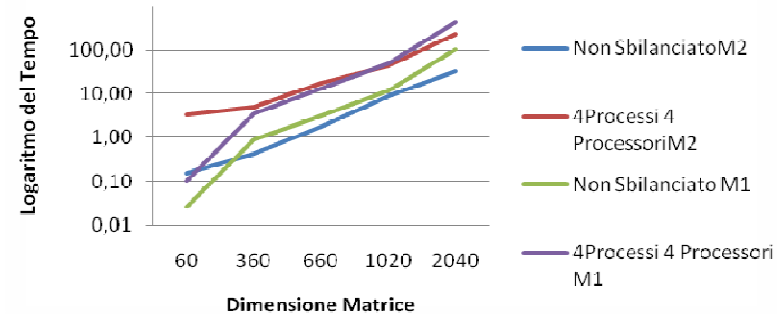
Mattia Boldrin; 13/10/2007

Metodo 1 & 2: sbilanciamento del carico

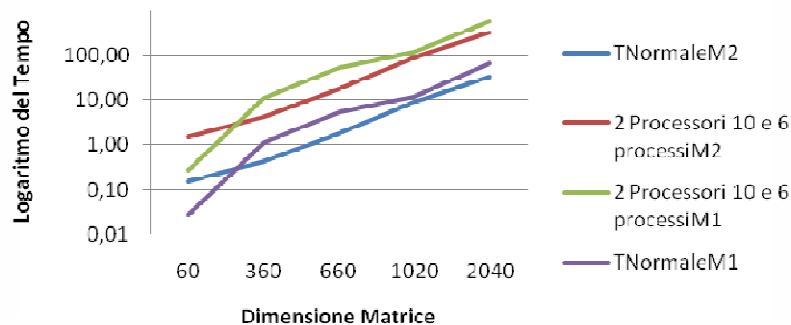
Sbilanciamento 4 processi con 2 processi su stesso processore peso comp 0



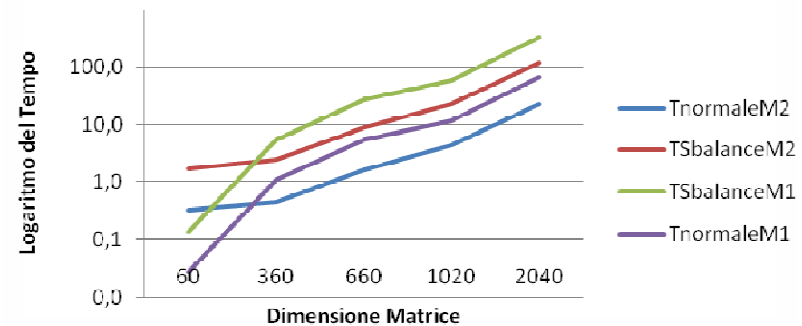
Confronto sbilanciamento con 16 processori peso 0



Confronto sbilanciamento con 16 processori peso 0



Confronto sbilanciamento con 25 processi con 5 processori peso 0



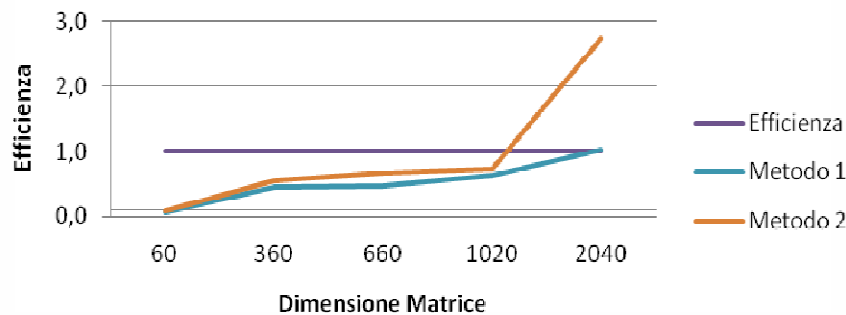
Diapositiva 36

MB26

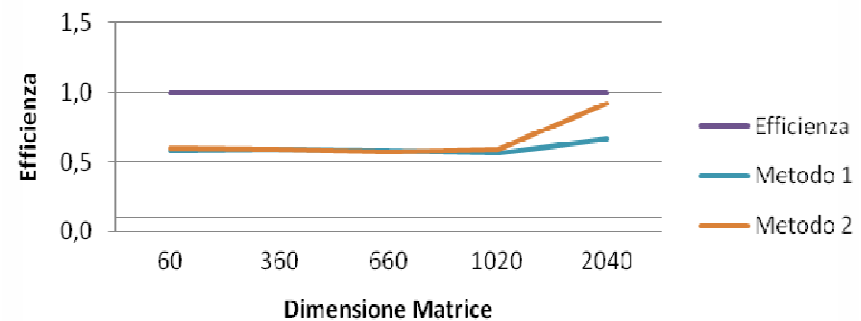
201, 203, 204, 205
Mattia Boldrin; 13/10/2007

Confronto delle efficienze: carico sbilanciato [1/2]

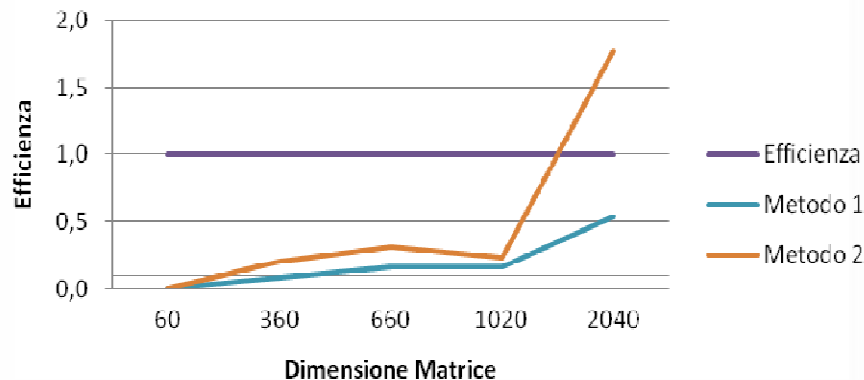
Sbilanciamento 4 processi con 2 processi su stesso processore peso comp 6 con 2 processori



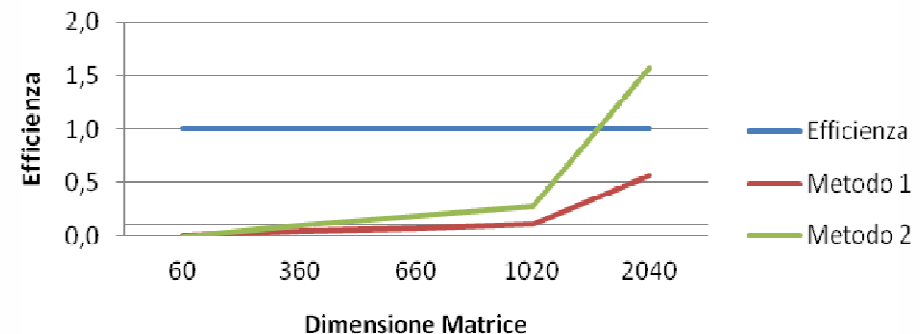
Sbilanciamento 4 processi con 2 processi su stesso processore peso comp 6 con 3 processori



Sbilanciamento 16 processi con 4 processi su 4 processori peso 0



Sbilanciamento 16 processi con 2 processori con rispettivamente 10 e 6 processi peso 6



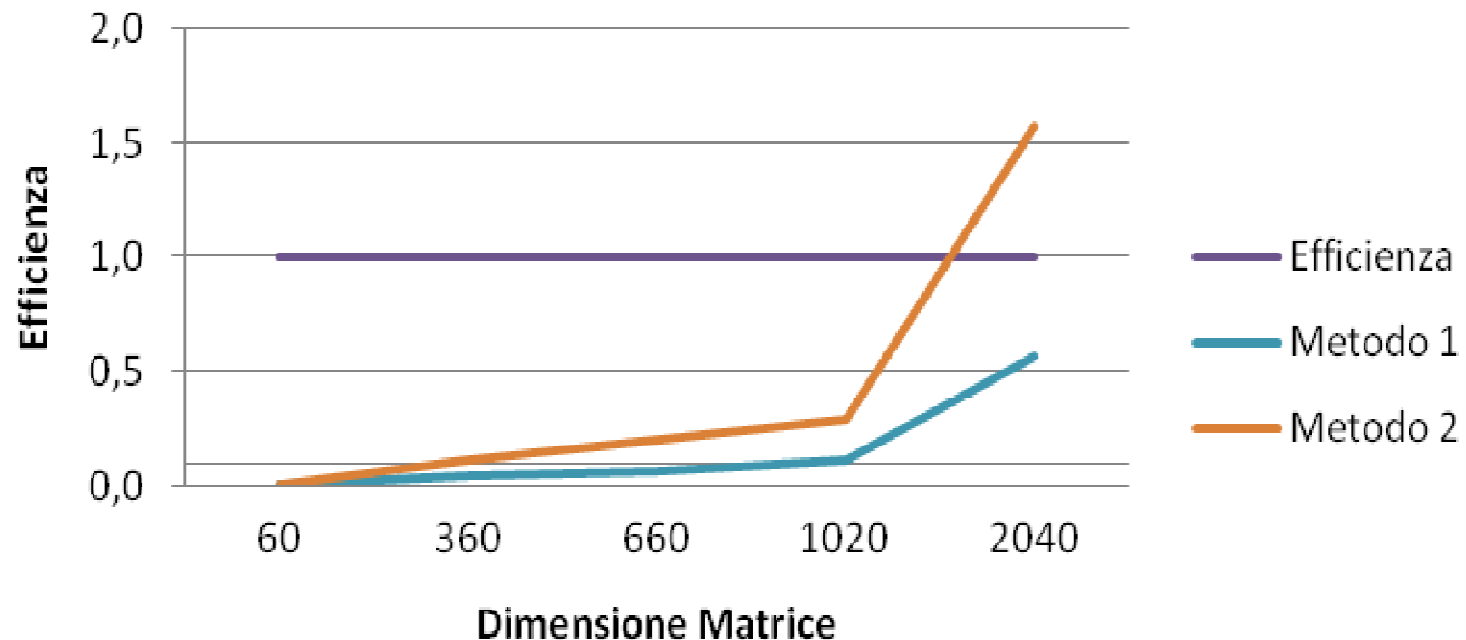
Diapositiva 37

MB31

217, 218, 219, 220

Mattia Boldrin; 13/10/2007

Sbilanciamento 25 processi con 5 processori allocati con 5 processi



Diapositiva 38

MB32

217, 218, 219, 220

Mattia Boldrin; 13/10/2007

Tempi client

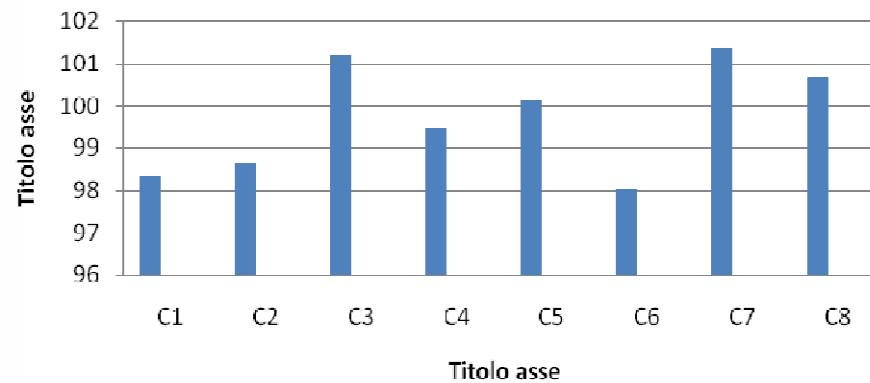
Metodo 1, Matrice 2040, Processi 9, Peso 0



Metodo 2, Matrice 2040, Processi 9, Peso 0



Metodo 3, Matrice 2040, Processi 9, Peso 0



Diapositiva 39

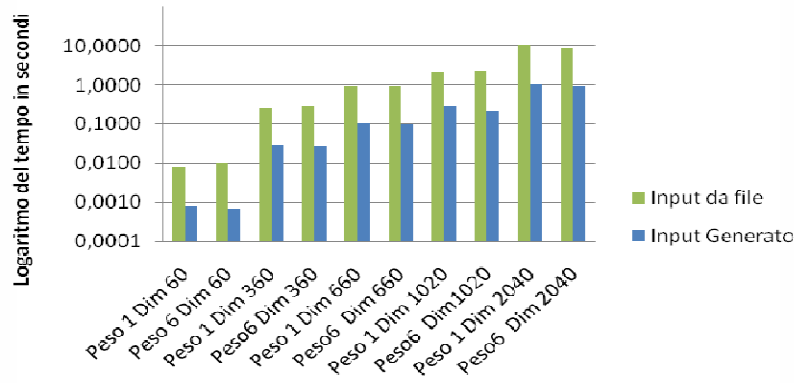
MB9

52, 53, 54

Mattia Boldrin; 13/10/2007

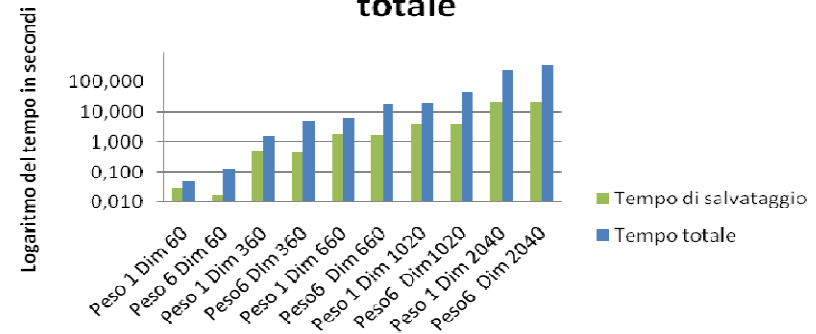
Tempi di I/O

Tempo di lettura e generazione della matrice



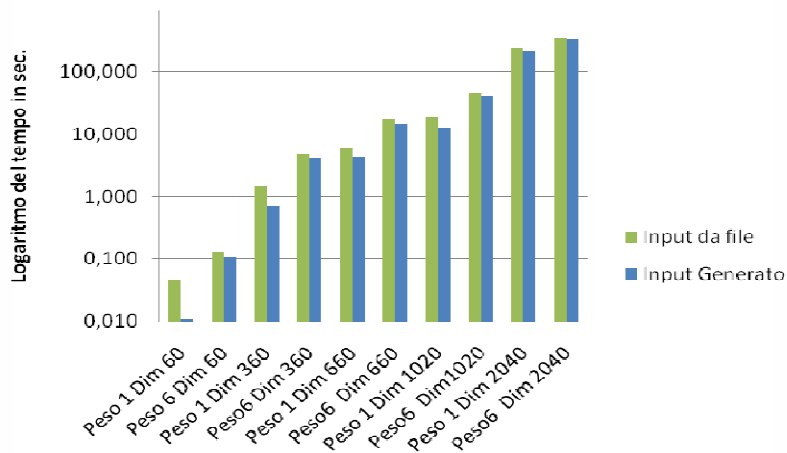
Casi peso computazionale e dimensione Matrice

Tempo di salvataggio rispetto al tempo totale



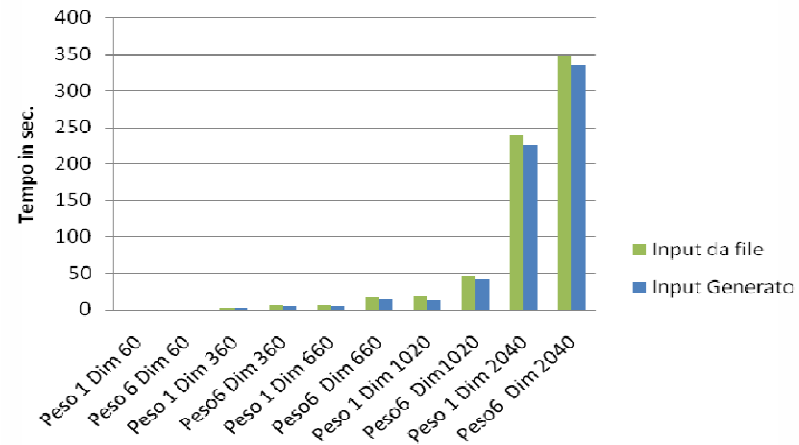
Casi peso computazionale e dimensione Matrice

Tempo di esecuzione totale



Casi: peso computazionale e dimensione Matrice

Tempo di esecuzione totale



Casi: peso computazionale e dimensione Matrice

Diapositiva 40

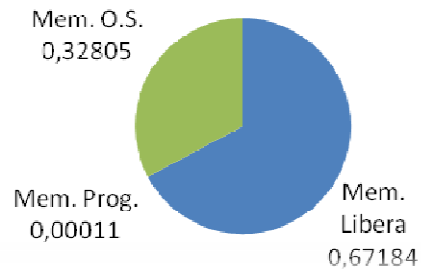
MB18

89, 92, 94, 95

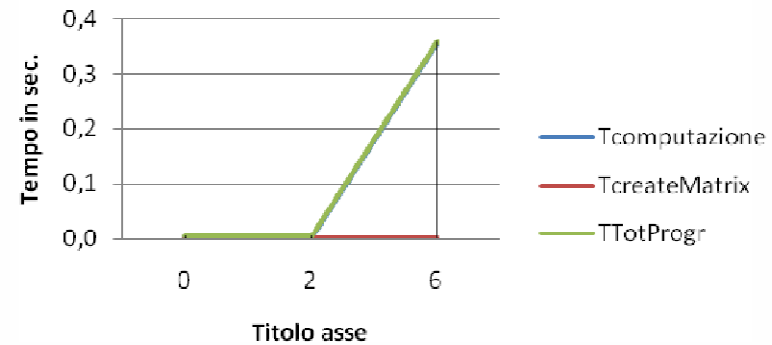
Mattia Boldrin; 13/10/2007

Partizionamento memoria, pesi e tempi

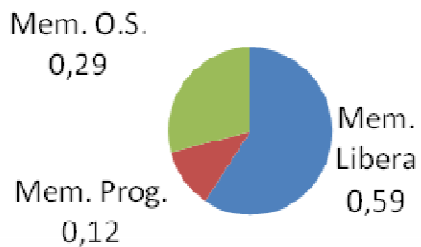
Partizionamento della memoria con dimensione 60



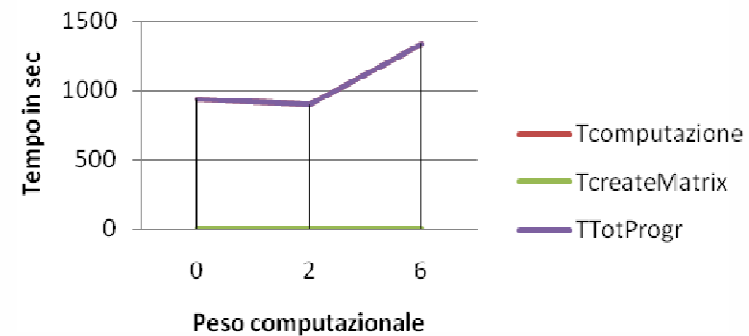
Tempo sequenziale dimensione matrice 60



Partizionamento della memoria con dimensione 2040



Tempo sequenziale con dimensione 2040



Diapositiva 41

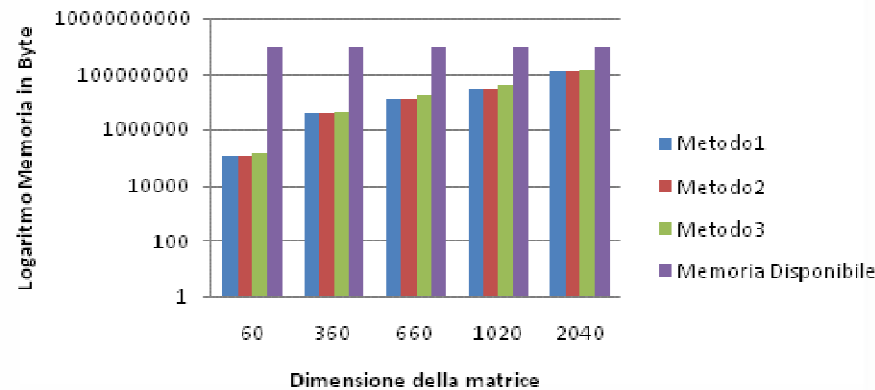
MB19

117, 115, 123, 124

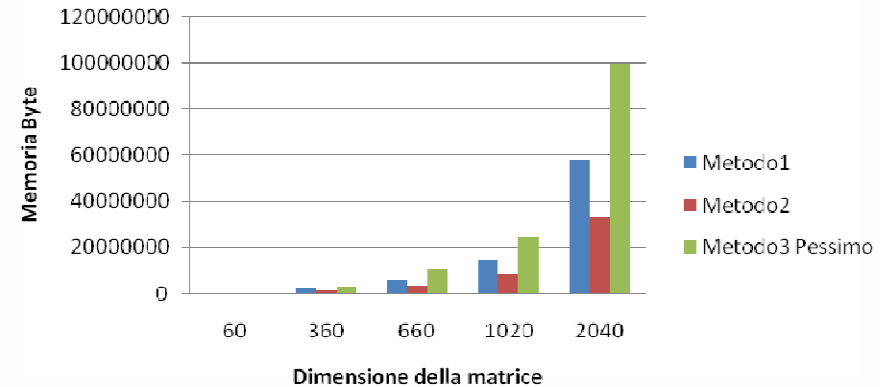
Mattia Boldrin; 13/10/2007

Quantità di memoria allocata

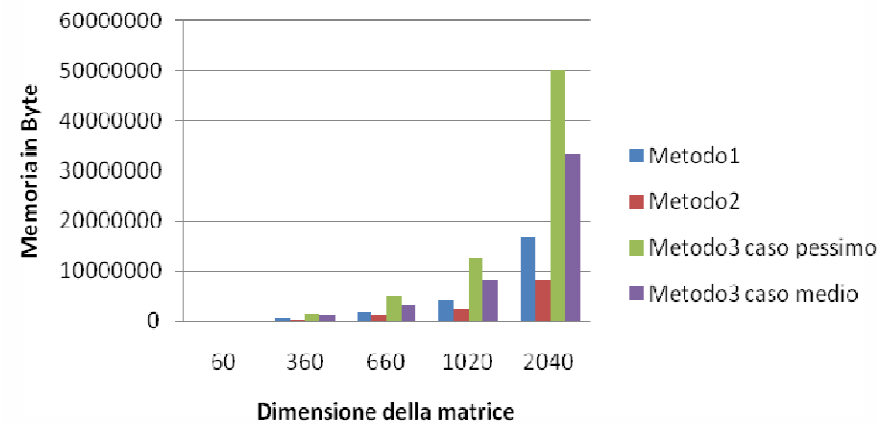
Memoria totale server



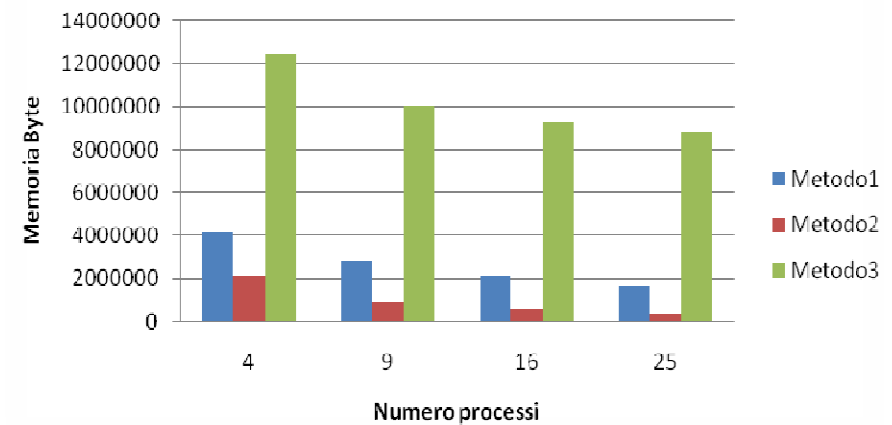
Memoria totale client 4 Processi



Memoria Supporto Client 4 processi

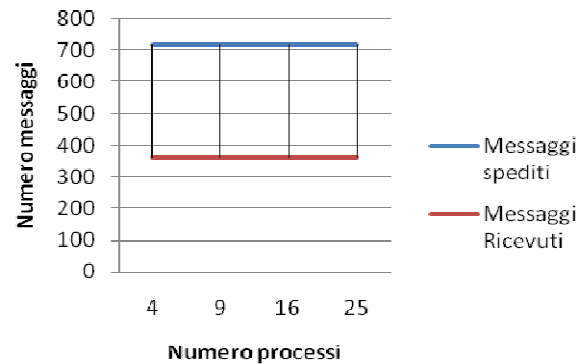


Memoria totale client dimensione 360

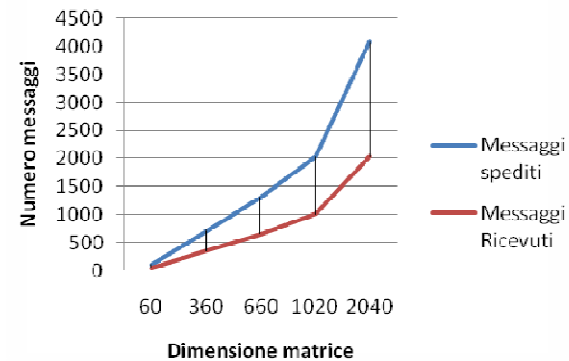


Numero di messaggi spediti

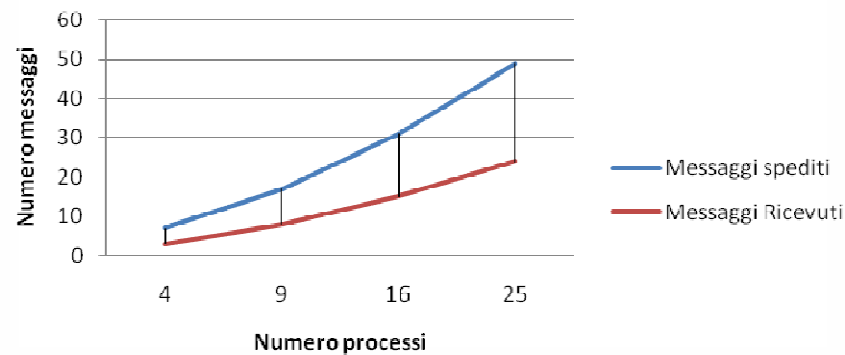
Numero Messaggi metodo 3 min dim 360



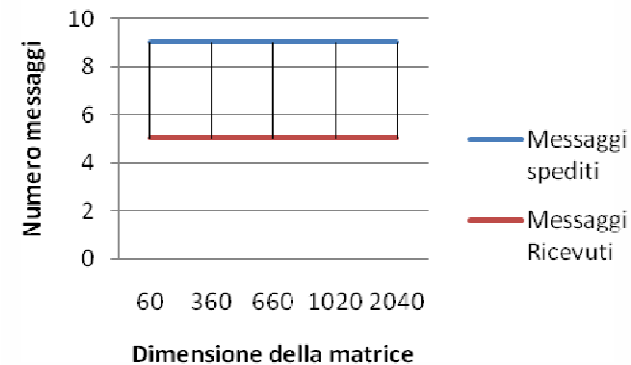
Numero Messaggi metodo 3min nproc 4



Numero Messaggi metodo 1 dimensione 360



Numero Messaggi metodo 1 nproc4



Conclusioni

In questo progetto abbiamo affrontato:

- L'implementazione di un programma per il calcolo del prodotto di matrici dense utilizzando la libreria MPICH.
- Il programma implementa i seguenti metodi:
 - Calcolo sequenziale.
 - Matrix-Matrix Multiplication.
 - Cannon.
 - Processor Farm Multiplication.
- Abbiamo testato il programma per valutare la correttezza degli algoritmi e fornire informazioni riguardanti:
 - L'input e output da file.
 - Il tempo computazionale.
 - L'occupazione di memoria.
 - La ripartizione dei tempi.
 - Gli speedUp.
 - L'efficienza.
 - La simulazione di un ambiente eterogeneo grazie all'allocazione di più processi per macchina.

Bibliografia

- [1] A. Grama, A. Gupta, G. Karypis, V. Kumar. *Introduction to Parallel Computing*, 2°Ed., Addison-Wesley, 2003.
- [2] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, *The Sourcebook of Parallel Computing*. October 2002, ISBN 1-55860-871-0, Morgan Kaufmann Publishers.
- [3] I. Foster. *Designing and Building Parallel Programs*. Addison-Wesley, 1995, Versione online disponibile presso <http://www-unix.mcs.anl.gov/dbpp>
- [4] B. Wilkinson, M. Allen. *Parallel Programming: Techniques and Applications Using Networked Workstation and Parallel Computers*. Prentice-Hall, 1999. Versione online disponibile presso <http://ww1.ucmss.com/books/LFS/CSREA2006/PDP4104.pdf>
- [5] M. Wainer, S. Rahimi and D. Lewis, *An Analysis of Java Distributed Computing Communication Trade-offs: Performance and Programming*, Department of Computer Science Southern Illinois University Carbondale, Illinois
- [6] Complessità metodo cannon
http://www.ipp.mpg.de/de/for/bereiche/stellarator/Comp_sci/CompScience/csep/csep1.phy.ornl.gov/la/node6.html
- [7] Complessità metodo cannon: <http://www.cs.mu.oz.au/498/notes/node30.html>
- [9] Sito ufficiale della libreria MPICH, <http://www-unix.mcs.anl.gov/mpi/mpich1/>
- [10] Sito ufficiale del corso: <http://www.dsi.unive.it/~calpar/>