

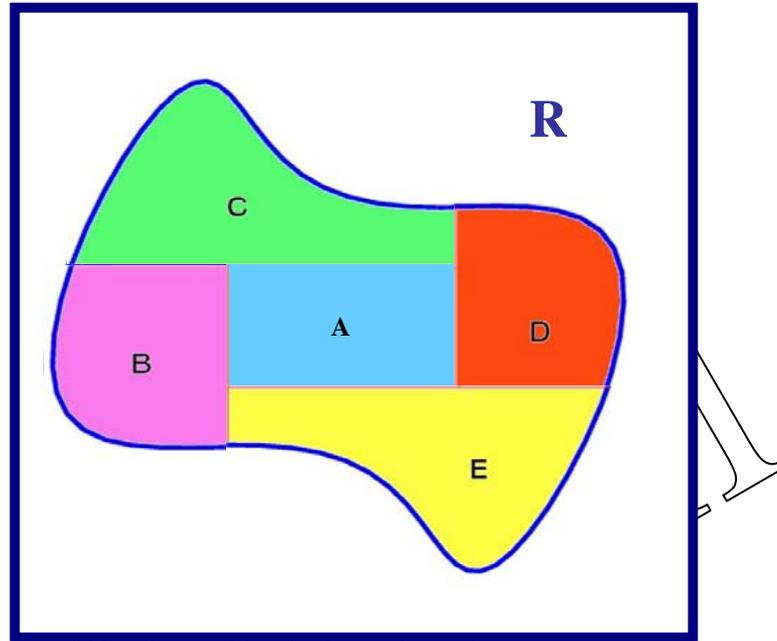
# PROBLEMA

Progettare un algoritmo parallelo per architettura MIMD-DM per il calcolo dell'integrale definito:

$$\int_R f(t_1, t_2, \dots, t_n) dt_1 dt_2 \dots dt_n$$

R = dominio di integrazione

## Esempio: n=2



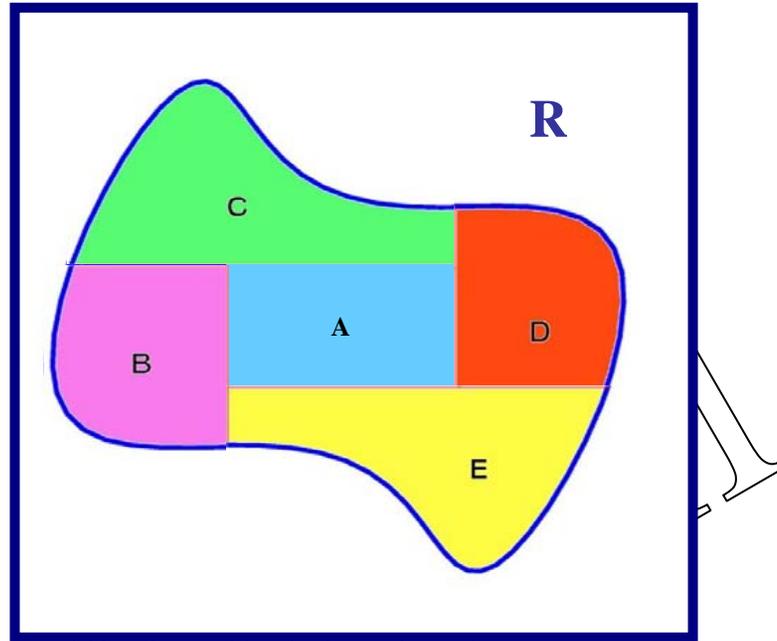
$$\int_R f(t_1, t_2) dt_1 dt_2$$

$$\text{Area}(\mathbf{R}) = \text{Area}(\mathbf{E}) + \text{Area}(\mathbf{D}) + \text{Area}(\mathbf{C}) + \text{Area}(\mathbf{B}) + \text{Area}(\mathbf{A})$$

**Calcolo dell'integrale**

**Decomposizione del dominio**

## Esempio: n=2



$$\int_R f(t_1, t_2) dt_1 dt_2$$

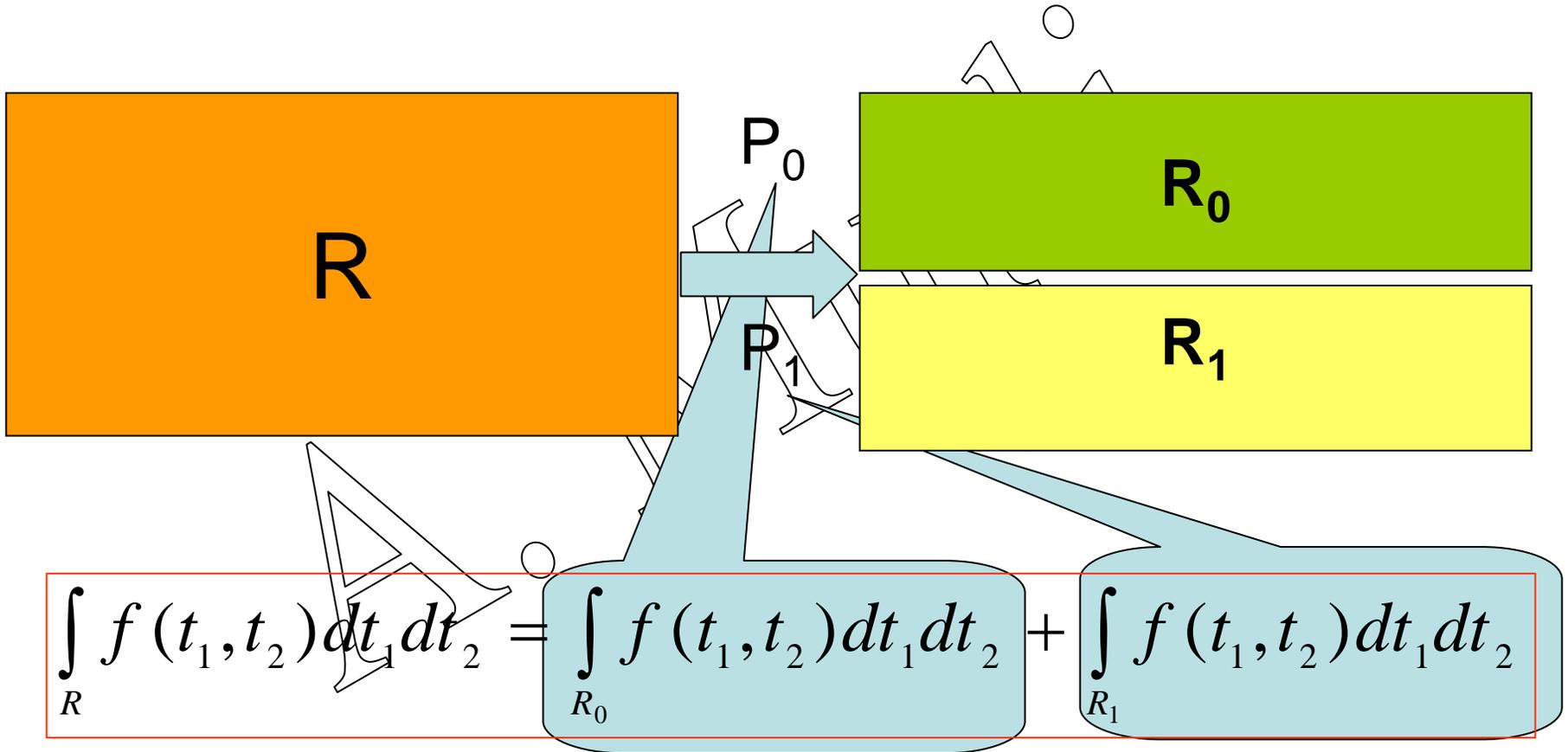
$$\text{Area}(\mathbf{R}) = \text{Area}(\mathbf{E}) + \text{Area}(\mathbf{D}) + \text{Area}(\mathbf{C}) + \text{Area}(\mathbf{B}) + \text{Area}(\mathbf{A})$$

**Decomposizione del dominio**

**Introduzione del parallelismo**

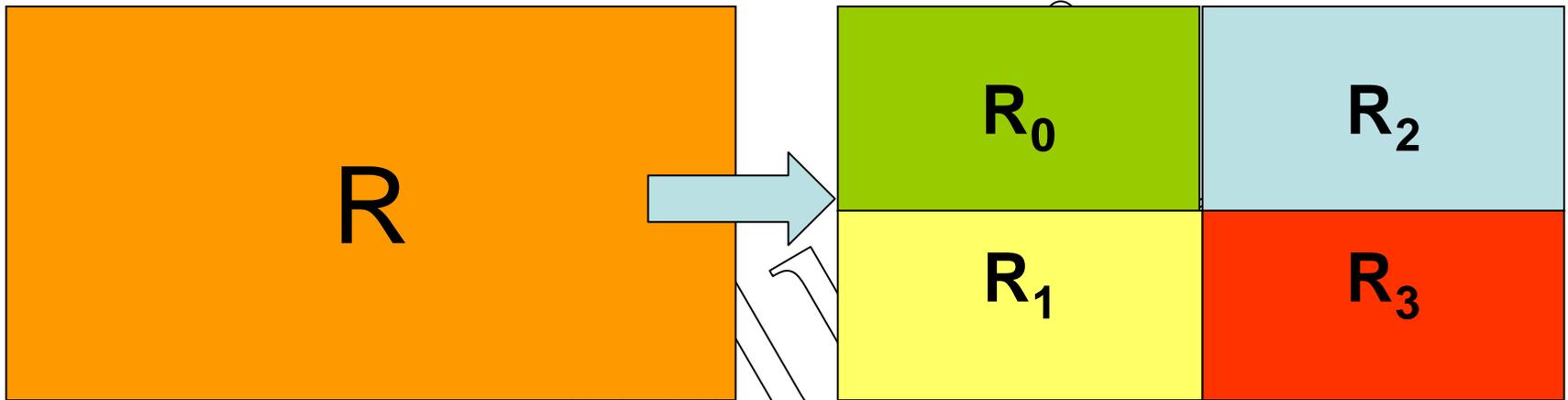
# Esempio: $n=2, p=2$

Suddividiamo il dominio di integrazione  $R$  tra i processori



# In generale: $n=2, p=4$

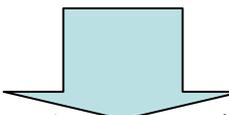
Suddividiamo il dominio di integrazione  $R$  tra i processori



$$\int_R f(t_1, t_2) dt_1 dt_2 = \int_{R_0} f(t_1, t_2) dt_1 dt_2 + \dots + \int_{R_3} f(t_1, t_2) dt_1 dt_2$$

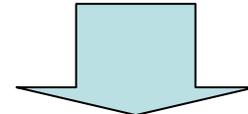
# Algoritmo parallelo (I strategia)

decomposizione (statica) del dominio di integrazione e sua distribuzione tra i processori



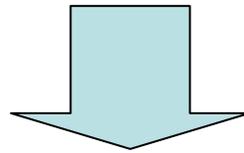
ogni processore calcola l'integrale sul proprio sottodominio

(eventualmente utilizzando su ciascuno un metodo diverso)



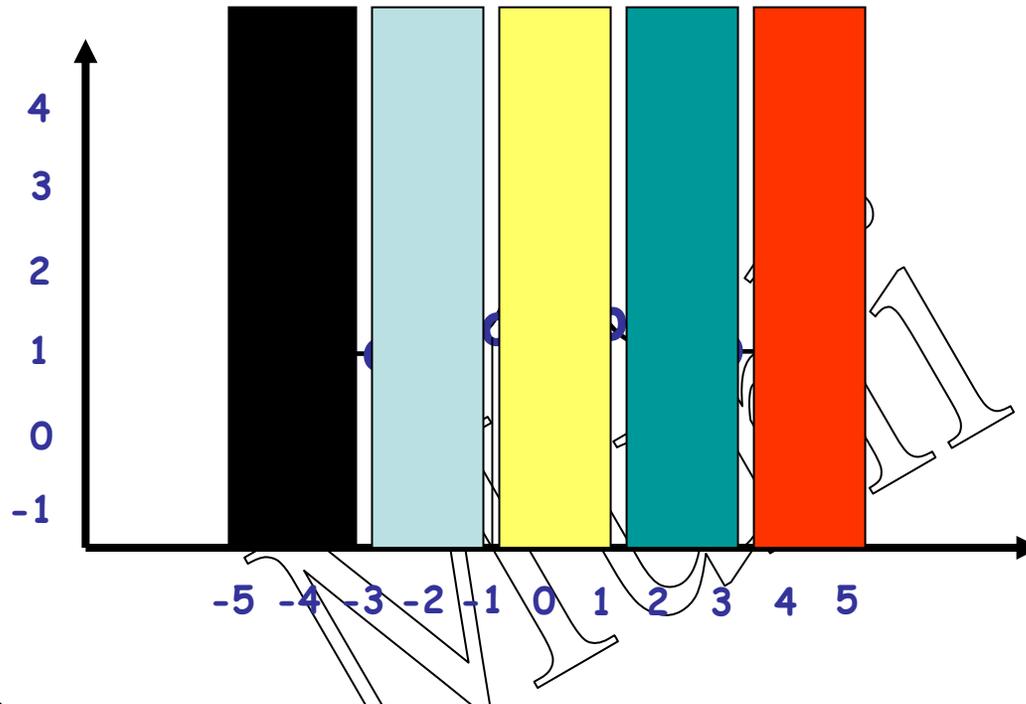
**Introduzione del parallelismo al livello del problema  
(coarse grained)**

# Algoritmo parallelo (I strategia)



l'algoritmo parallelo viene ricondotto al calcolo della **somma di n addendi** (....e quindi con una scarsa efficienza)

# UN PROBLEMA IN PIU' .....



la funzione integranda può avere un comportamento diverso in uno o più sottodomini per cui il

**carico di lavoro non è equamente bilanciato tra i processori**

# BILANCIAMENTO DEL CARICO

Nel progettare un algoritmo parallelo per il calcolo di un integrale

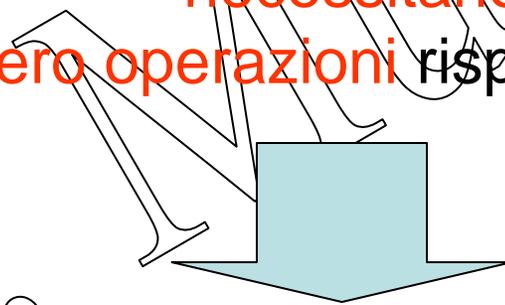
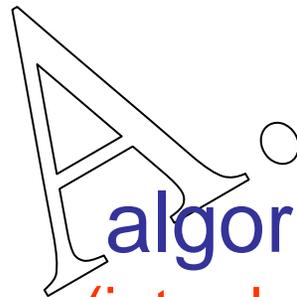
l'obiettivo principale è

distribuire tra i processori il calcolo  
relativo ai sottodomini che

*per il raggiungimento della tolleranza richiesta*

necessitano di

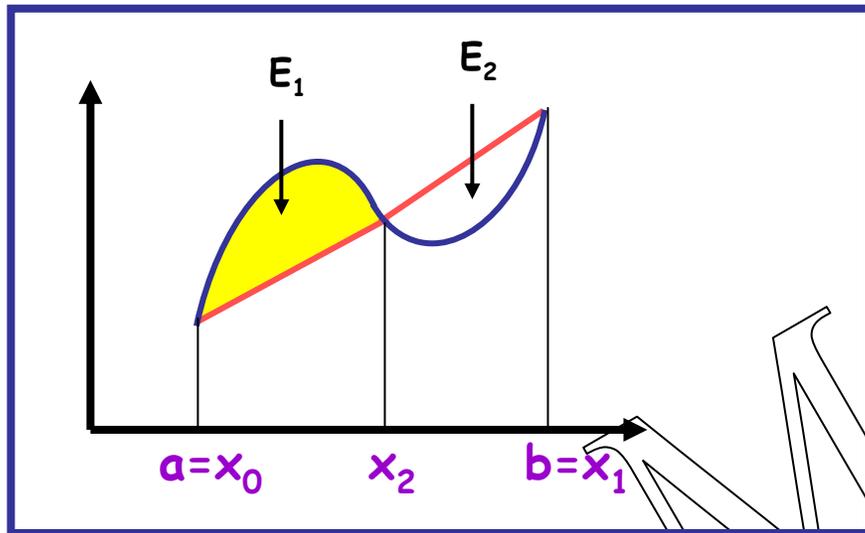
un maggior numero operazioni rispetto agli altri sottodomini



**algoritmi adattativi paralleli**

(introduzione del parallelismo al livello  
dell'algoritmo / metodo numerico)

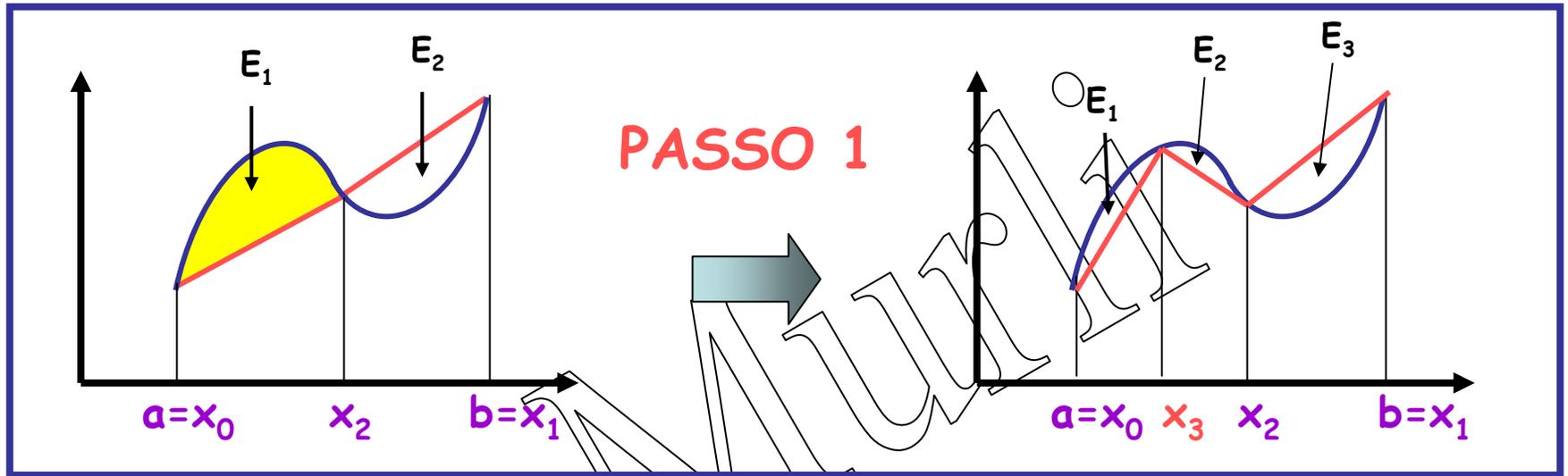
# Algoritmi adattativi (str. globale)



**PASSO 0: decomposizione del dominio**

suddividiamo il dominio di integrazione in sottodomini e assegniamo ciascun sottodominio ad un processore

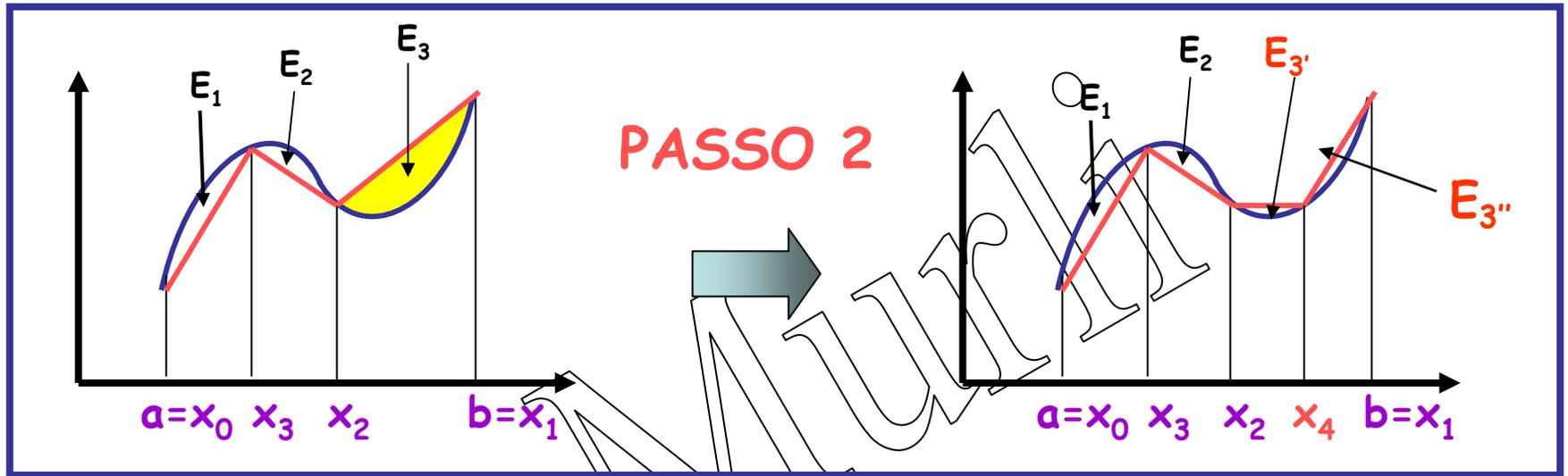
# Strategia adattativa (globale)



**PASSO 1: calcolo del massimo**

si cerca tra tutti i sottodomini quello che ha il massimo errore

# Strategia adattativa (globale)



## PASSO 2: distribuzione del carico

il sottodominio  $E_3$  (poi  $E'_3$ , poi  $E''_3 \dots$ ) viene suddiviso in sottodomini *ciascuno assegnato ad un processore*

# Algoritmo parallelo (II strategia)

decomposizione *dinamica*  
del dominio di integrazione



**MIGLIORE BILANCIAMENTO DEL CARICO**

# Algoritmo parallelo (II strategia)

ad ogni passo i processori devono comunicare per

- 1) scambiarsi le informazioni sul massimo errore  
(NON E' NOTO A PRIORI L'ESITO DI OGNI PASSO)
- 2) suddividersi il carico relativo al sottodominio più "difficile"

# PROBLEMA

Individuare un criterio che,  
pur conservando il bilanciamento dinamico  
del carico di lavoro tra processori,  
consenta di evitare (o ridurre) le  
comunicazioni collettive tra i processori.

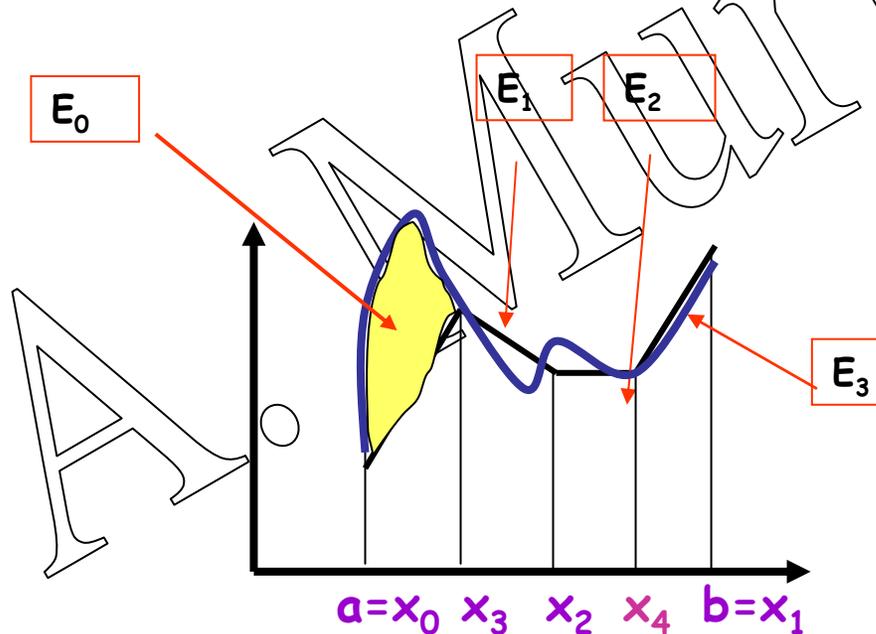
A.

# ALGORITMO PARALLELO (II STRATEGIA)

## ESEMPIO: $P=4$

Supponiamo che il dominio di integrazione sia suddiviso inizialmente tra i processori .

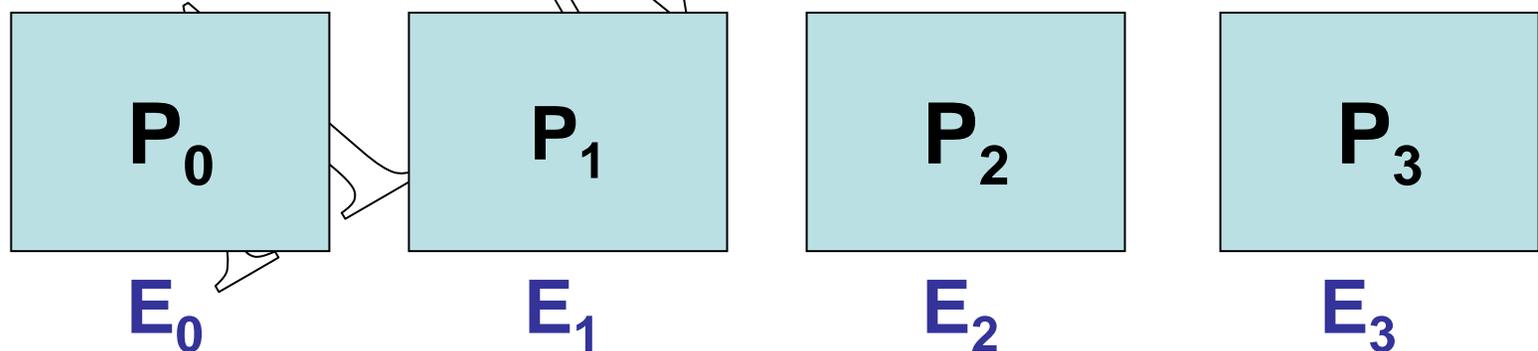
Ogni processore elabora il proprio sottodominio



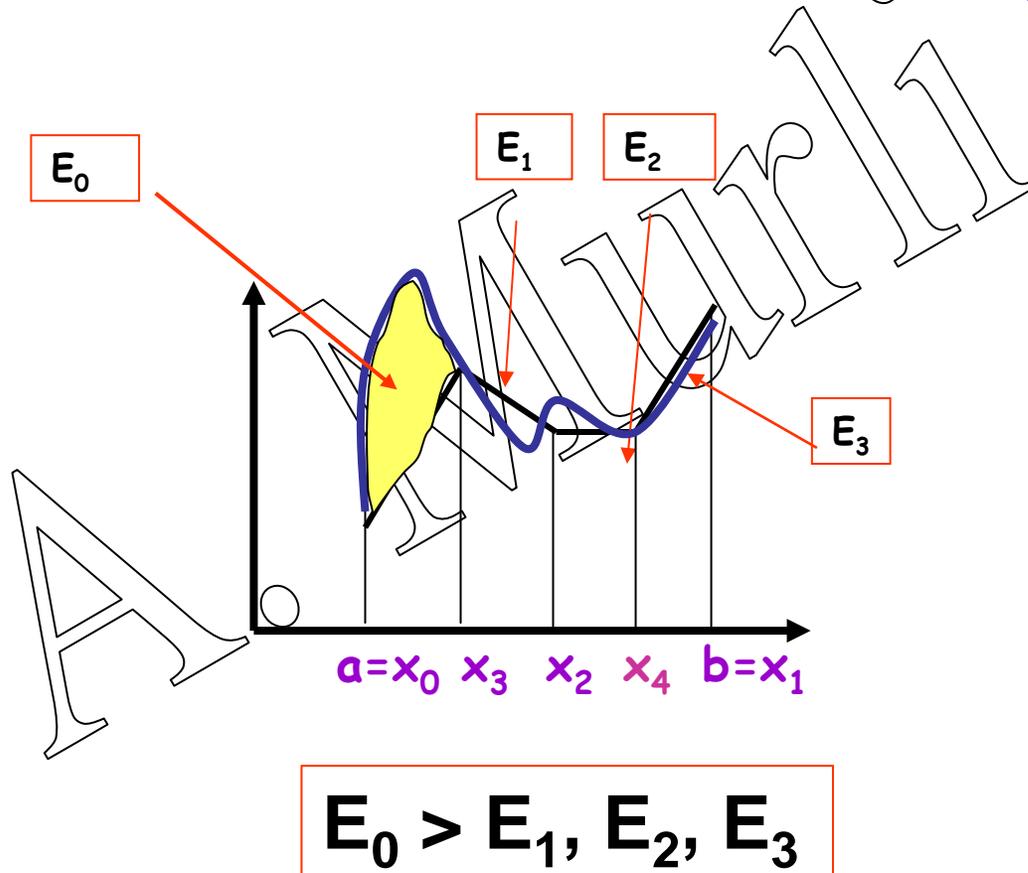
# Algoritmo parallelo

AL generico passo  $k$ ,

- ciascun processore suddivide il proprio dominio in un certo numero di parti (ad es. 2)
- ciascun processore calcola il **massimo errore** su ciascun sottodominio in proprio possesso

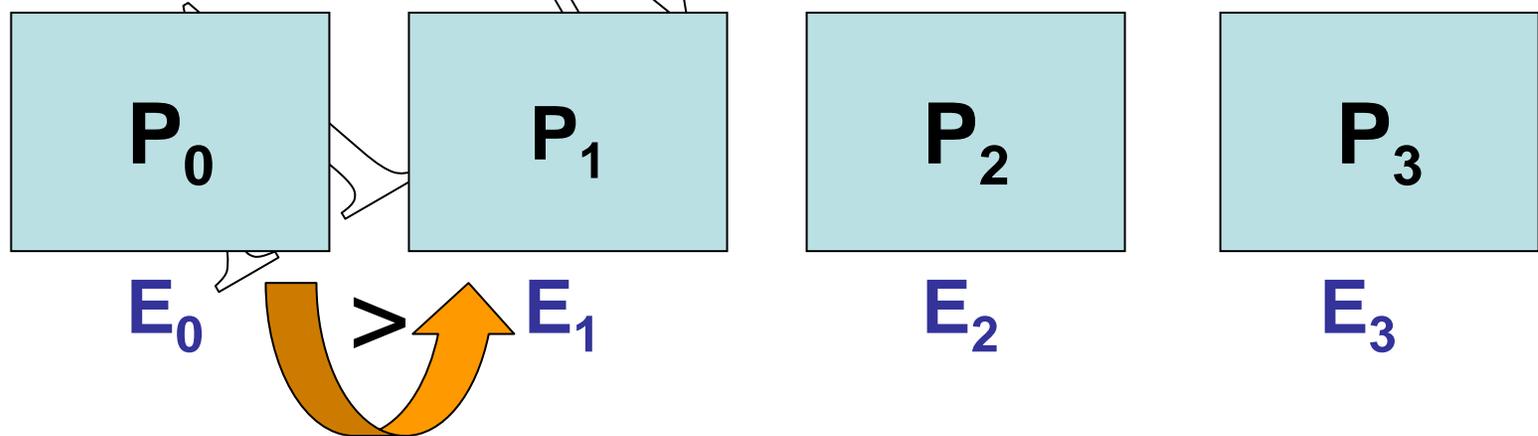


Supponiamo che ci sia un solo sottodominio  
“più difficile da integrare” rispetto agli altri e che sia il primo  
a sinistra (quello con errore  $E_0$ )



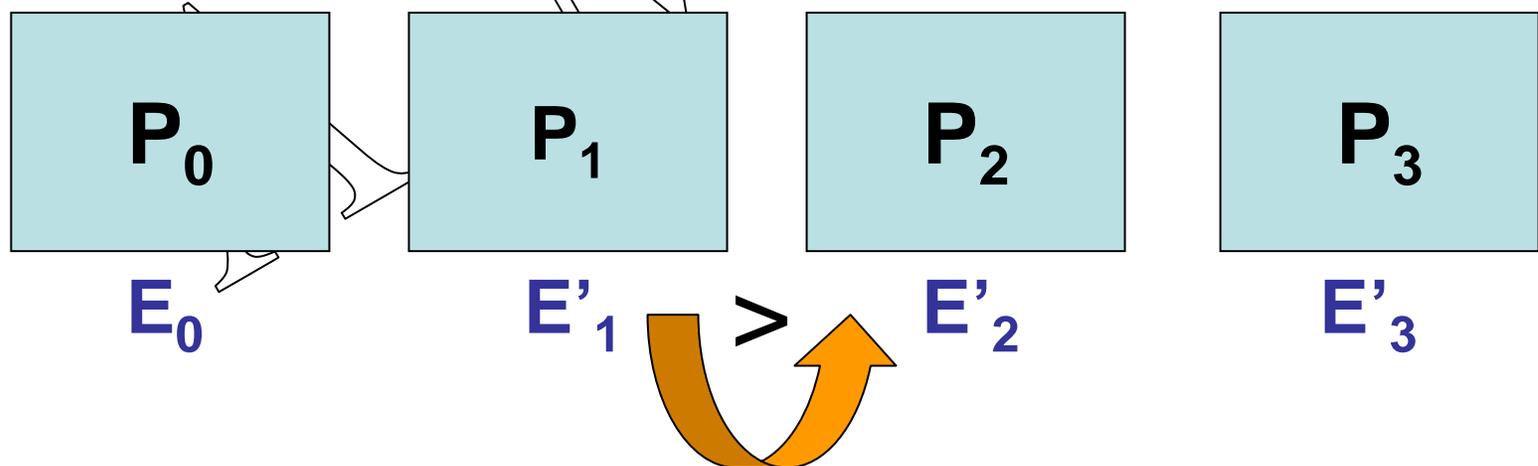
# Algoritmo parallelo, $n=1$ , $p=4$

- il processore che ha il sottodominio il cui massimo errore tra i propri sottodomini risulta essere anche più grande del massimo errore del processore **che si trova alla sua destra** lo invia solo a tale processore



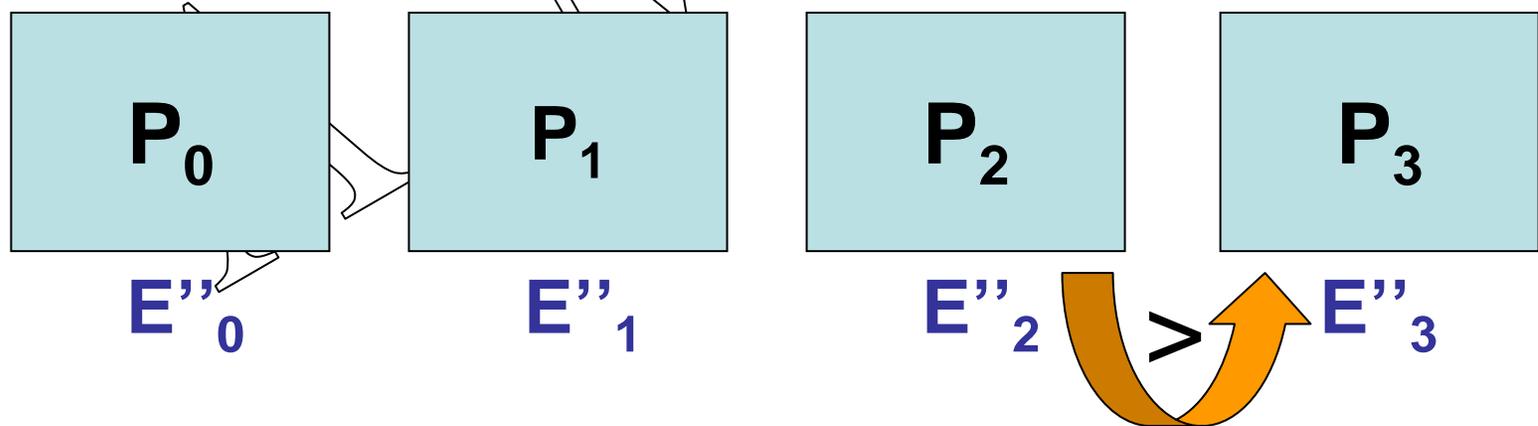
# Algoritmo parallelo, $n=1$ , $p=4$

- Supponiamo che il sottodominio che inizialmente era in  $P_0$ , anche dopo le successive suddivisioni locali, attraversi tutti i processori perché risulta avere sempre errore massimo



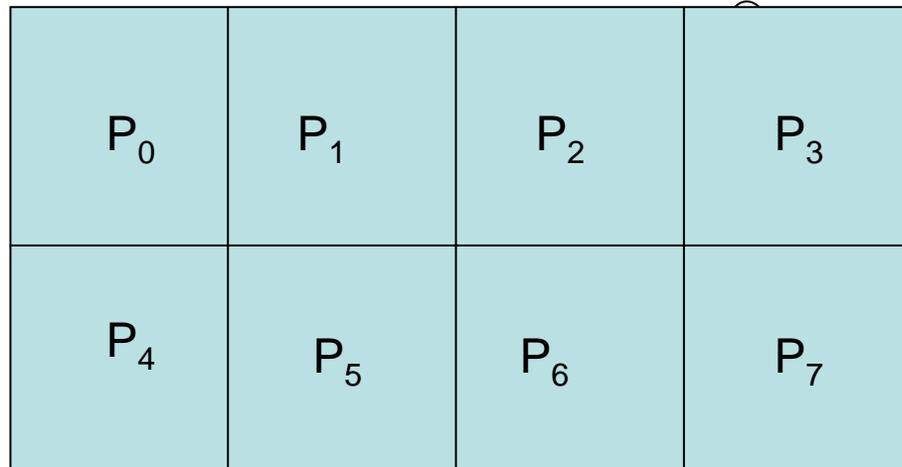
# Algoritmo parallelo, $n=1$ , $p=4$

- In questo modo il carico di lavoro relativo al sottointervallo che richiede più operazioni viene **condiviso gradualmente** con gli altri processori



# ESEMPIO $n=2, p=8$

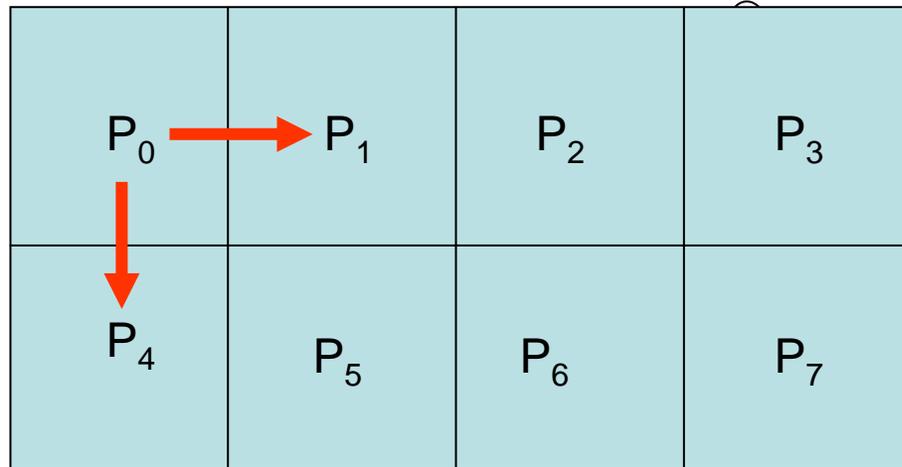
Il dominio di integrazione  $R$  viene suddiviso tra gli 8 processori



- Al generico passo  $k$ ,
  - Ciascun processore  $P_i$  ha calcolato il massimo errore  $E_i$  sulle parti in cui è stato diviso ciascun sottodominio  $S_i$

# ESEMPIO $n=2, p=8$

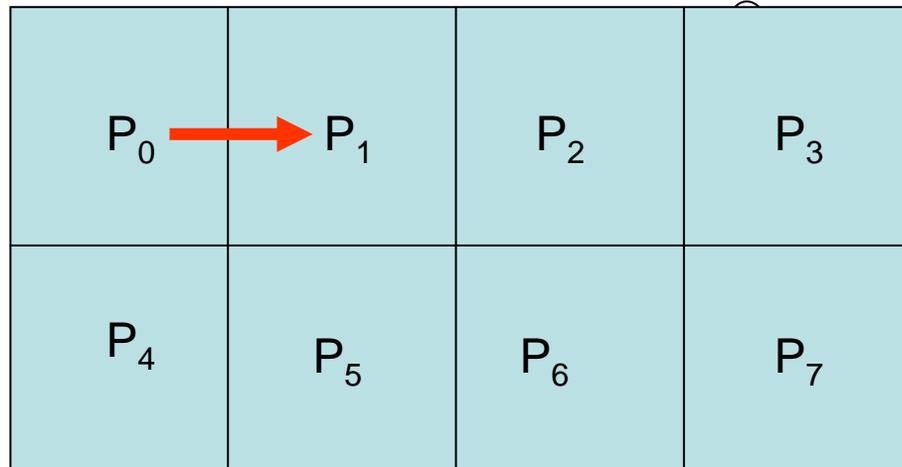
Il dominio di integrazione  $R$  viene suddiviso tra gli 8 processori



- Al generico passo  $k$ ,
  - Il sottodominio con errore massimo viene spedito al processore alla destra o in basso alternando la direzione orizzontale e quella verticale

# ESEMPIO $n=2, p=8$

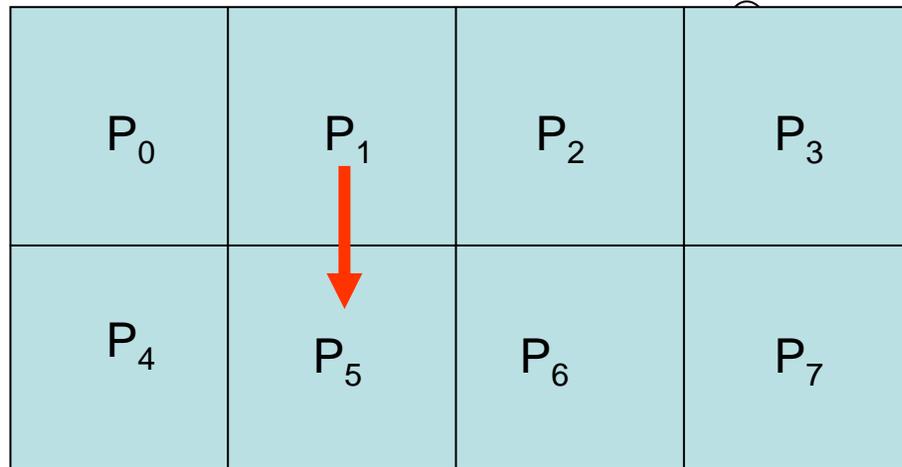
Supponiamo che il sottodominio con il massimo errore sia in  $P_0$



- Così facendo ...

- Il sottodominio con errore massimo attraversa tutti i processori della griglia

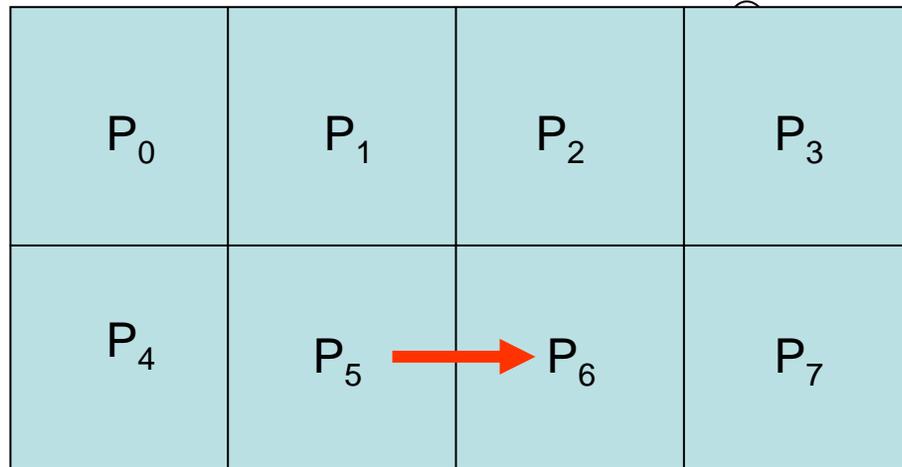
# ESEMPIO $n=2, p=8$



- Così facendo ...

- Il sottodominio con errore massimo attraversa tutti i processori della griglia

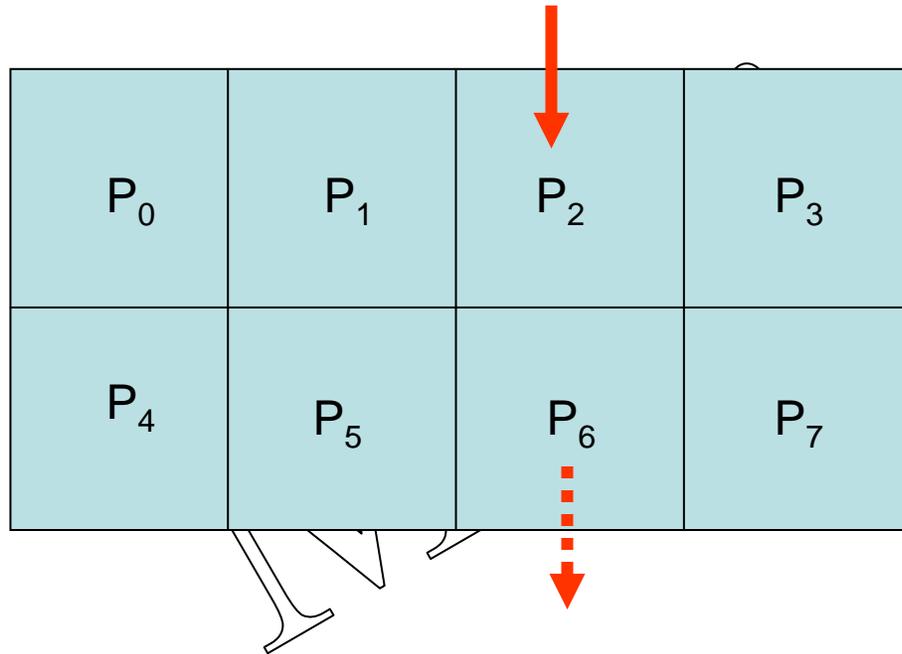
# ESEMPIO $n=2, p=8$



- Così facendo . . .

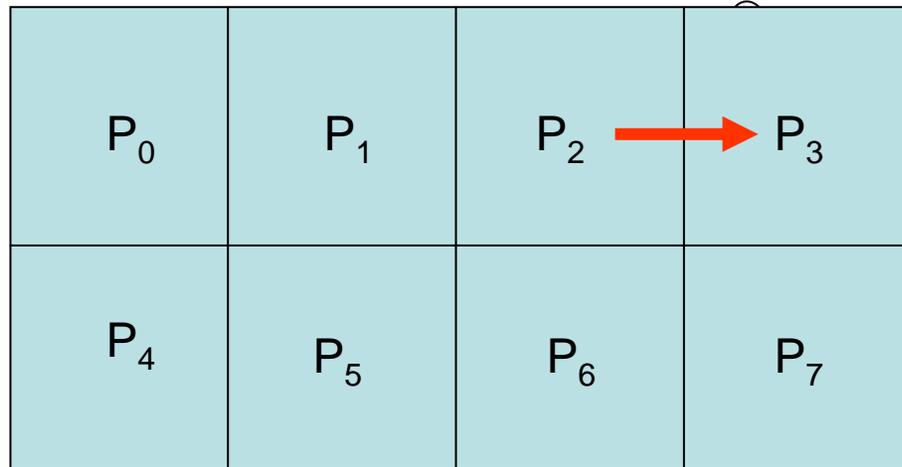
- Il sottodominio con errore massimo attraversa tutti i processori della griglia

# ESEMPIO $n=2, p=8$



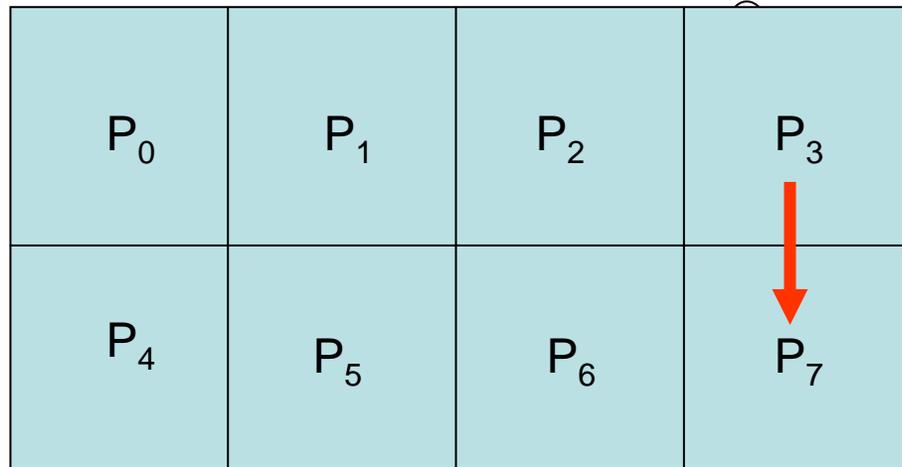
- Così facendo ...
  - Il sottodominio con errore massimo attraversa tutti i processori della griglia

# ESEMPIO $n=2, p=8$



- Così facendo ...
  - Il sottodominio con errore massimo attraversa tutti i processori della griglia

# ESEMPIO $n=2, p=8$



- Così facendo . . .

- Il sottodominio con errore massimo **attraversa tutti i processori della griglia**

# In generale

- Un buon bilanciamento dinamico del carico è necessario quando non è noto a priori l'esito (l'evoluzione) dell'algoritmo
- Un buon bilanciamento dinamico del carico richiede necessariamente lo scambio di informazioni tra i processori
- l'obiettivo è evitare o ridurre comunicazioni globali che richiedono una forte sincronizzazione tra tutti i processori utilizzando al loro posto comunicazioni locali tra coppie di processori perché comportano meno overhead di sincronizzazione