

Calcolo Parallelo e Distribuito

Somma di N numeri

PROBLEMA:

Calcolo della somma di N
numeri

$$a_0 + a_1 + \dots + a_{N-1}$$

su un calcolatore parallelo
tipo **MIMD** con p processori

Somma di N numeri

Su un calcolatore monoprocesore la somma è
calcolata eseguendo le $N-1$ addizioni una per volta
secondo un ordine prestabilito

```
sumtot := a0
sumtot := sumtot + a1
sumtot := sumtot + a2
...
sumtot := sumtot + aN-1
```

Qual è

l'ALGORITMO PARALLELO?

Somma di N numeri

Su un calcolatore monoprocesore la somma è
calcolata eseguendo le $N-1$ addizioni una per volta
secondo un ordine prestabilito

```
begin
  sumtot := a0;
  for i=1 to N-1 do
    sumtot := sumtot + ai;
  endfor
end
```

Qual è

l'ALGORITMO PARALLELO?

CALCOLO PARALLELO

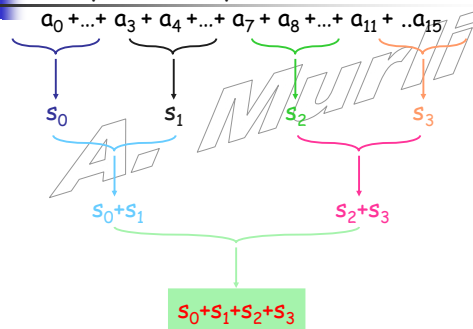
Decomporre un problema di dimensione N in P sottoproblemi di dimensione N/P e risolverli **contemporaneamente** su più calcolatori



A. Murli - Lezioni di Calcolo Parallelo e Distribuito

5

Esempio: $N=16, p=4$

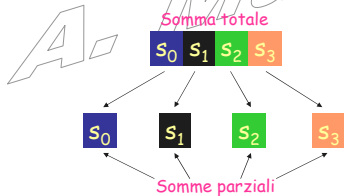


A. Murli - Lezioni di Calcolo Parallelo e Distribuito

6

IDEA

Suddividere la somma in somme parziali ed assegnare ciascuna somma parziale ad un processore

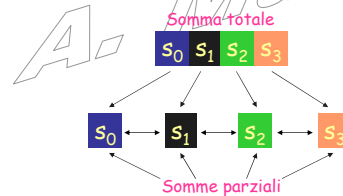


A. Murli - Lezioni di Calcolo Parallelo e Distribuito

7

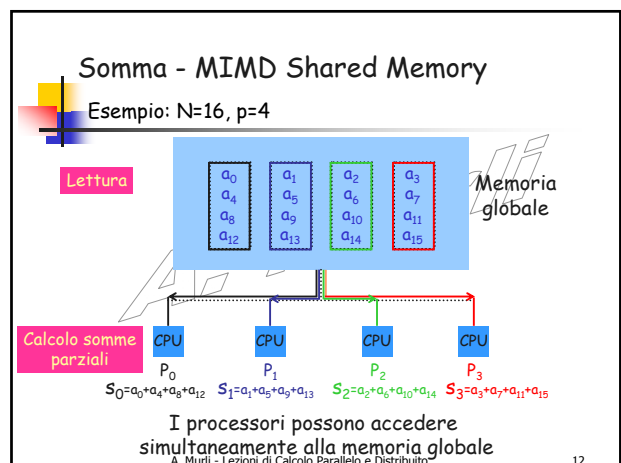
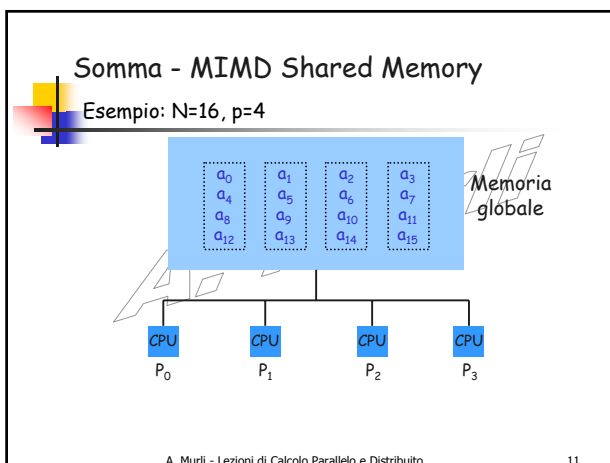
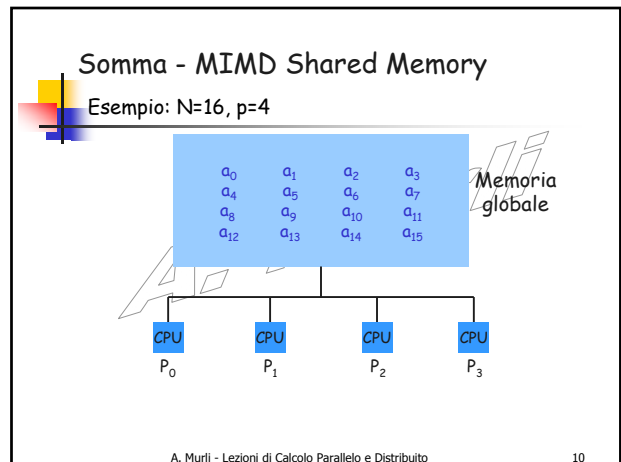
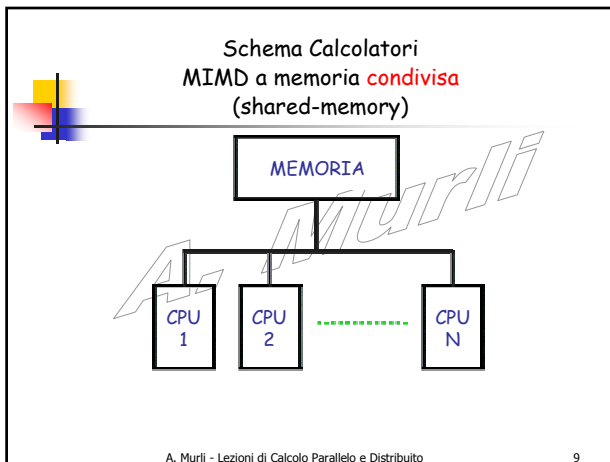
IDEA

Le somme parziali devono poi essere **combinate in modo opportuno** per ottenere la somma totale



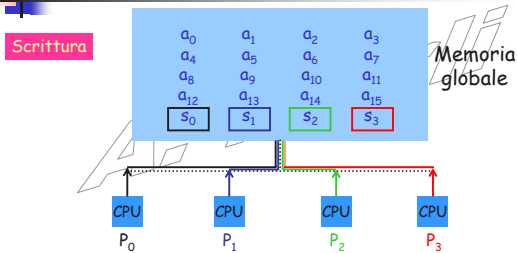
A. Murli - Lezioni di Calcolo Parallelo e Distribuito

8



Somma - MIMD Shared Memory

Esempio: $N=16$, $p=4$



I processori possono accedere
simultaneamente alla memoria globale

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

13

Somma - MIMD Shared Memory

Come calcolare
la somma totale?

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

14

Somma - MIMD Shared Memory

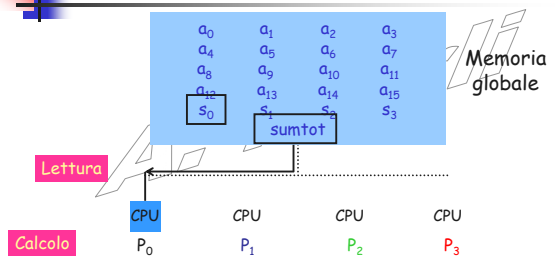
Ogni processore P_i
calcola la sua somma parziale s_i
deve aggiungere tale valore ad una variabile
(ad es. **sumtot**) che conterrà la somma finale.
I processori devono accedere a **sumtot**
uno alla volta.

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

15

Somma - MIMD Shared Memory

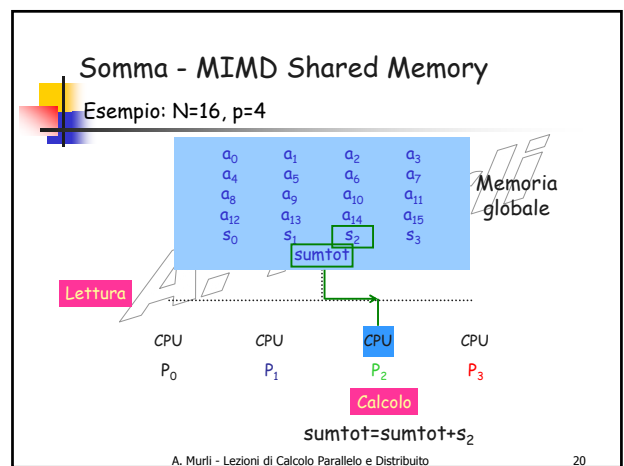
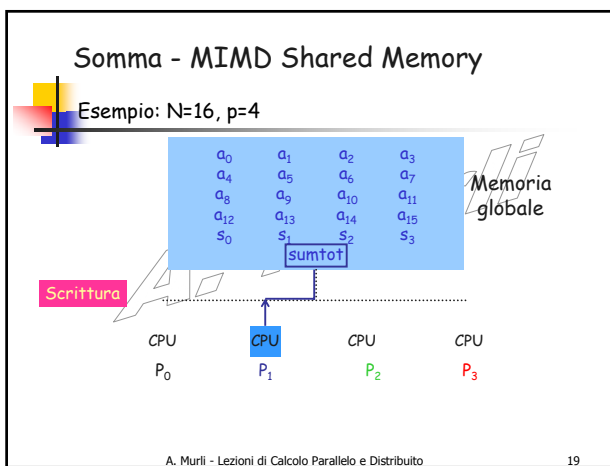
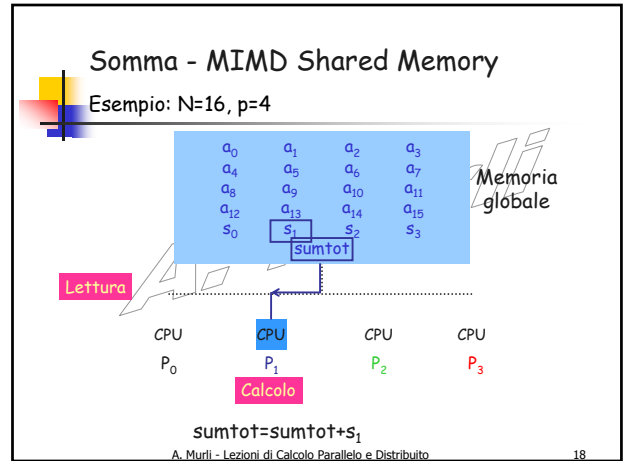
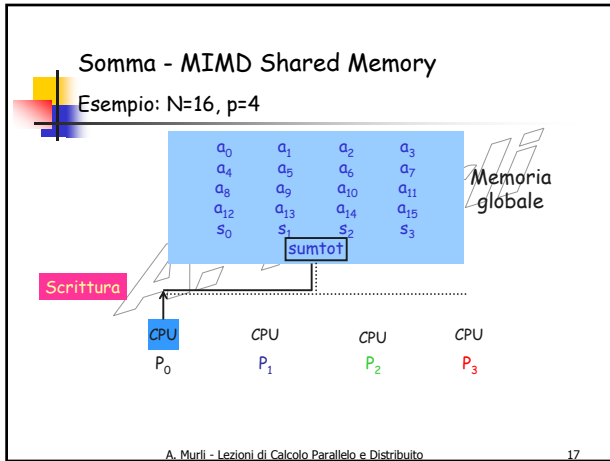
Esempio: $N=16$, $p=4$

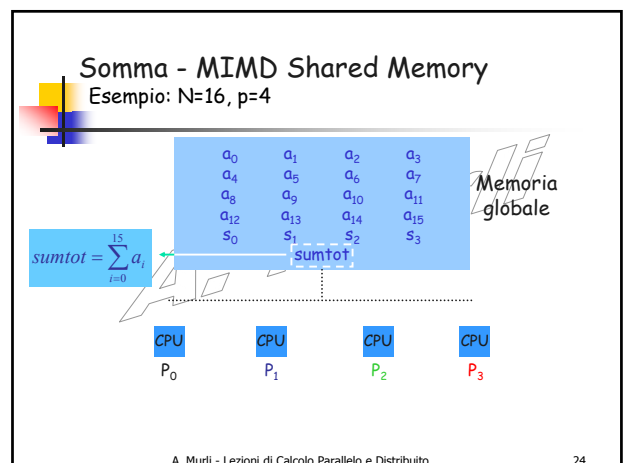
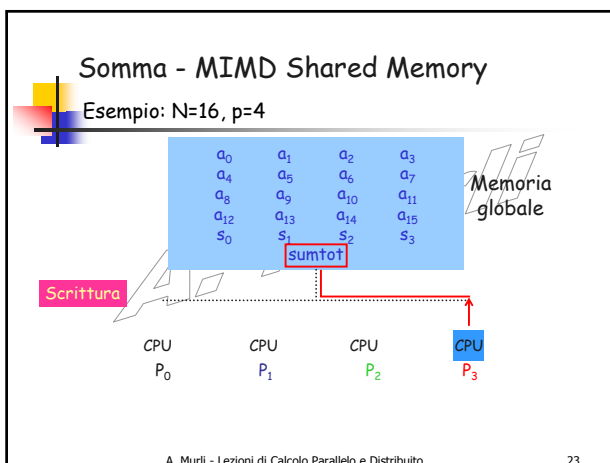
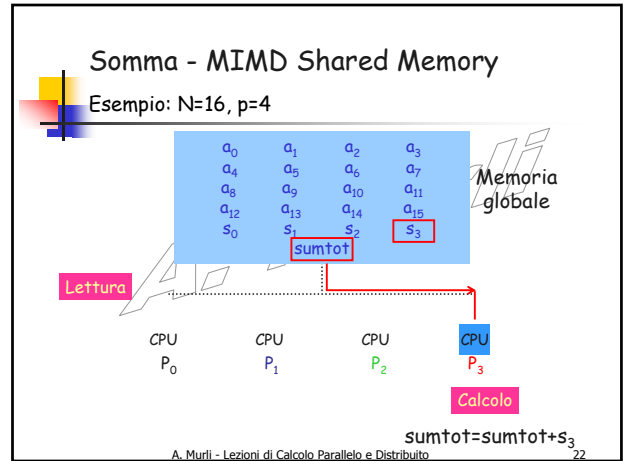
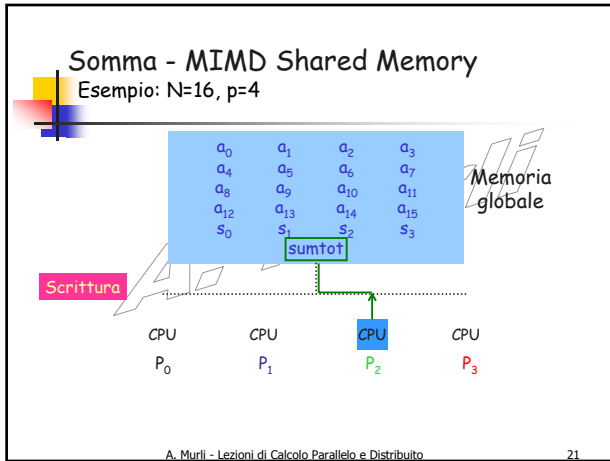


$sumtot = sumtot + s_0$

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

16





Somma - MIMD Shared Memory

Affinchè il valore di *sumtot* sia **correttamente aggiornato** è necessario che ciascun processore abbia **accesso esclusivo** a tale variabile durante il suo aggiornamento

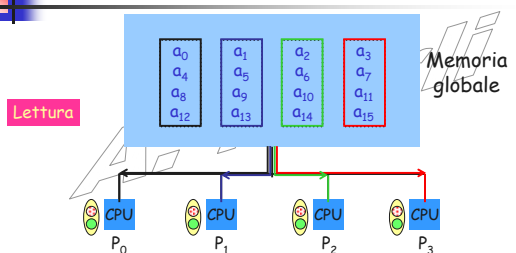
Sincronizzazione degli accessi in memoria per ottenere il **parallelismo**

PROBLEMA

Come gestire la
sincronizzazione tra i processori ?

Somma - MIMD Shared Memory

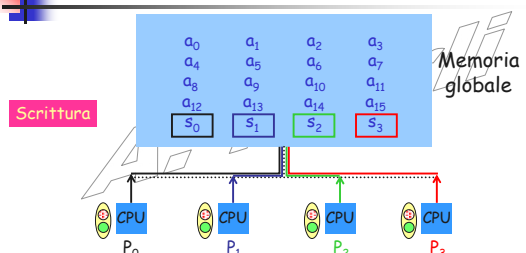
Esempio: $N=16, p=4$



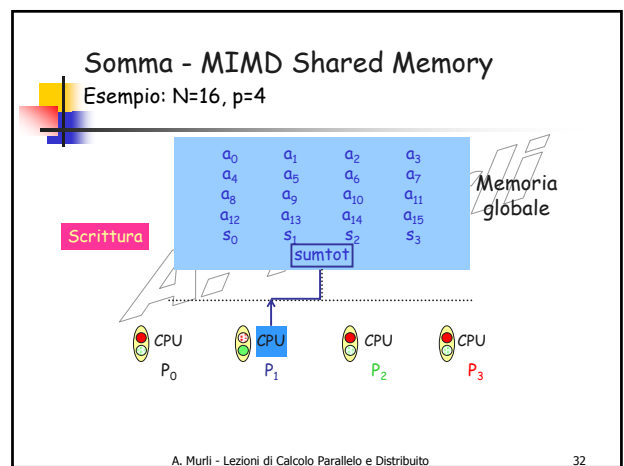
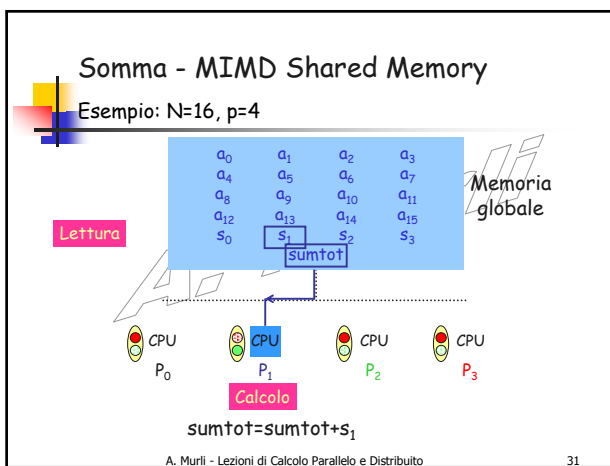
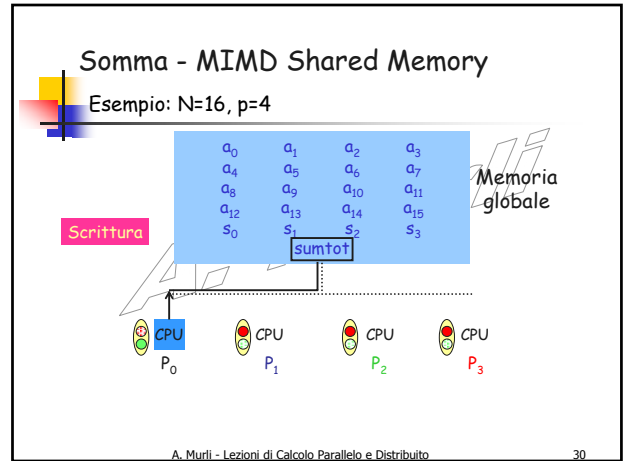
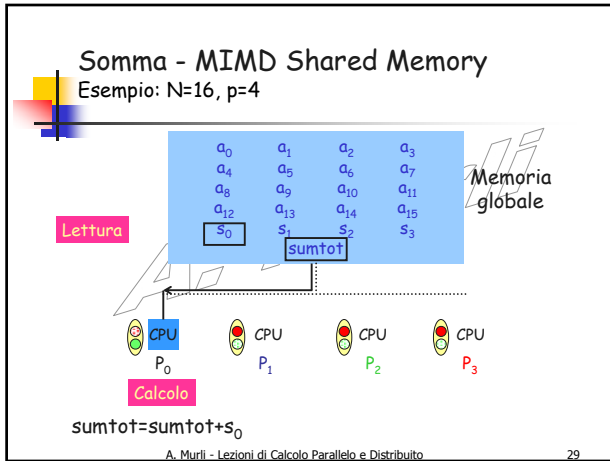
I processori possono accedere simultaneamente alla memoria globale su dati differenti

Somma - MIMD Shared Memory

Esempio: $N=16, p=4$

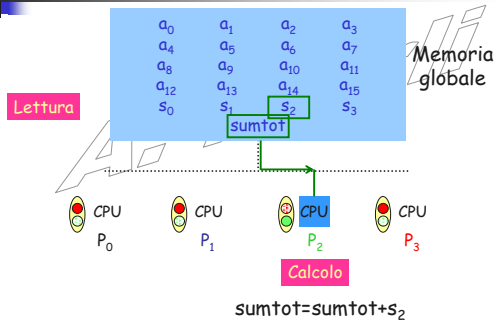


I processori possono accedere simultaneamente alla memoria globale su dati differenti



Somma - MIMD Shared Memory

Esempio: $N=16, p=4$

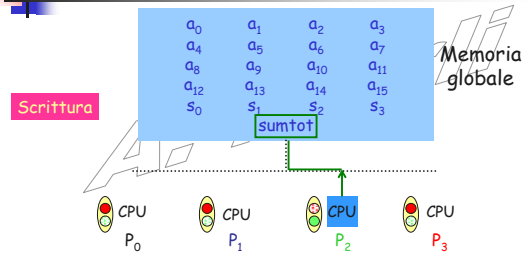


A. Muri - Lezioni di Calcolo Parallelo e Distribuito

33

Somma - MIMD Shared Memory

Esempio: $N=16, p=4$

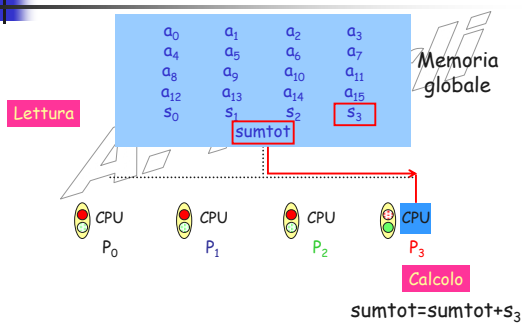


A. Muri - Lezioni di Calcolo Parallelo e Distribuito

34

Somma - MIMD Shared Memory

Esempio: $N=16, p=4$

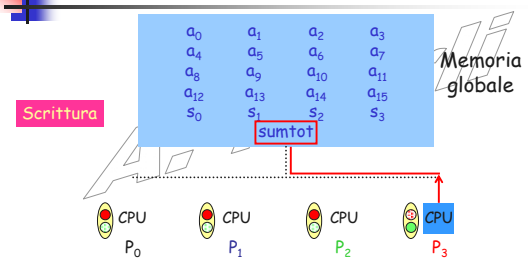


A. Muri - Lezioni di Calcolo Parallelo e Distribuito

35

Somma - MIMD Shared Memory

Esempio: $N=16, p=4$

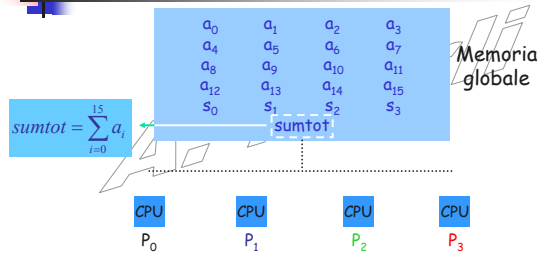


A. Muri - Lezioni di Calcolo Parallelo e Distribuito

36

Somma - MIMD Shared Memory

Esempio: $N=16, p=4$



Semafori

I **Semafori** sono contatori usati per controllare l'accesso alle risorse condivise fra più processi (threads) che condividono la medesima memoria

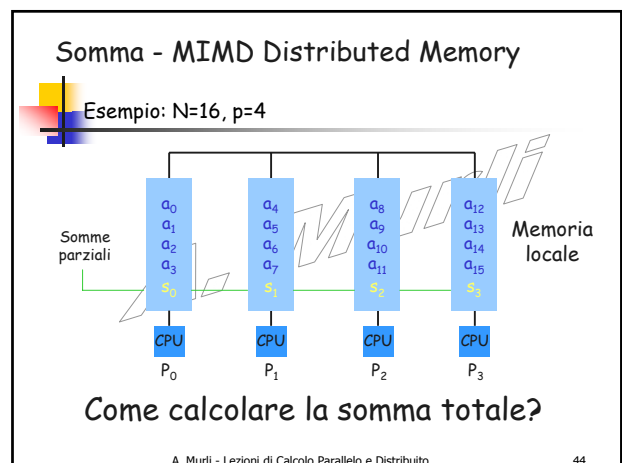
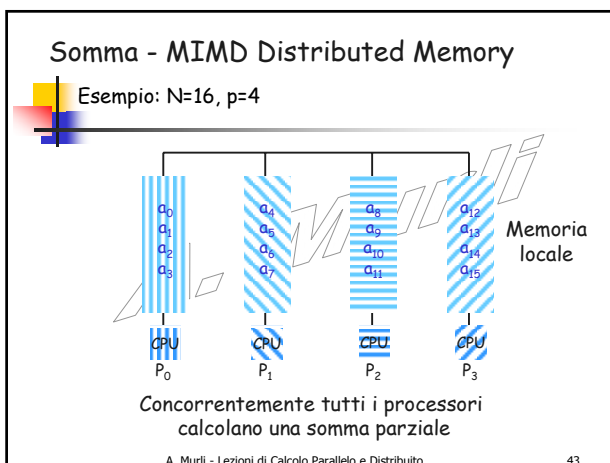
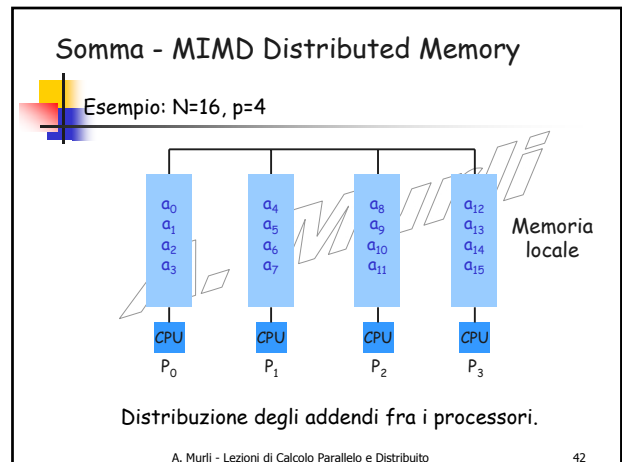
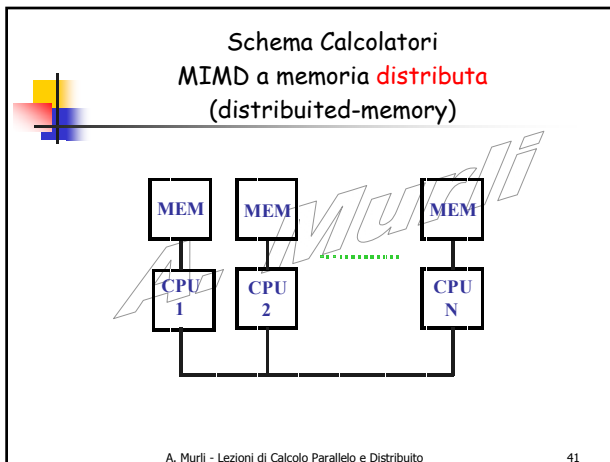
Sono utilizzati come meccanismi di **bloccaggio** per evitare che altri threads accedano alla memoria condivisa **contemporaneamente** al thread che la sta già utilizzando.

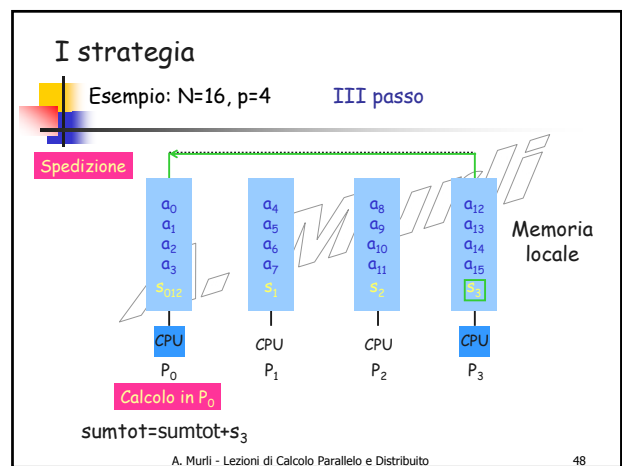
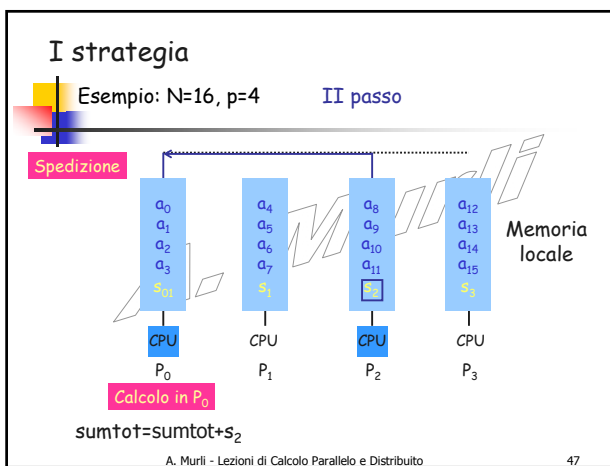
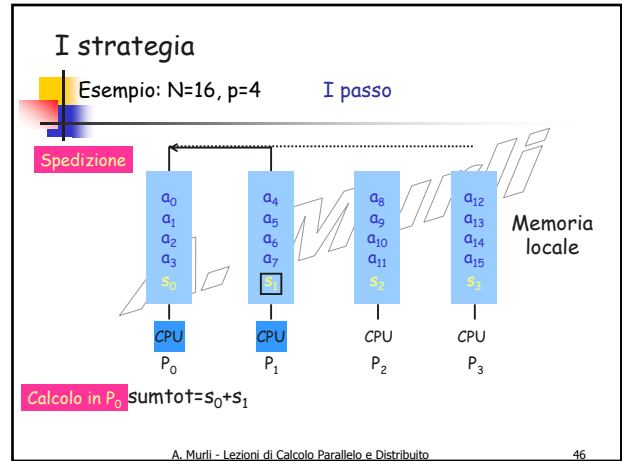
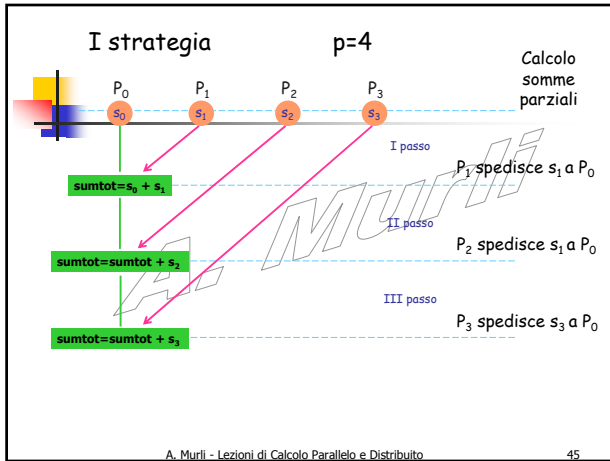
Algoritmo per la somma di $N=kp$ numeri su MIMD-Shared Memory

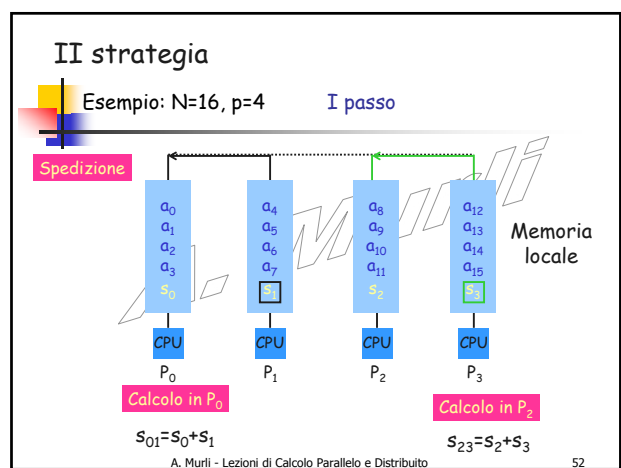
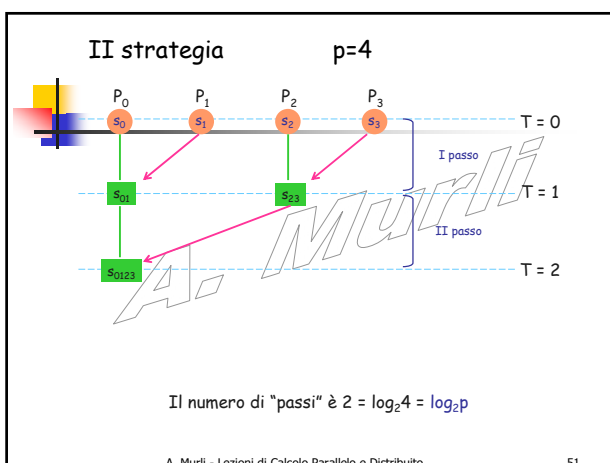
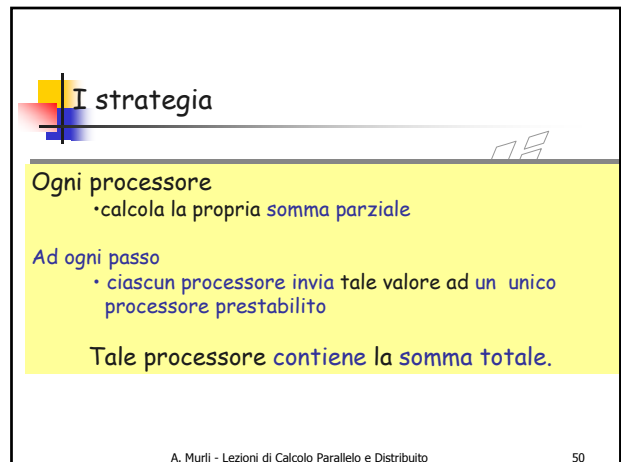
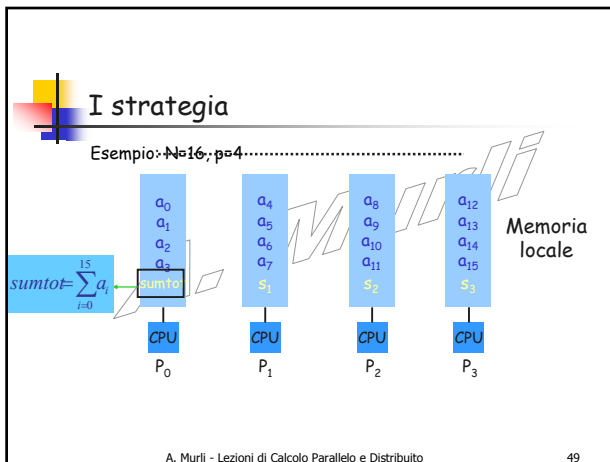
```
begin
  forall  $P_i, 0 \leq i \leq p-1$  do
    sumtot := 0
     $s_i := 0$ 
     $h := i * (n/p)$ 
    for  $j = h$  to  $h+(n/p)-1$  do
       $s_i := s_i + a_j$ 
    endfor
    lock(sumtot)
    sumtot := sumtot +  $s_i$ 
    unlock(sumtot)
  end forall
end
```

Uno strumento per lo sviluppo di software in ambiente di calcolo MIMD-Shared Memory

Open Multi Processing
OpenMP



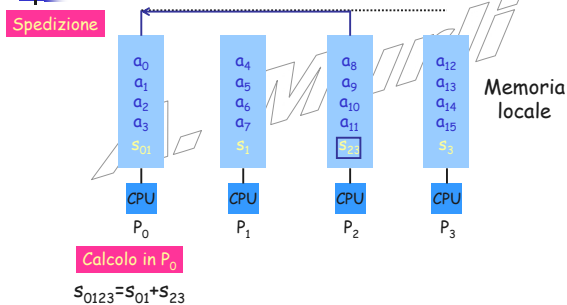




II strategia

Esempio: $N=16, p=4$

II passo



A. Muri - Lezioni di Calcolo Parallelo e Distribuito

53

II strategia

Ogni processore

· calcola la propria somma parziale.

Ad ogni passo,

- coppie distinte di processori comunicano contemporaneamente
- in ogni coppia, un processore invia all'altro la propria somma parziale che provvede all'aggiornamento della somma

Il risultato è in un unico processore prestabilito

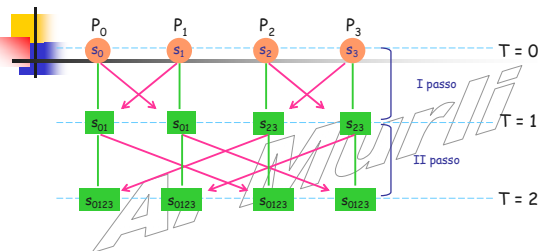
Operazioni concorrenti

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

54

III strategia

$p=4$



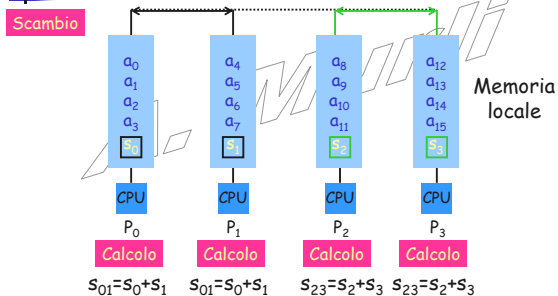
A. Muri - Lezioni di Calcolo Parallelo e Distribuito

55

III strategia

Esempio: $N=16, p=4$

I passo



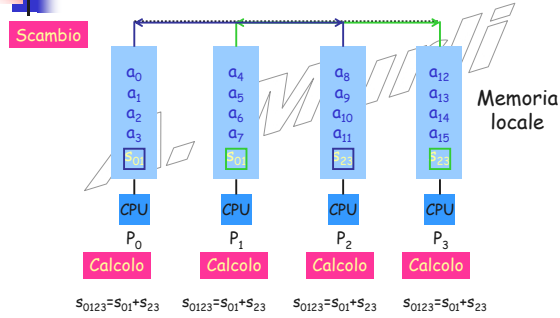
A. Muri - Lezioni di Calcolo Parallelo e Distribuito

56

III strategia

Esempio: $N=16, p=4$

II passo



A. Muri - Lezioni di Calcolo Parallelo e Distribuito

57

III strategia

Ogni processore

- calcola la propria somma parziale.

Ad ogni passo

- coppie distinte di processori comunicano contemporaneamente:
- in ogni coppia i processori si scambiano le proprie somme parziali

Il risultato è in **tutti** i processori

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

58

I, II, III strategia

Ogni processore calcola la sua **somma parziale** ed **invia** tale valore agli altri processori in modo da ottenere la somma totale

PARALLELISMO

COMUNICAZIONE
TRA I PROCESSORI

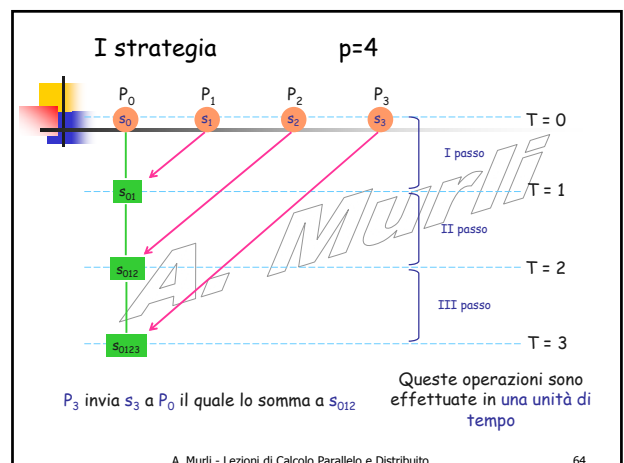
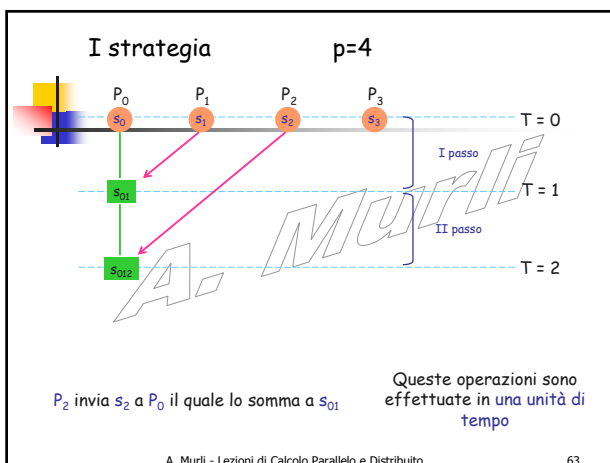
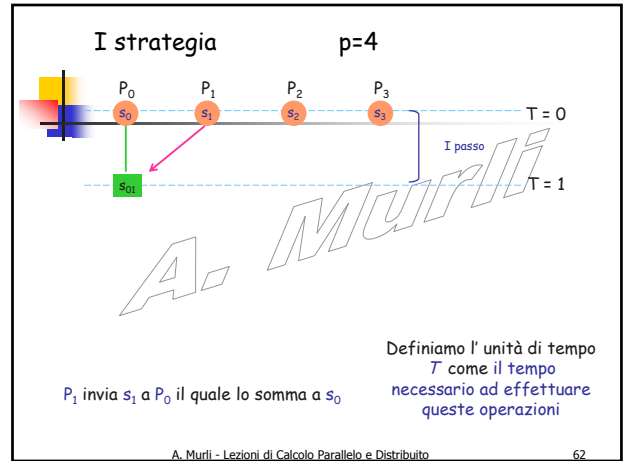
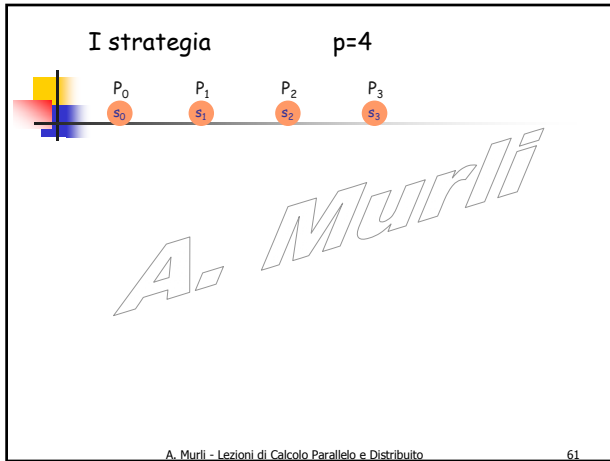
A. Muri - Lezioni di Calcolo Parallelo e Distribuito

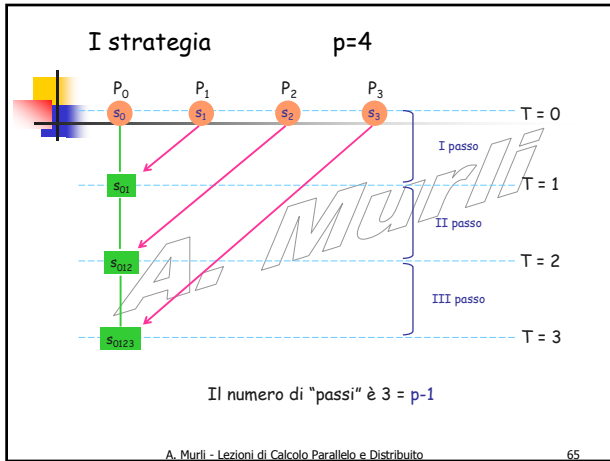
59

Dopo quanto tempo si ha la somma finale in ciascuna strategia ?

A. Muri - Lezioni di Calcolo Parallelo e Distribuito

60





I strategia $p=4$

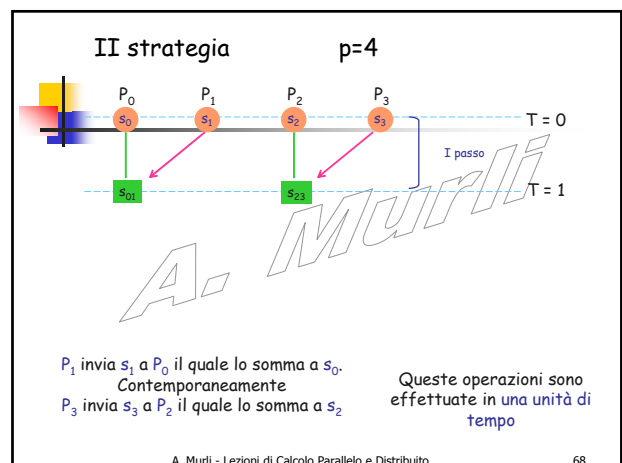
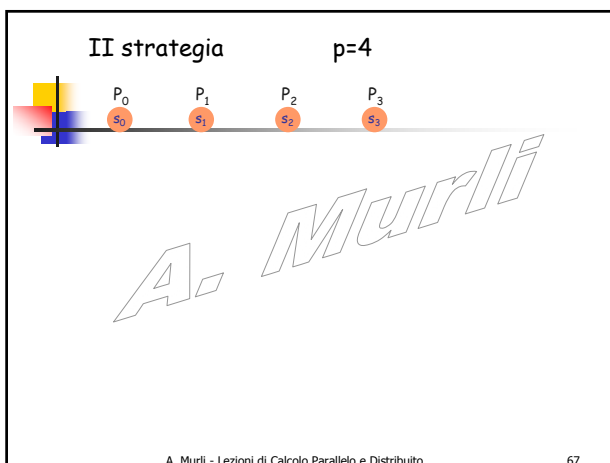
Le unità di tempo sono scandite dalle operazioni che possono essere eseguite contemporaneamente da p processori

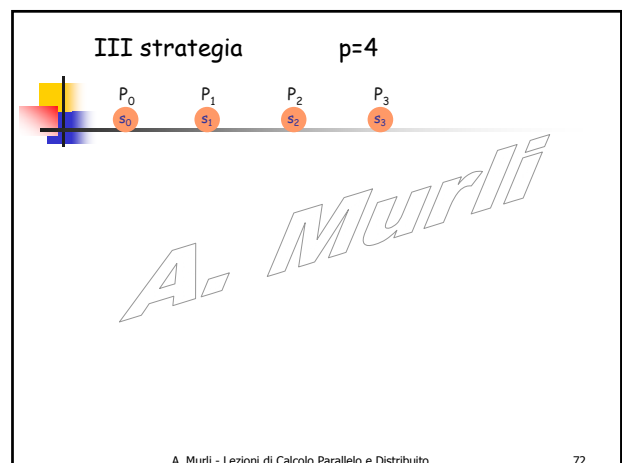
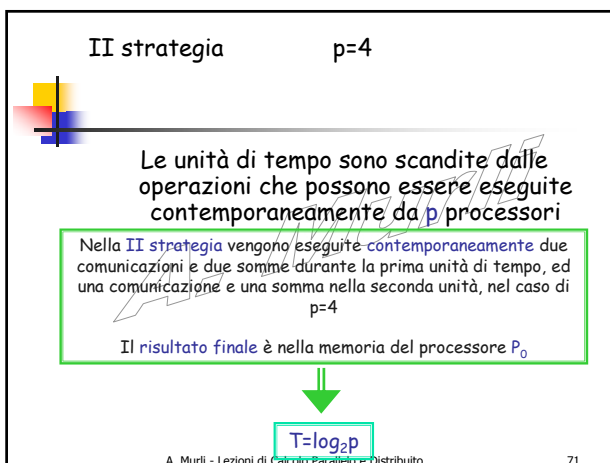
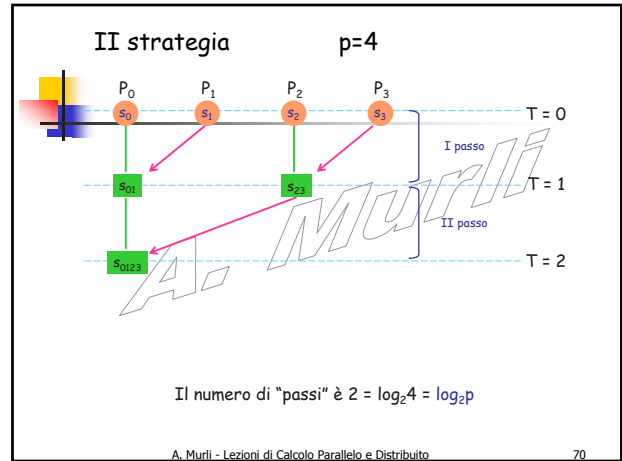
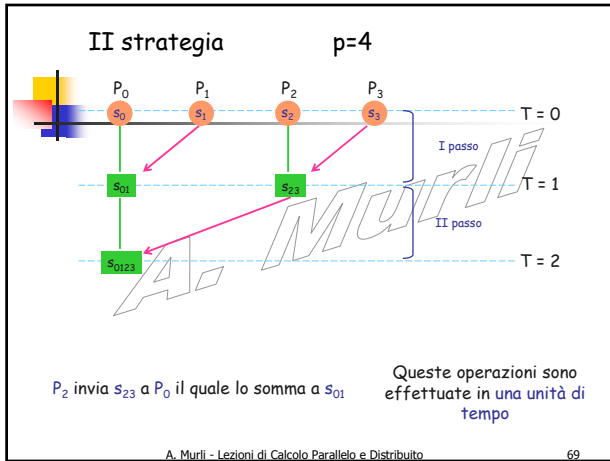
Nella **I strategia** il tempo è scandito dall'attività del processore P_0 che è incaricato di eseguire la **raccolta delle somme parziali** per ottenere la **somma finale**

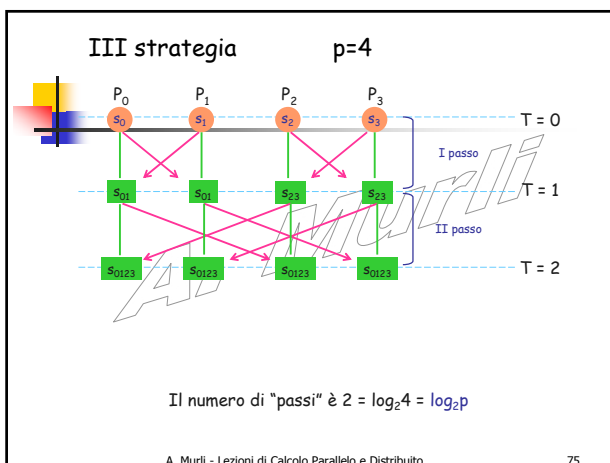
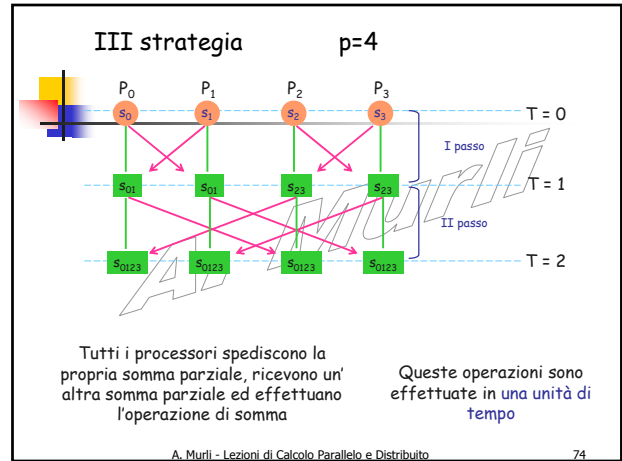
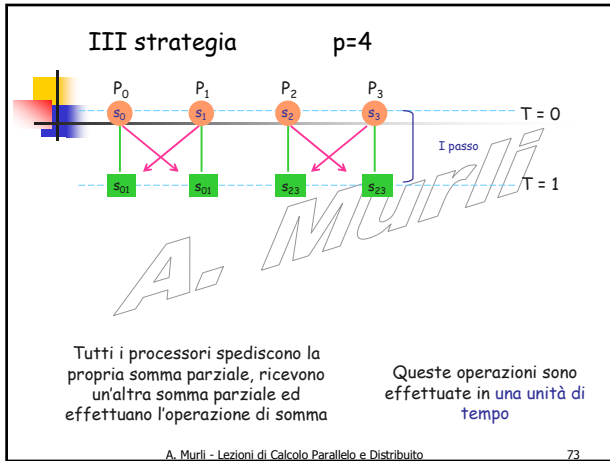
In ogni unità di tempo avviene una comunicazione tra processori

$T=p-1$

A. Murli - Lezioni di Calcolo Parallelo e Distribuito 66







Algoritmo per la somma di $N=kp$ numeri su **MIMD-Distributed Memory**

I Strategia

```
begin
  forall  $P_i$ ,  $0 \leq i \leq p-1$  do
     $sum_i := 0$ 
     $h := i * (n/p)$ 
    for  $j = h$  to  $h+(n/p)-1$  do
       $sum_i := sum_i + a_j$ 
    endfor
    if  $P_0$  then
      for  $k = 1$  to  $p-1$  do
        recv( $sum_k$ ,  $P_k$ )
         $sumtot := sumtot + sum_k$ 
      endfor
    else if  $P_i$  then
      send( $sum_i$ ,  $P_0$ )
    endif
  endforall
end
```

A. Murli - Lezioni di Calcolo Parallelo e Distribuito

77

Uno strumento software per lo sviluppo di
algoritmi in ambiente di calcolo
MIMD-Distributed Memory

Message Passing Interface
MPI

A. Murli - Lezioni di Calcolo Parallelo e Distribuito

78

Fine Lezione

A. Murli - Lezioni di Calcolo Parallelo e Distribuito

79