Calcolo Speed Up ed Efficienza
con la legge di Ware-Amdahl

Prodotto Matrice-Vettore

Strategia I - Quanti calcoli?

$$Ax = b \quad A \in \Re^{nxm} \quad x \in \Re^{m} \quad b \in \Re^{n}$$

- Distribuzione della matrice per righe: ad ogni processore va un insieme di righe intere
- Il vettore X viene dato a tutti i processori
- Ogni processore calcola un prodotto matrice-vettore più piccolo

$$A_i x = b_i \quad A_i \in \Re^{rxm} \quad x \in \Re^{m} \quad b_i \in \Re^{r}$$

- Non si deve ricompilare il risultato

Strategia I - Quanti calcoli?

Dati : p processi, **n (≥ p),** dati del processore i:

$$r_i = \begin{cases} \dfrac{n}{p} & se \quad i \geq n\%p \\ \dfrac{n}{p}+1 & se \quad i < n\%p \end{cases}$$

Algoritmo sequenziale

```
for i=0,n-1 do
        y_i =0
   for j=0,n-1 do
            y_i=y_i+a_ij x_j
   endfor
endfor
```

$$\#op\_prod\_locale = r_i \cdot (2m-1)$$

Strategia I - Quanti calcoli?

Dati : p processi, **n (≥ p)**

$$r_i = \begin{cases} \dfrac{n}{p} & se \quad i \geq n\%p \\ \dfrac{n}{p}+1 & se \quad i < n\%p \end{cases}$$

$$\#op\_totali = \sum_{i=0}^{p-1} r_i \cdot (2m-1)$$

Algoritmo sequenziale

```
for i=0,n-1 do
        y_i =0
   for j=0,n-1 do
            y_i=y_i+a_ij x_j
   endfor
endfor
```

Strategia I – Legge di Ware-Amdahl

Per la legge di Ware-Amdahl

$$S(p) = \frac{1}{\alpha + \frac{1-\alpha}{p}}$$

α parte sequenziale, ovvero frazione di operazioni eseguite da un solo processore

α-1 parte parallela, ovvero frazione di operazioni eseguite contemporaneamente da più di un processore

- Nel nostro caso:

$$\alpha = \frac{0}{\sum_{i=0}^{p-1} r_i \cdot (2m-1)} \qquad 1-\alpha = 1 - \frac{0}{\sum_{i=0}^{p-1} r_i \cdot (2m-1)} = \frac{\sum_{i=0}^{p-1} r_i \cdot (2m-1)}{\sum_{i=0}^{p-1} r_i \cdot (2m-1)} = 1$$

Strategia I – Legge di Ware-Amdahl general.

Per la legge di Ware-Amdahl generalizzata

$$S(p) = \frac{1}{\alpha_1 + \sum_{k=2}^{p} \frac{\alpha_k}{k}}$$

α_k frazione di operazioni eseguite da k processori

- Nel nostro caso:

$$\alpha_1 = \frac{0}{\sum_{i=0}^{p-1} r_i \cdot (2m-1)} \qquad \alpha_p = \frac{\sum_{i=0}^{p-1} r_i \cdot (2m-1)}{\sum_{i=0}^{p-1} r_i \cdot (2m-1)}$$

Tutti gli altri sono nulli!

Strategia II - Quanti calcoli?

$$Ax = b \quad A \in \Re^{nxm} \quad x \in \Re^{m} \quad b \in \Re^{n}$$

- Distribuzione della matrice per colonne: ad ogni processore va un insieme di colonne intere
- Il vettore X viene distribuito tra tutti i processori
- Ogni processore calcola un prodotto matrice-vettore più piccolo

$$A_i x_i = s_i \quad A_i \in \Re^{nxc} \quad x_i \in \Re^{c} \quad s_i \in \Re^{n}$$

- Si deve ricompilare il risultato

$$b = \sum_i s_i$$

Strategia II - Quanti calcoli?

Dati : q processi, **m (≥ q)**, colonne del processore i:

$$c_i = \begin{cases} \dfrac{m}{q} & se \quad i \geq m\%q \\ \dfrac{m}{q}+1 & se \quad i < m\%q \end{cases}$$

$$\#op\_prod\_locale = n \cdot (2c_i - 1)$$

Algoritmo a blocchi

```
begin
    y=0
    for i=0 to p-1 do
            r_i=A_i x_i
            y=y+r_i
    endfor
end
```

Strategia II - Quanti calcoli?

Dati : q processi, **m (≥ q)**, colonne del processore i:

$$c_i = \begin{cases} \dfrac{m}{q} & se \quad i \geq m\%q \\ \dfrac{m}{q}+1 & se \quad i < m\%q \end{cases}$$

Algoritmo a blocchi

```
begin
    y=0
    for i=0 to p-1 do
            r_i=A_i x_i
            y=y+r_i
    endfor
end
```

$$\#op\_ricompilazione = n \cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}$$

Processori attivi al passo i

Strategia II - Quanti calcoli?

Dati : q processi, **m (≥ q)**, colonne del processore i:

$$c_i = \begin{cases} \dfrac{m}{q} & se \quad i \geq m\%q \\ \dfrac{m}{q}+1 & se \quad i < m\%q \end{cases}$$

Algoritmo a blocchi

```
begin
    y=0
    for i=0 to p-1 do
            r_i=A_i x_i
            y=y+r_i
    endfor
end
```

$$\#op\_ricompilazione = n \cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}$$

Ognuno compie n somme

Strategia II - Quanti calcoli?

Dati : q processi, **m (≥ q)**, colonne del processore i:

$$c_i = \begin{cases} \dfrac{m}{q} & se \quad i \geq m\%q \\ \dfrac{m}{q}+1 & se \quad i < m\%q \end{cases}$$

Algoritmo a blocchi

```
begin
    y=0
    for i=0 to p-1 do
            r_i=A_i x_i
            y=y+r_i
    endfor
end
```

$$\#op\_totali = \sum_{i=0}^{q-1} n \cdot (2c_i - 1) + n \cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}$$

Strategia II – Legge di Ware-Amdahl

Per la legge di Ware-Amdahl

$$S(q) = \frac{1}{\alpha + \frac{1-\alpha}{q}}$$

α parte sequenziale, ovvero frazione di operazioni eseguite da un solo processore

α-1 parte parallela, ovvero frazione di operazioni eseguite contemporaneamente da più di un processore

- Nel nostro caso:

$$\alpha = \frac{n}{\sum_{i=0}^{q-1} n\cdot(2c_i - 1) + n\cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}}$$

$$1-\alpha = 1 - \frac{n}{\sum_{i=0}^{q-1} n\cdot(2c_i - 1) + n\cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}} = \frac{\sum_{i=0}^{q-1} n\cdot(2c_i - 1) + n\cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i} - n}{\sum_{i=0}^{q-1} n\cdot(2c_i - 1) + n\cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}}$$

Strategia II – Legge di Ware-Amdahl generalizzata

Per la legge di Ware-Amdahl generalizzata

$$S(q) = \frac{1}{\alpha_1 + \sum_{k=2}^{q} \frac{\alpha_k}{k}}$$

α_k frazione di operazioni eseguite da k processori

Tutti gli altri sono nulli!

- Nel nostro caso:

$$\alpha_1 = \frac{n}{\sum_{i=0}^{q-1} n \cdot (2c_i - 1) + n \cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}}$$

$$\alpha_{2^i} = \frac{\frac{q}{2^i} \cdot n}{\sum_{i=0}^{q-1} n \cdot (2c_i - 1) + n \cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}} \quad \forall i \in [1, (\log_2 q) - 1]$$

$$\alpha_p = \frac{\sum_{i=0}^{q-1} n \cdot (2c_i - 1)}{\sum_{i=0}^{q-1} n \cdot (2c_i - 1) + n \cdot \sum_{i=1}^{\log_2 q} \frac{q}{2^i}}$$