

Calcolo Parallelo

Valutazione dell'efficienza di algoritmi e software in ambiente parallelo

a.a. 2003-2004

Almerico Murli

Università degli Studi di Napoli Federico II
&
Istituto di Calcolo e Reti ad Alte Prestazioni del CNR
Sezione di Napoli

1

Problema

Valutare l'efficienza di un algoritmo
in ambiente di calcolo parallelo

A. MURLI

Cosa si intende per "EFFICIENZA" ?

Efficienza

A. Murli - Calcolo Parallelo e Distribuito

2

Efficienza di un algoritmo sequenziale

- **COMPLESSITA' di TEMPO $T(n)$**
Numero di operazioni eseguite dall'algoritmo
- **COMPLESSITA' di SPAZIO $S(n)$**
Numero di variabili utilizzate dall'algoritmo

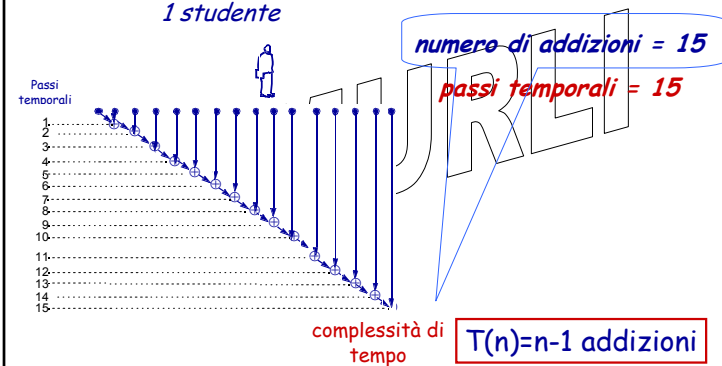
Efficienza

A. Murli - Calcolo Parallelo e Distribuito

3

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO SEQUENZIALE
1 studente



Efficienza

A. Murli - Calcolo Parallelo e Distribuito

4

In un algoritmo sequenziale

Il numero complessivo di operazioni
determina anche

Il numero dei passi temporali
(Il tempo di esecuzione)

In un software sequenziale

l'efficienza del software
dipende dal

tempo di esecuzione
delle $T(n)$ operazioni f.l.p.

Cosa si intende per
efficienza di un algoritmo
in ambiente parallelo?

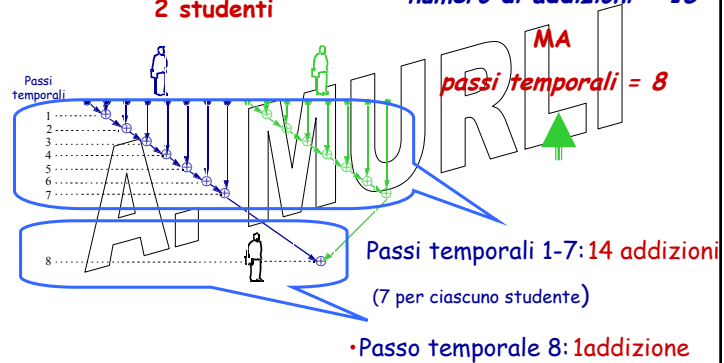
Problema

Si può ancora legare il tempo di esecuzione
al numero delle operazioni eseguite dall'algoritmo ?

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO
2 studenti

numero di addizioni = 15



Efficienza

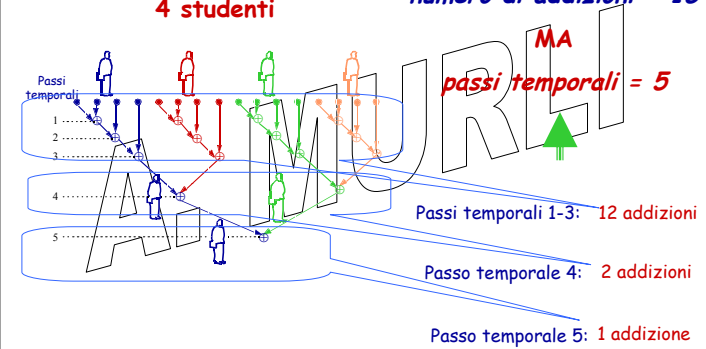
A. Muri - Calcolo Parallelo e Distribuito

9

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO
4 studenti

numero di addizioni = 15



Efficienza

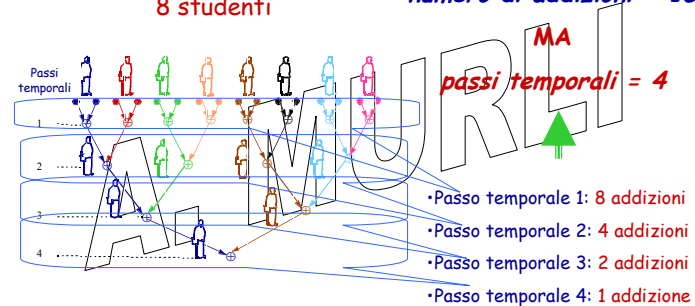
A. Muri - Calcolo Parallelo e Distribuito

10

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO
8 studenti

numero di addizioni = 15



Efficienza

A. Muri - Calcolo Parallelo e Distribuito

11

Nell' algoritmo parallelo per la somma

Il numero delle operazioni non è legato al
Numero dei passi temporali

A. MURRI

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

12

Infatti ...

Un calcolatore parallelo è in grado di eseguire più operazioni

concorrentemente
(allo stesso passo temporale)

Il tempo di esecuzione non è proporzionale alla complessità di tempo
(cioè al numero di operazioni effettuate)

La complessità di tempo non è adatta a misurare l'efficienza di un algoritmo parallelo

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

13

... e allora

Che cosa si intende per efficienza di un algoritmo in ambiente parallelo?

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

14

Esempio:

p =numero di processori

T_p =tempo di esecuzione su p processori

$p = 2$

Ci aspettiamo che T_1 sia il doppio di T_2 :

$$\frac{T_1}{T_2} = 2$$

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

15

Esempio:

p =numero di processori

T_p =tempo di esecuzione su p processori

$p = 4$

Ci aspettiamo che T_1 sia il quadruplo di T_4 :

$$\frac{T_1}{T_4} = 4$$

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

16

In generale

Con p processori ci aspettiamo che T_1 sia p volte T_p

$$\frac{T_1}{T_p} = p$$

ovvero

ci aspettiamo di ridurre p volte il tempo di esecuzione

IDEA

Misuriamo quanto si riduce il tempo di esecuzione su p processori rispetto al tempo di esecuzione Su 1 processore

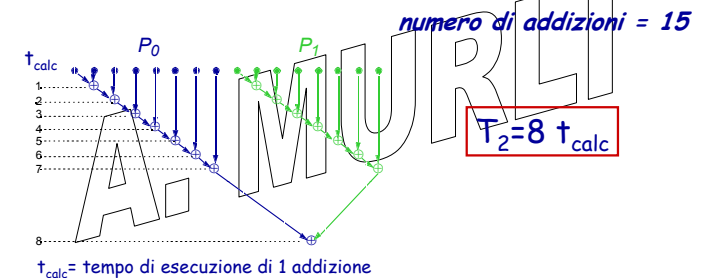
Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO SEQUENZIALE $p=1$



Esempio: calcolo della somma di $n=16$ numeri

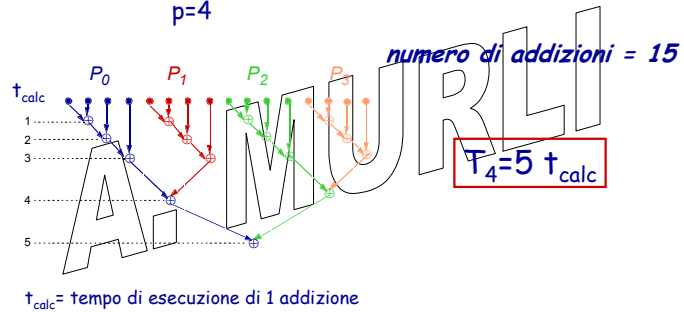
ALGORITMO PARALLELO $p=2$



Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO

$p=4$



Efficienza

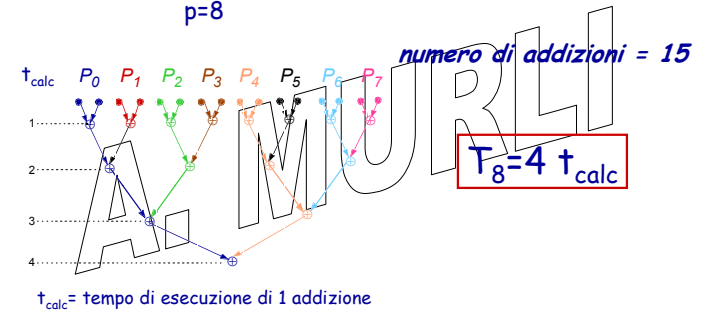
A. Muri - Calcolo Parallelo e Distribuito

21

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO

$p=8$



Efficienza

A. Muri - Calcolo Parallelo e Distribuito

22

Come si esprime T_p
 Ovvero il tempo di esecuzione
 Per il calcolo della somma di N numeri su p
 Processori?

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

23

In conclusione ...

p	T_p
1	$15 t_{\text{calc}}$
2	$8 t_{\text{calc}}$
4	$5 t_{\text{calc}}$
8	$4 t_{\text{calc}}$

Qual è l'algoritmo più veloce?

Quanto è più veloce di quello sequenziale?

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

24

Esempio: calcolo della somma di $n=16$ numeri

p	T_p	$\frac{T_1}{T_p}$
1	15 t_{calc}	1.00
2	8 t_{calc}	1.88
4	5 t_{calc}	3.00
8	4 t_{calc}	3.75

Maggiore
riduzione del
tempo ovvero
maggiore
aumento della
velocità

L'algoritmo su 8 processori è il più veloce
E' più veloce di 3.75 volte di quello su 1 processore

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

25

Speedup

In generale

$$S_p = \frac{T_1}{T_p}$$

SPEEDUP

su p processori

misura la riduzione del tempo di esecuzione, ovvero
l'aumento della velocità di esecuzione, rispetto all'
algoritmo su 1 processore

SPEEDUP IDEALE

$$S_p^{ideale} = p$$

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

26

Quindi

Se si vuole calcolare la somma di 16
numeri nel minor tempo possibile
l'algoritmo su 8 processori è da
preferire

Infatti, aumentando il numero di processori si riduce
il tempo impiegato per eseguire le operazioni
richieste

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

27

MA....

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

28

Esempio: calcolo della somma di $n=16$ numeri

p	Speedup ottenuto	Speedup ideale
2	1.88	2
4	3.00	4
8	3.75	8

Lo speedup su 8 processori è il maggiore

MA

Lo speedup su 2 processori è "il più vicino" allo speedup ideale...

Cioè

ho utilizzato 8 processori per avere un incremento di

Appena 4 volte

In altre parole

... lo speed up non basta!

... e allora ?

Esempio: calcolo della somma di $n=16$ numeri

... se si rapporta lo speedup al numero di processori...

p	S_p	$\frac{S_p}{p}$
2	1.88	0.94
4	3.00	0.75
8	3.75	0.47

Rapporto più grande

maggior sfruttamento dei processori per $p=2$

In altre parole

l'utilizzo di un maggior numero di processori NON è sempre una garanzia di sviluppo di algoritmi paralleli "efficaci"

OVVERO

Di algoritmi che sfruttano tutte le risorse della macchina parallela !

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

33

Come misurare se e quanto è stata sfruttato
Il calcolatore parallelo ?

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

34

Efficienza

$$E_p = \frac{S_p}{p}$$

EFFICIENZA
su p processori

misura quanto l'algoritmo sfrutta il parallelismo del calcolatore

EFFICIENZA IDEALE

$$E_p^{\text{ideale}} = \frac{S_p^{\text{ideale}}}{p} = 1$$

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

35

Infatti, per la somma con $p=8$



Efficienza

A. Muri - Calcolo Parallelo e Distribuito

36

Problema

Se si esegue l'algoritmo su un **calcolatore MIMD a memoria distribuita**, il tempo di esecuzione dipende solo dal numero di operazioni eseguite in differenti passi temporali?

Efficienza

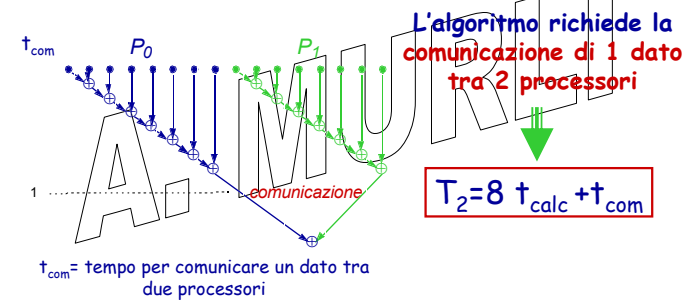
A. Muri - Calcolo Parallelo e Distribuito

37

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO

$p=2$



Efficienza

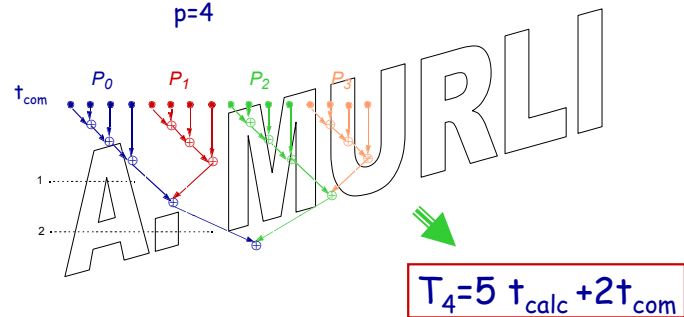
A. Muri - Calcolo Parallelo e Distribuito

38

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO

$p=4$



Efficienza

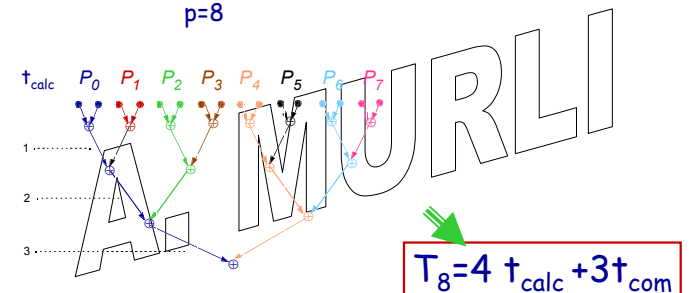
A. Muri - Calcolo Parallelo e Distribuito

39

Esempio: calcolo della somma di $n=16$ numeri

ALGORITMO PARALLELO

$p=8$



Efficienza

A. Muri - Calcolo Parallelo e Distribuito

40

Esempio: calcolo della somma di $n=16$ numeri

Supponiamo $\frac{t_{com}}{t_{calc}} = 2$

p	T_p	S_p	E_p
2	$8t_{calc}$	1.88	0.94
4	$5t_{calc}$	3.00	0.75
8	$4t_{calc}$	3.75	0.47

Trascurando la comunicazione

$T_1 = 15T_{calc}$

p	T_p	S_p	E_p
2	$10t_{calc}$	1.50	0.75
4	$9t_{calc}$	1.67	0.42
8	$8t_{calc}$	1.88	0.24

Considerando la comunicazione

Considerando o meno il **tempo di comunicazione** le prestazioni dell' algoritmo possono cambiare notevolmente!

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

41

Overhead di comunicazione

T_p^{com} = tempo di comunicazione dell' algoritmo su p processori

T_p^{calc} = tempo di calcolo dell' algoritmo su p processori

$$OC_p^{totale} = \frac{T_p^{com}}{T_p^{calc}}$$

Overhead di comunicazione totale

Fornisce una misura del "peso" della comunicazione sul tempo di esecuzione dell' algoritmo

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

42

Esempio: calcolo della somma di $n=16$ numeri

Posto ad esempio:

$$OC_p = \frac{t_{com}}{t_{calc}} = 2$$

Overhead di comunicazione unitario

p	T_p^{calc}	T_p^{com}	OC_p^{totale}
2	$8t_{calc}$	$1t_{com}$	0.25
4	$5t_{calc}$	$2t_{com}$	0.80
8	$4t_{calc}$	$3t_{com}$	1.50

Su 8 processori il tempo di comunicazione pesa di più rispetto al tempo di esecuzione

Efficienza

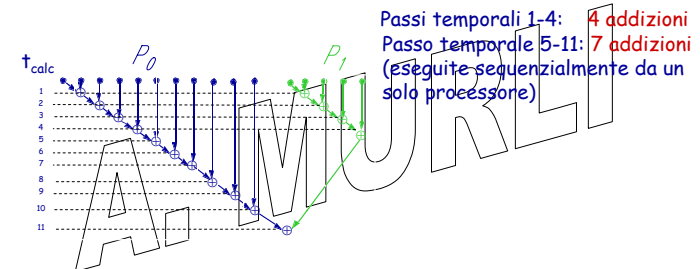
A. Muri - Calcolo Parallelo e Distribuito

43

Esempio

$n=16$

$p=2$



$$T_2 = 11 t_{calc}$$

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

44

Problema

L'algoritmo precedente **non e' bilanciato**

Una cattiva ripartizione del carico di lavoro tra i processori induce un tempo di attesa e quindi un aumento del tempo di esecuzione

Il tempo di esecuzione dipende dal bilanciamento del carico computazionale tra i processori

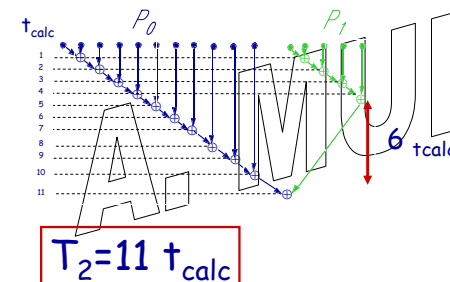
Efficienza

A. Muri - Calcolo Parallelo e Distribuito

45

Esempio

$n=16$ $p=2$



Di cui $6 t_{calc}$ sono di "sincronizzazione"

Tempo necessario per Coordinare il lavoro tra i processori

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

46

In conclusione

La valutazione delle prestazioni di un algoritmo parallelo deve tener conto di:

- Numero di processori
- Tempo di calcolo
- Tempo di comunicazione
- Bilanciamento del carico computazionale
- Tempo di sincronizzazione
-

Necessità di utilizzare più parametri di valutazione
($T_p, S_p, E_p, Oc_p, \dots$)

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

47

In conclusione

Inoltre, differenti caratteristiche hw/sw dei calcolatori paralleli:

- Numero di processori
- Architettura e potenza dei processori
- Tipo e numero di memorie
- Connessione tra i processori
- Connessione tra processori e memorie
- Omogeneità/eterogeneità del sistema
- Software message-passing
-

Una valutazione effettiva delle prestazioni di un algoritmo richiede l'implementazione in uno specifico ambiente di calcolo e la misura di T_p

Efficienza

A. Muri - Calcolo Parallelo e Distribuito

48

Quale algoritmo scegliere
per misurare T_1 ?

PRIMA SCELTA

T_1 = tempo di esecuzione dell'algoritmo
parallelo su 1 processore

S_p dà informazioni su quanto
l'algoritmo parallelo si presti all'
implementazione parallela

Difficoltà:

l'algoritmo parallelo su 1 processore potrebbe eseguire più
operazioni del necessario

SECONDA SCELTA

T_1 = tempo di esecuzione del migliore
algoritmo sequenziale

S_p fornisce una misura della
riduzione effettiva del tempo nella
risoluzione di un problema con p processori

Difficoltà:

- individuazione del miglior algoritmo sequenziale
- disponibilità del software che implementa tale algoritmo (se si vuole misurare effettivamente T_1)