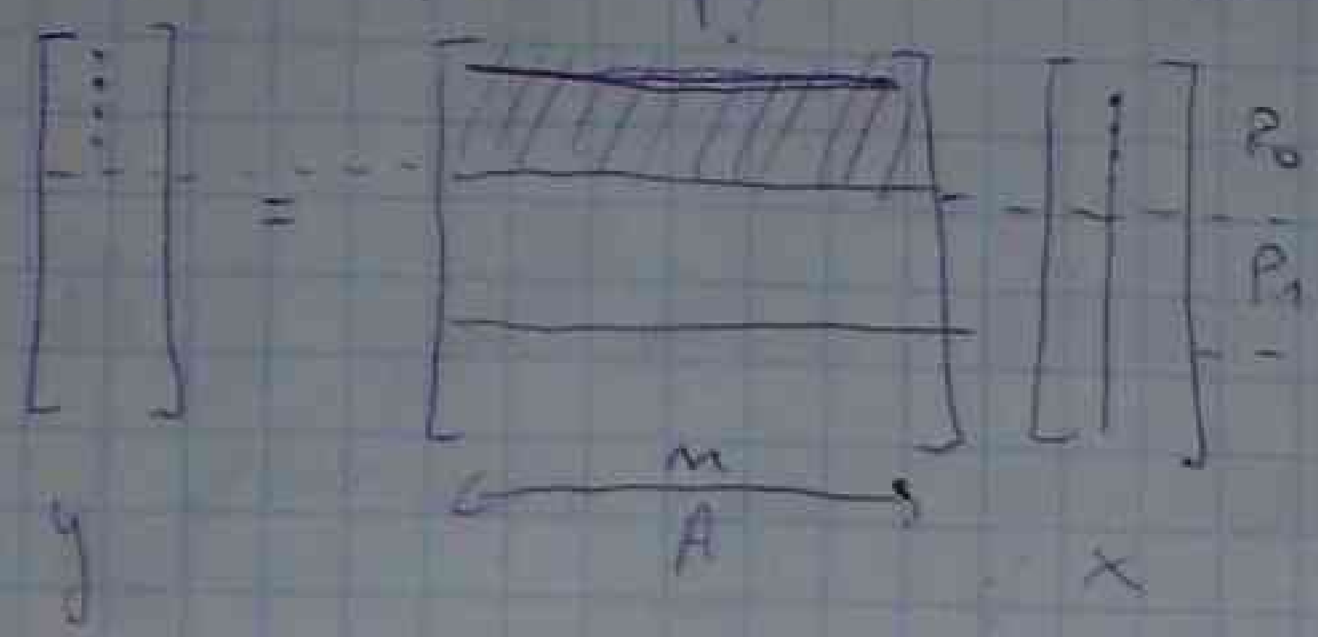
~~ćwiczenia 8 stycznia~~

Wykład 15 stycznia

Mnożenie macierz - wektor równoległe

1. Wstępem jest podział macierzy



podział wierszy (procesor dostaje blok wierszy)

Algorytm

Procesor brakuje reszty wektora x — musi go otrzymać (zebrać) od wszystkich innych procesorów części wektora x. Następny proc jest w identycznej sytuacji. Zbieranie wszyscy do wszystkich i rozgłaszanie wszyscy do wszystkich

1) Allgather - $\frac{m}{p}$ (wymiar macierzy / liczba procesorów) wyrazów w komunikacie

2) Obliczenia $2\frac{m^2}{p}$ operacji na procesach

$T_{||}(1) = 2m^2 \cdot t_0 = W = \boxed{\mathcal{L} p W_0}$ t_0 - średni czas potrzebny na wykonanie operacji

$T_{||}(p) = \frac{2m^2}{p} \cdot t_0 + t_s \log p + \frac{m}{p} \cdot t_w (p-1)$

(torus 2D: $2t_s(\sqrt{p}-1)$)

t_w - czas przesłania liczby

przyspieszenie $S = \frac{2m^2 t_0}{\frac{2m^2 t_0}{p} + t_s \log p + \frac{m(p-1)t_w}{p}} \xrightarrow{p\to\infty} 0$

efektywność $E(p) \xrightarrow{p\to\infty} 0$

! $2m^2 t_0 = \mathcal{L} \cdot p \cdot W_0 \Rightarrow m \sim \sqrt{p}$

$$S^s = \frac{c p W_0}{\frac{c p W_0}{p} + t_s \log p + \sqrt{\frac{c p W_0}{2 t_0}} \cdot \frac{(p-1)}{p} t_w} =$$

skalowane przyspieszenie

$$= \frac{1}{\frac{1}{p} + \frac{t_s \log p}{c p W_0} + \frac{t_w}{\sqrt{2 t_0 c p W_0}} \cdot \frac{p-1}{p}} \xrightarrow{p \to \infty} \infty$$

$$E^s(p) \xrightarrow{p \to \infty} 0$$

Nie jest to algorytm liniowo skalowalny (jeżeli praca rośnie liniowo z liczbą procesorów), efektywność zmierza do zera.

Czy istnieje funkcja $W(p)$, gdzie $\underbrace{E(p) \xrightarrow{p \to \infty} \varepsilon > 0}$ (przy p dążącym do nieskończoności efektywność jest silnie większa od zera) ???

$$T_{11}(p) = 2n^2 t_0 = cW = cp^2 W_0$$

$$S(p, W) = \frac{c p^2 W_0}{\frac{c p^2 W_0}{p} + t_s \log p + \sqrt{\frac{c W_0}{2 t_0}} \cdot p \cdot (p-1) \frac{t_w}{p}} =$$

$W = p^2 W_0$

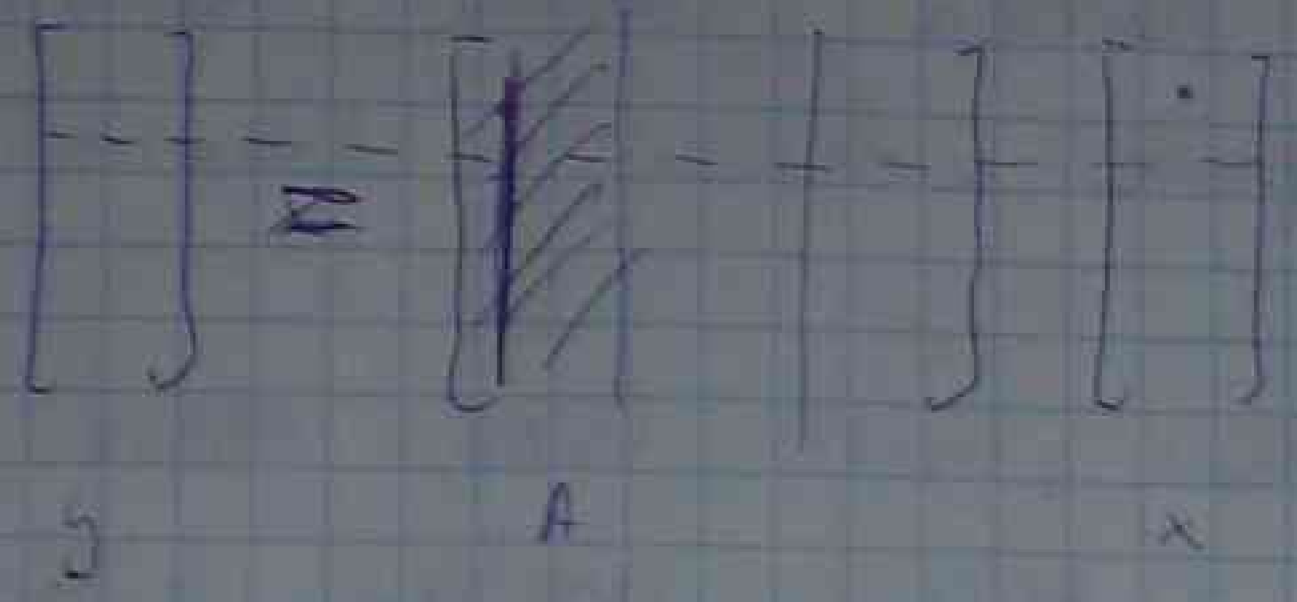
$$= \frac{1}{\frac{1}{p} + \frac{\log p}{p^2} \cdot \frac{t_s}{c W_0} + \frac{1}{\sqrt{2 t_0 c W_0}} \cdot \frac{(p-1)}{p^2} t_w}$$

$$E(p, W) = \frac{1}{1 + \frac{\log p}{p} \cdot \frac{t_s}{c W_0} + \frac{(p-1)}{p} \cdot \frac{t_w}{\sqrt{2 t_0 c W_0}}} \xrightarrow{p \to \infty} \text{const}$$

Im większe przepustowości, mniejsze opóźnienia, tym większa efektywność.

Praktycznie: przyspieszenie ograniczone przez pamięć, $p \sim n^2$

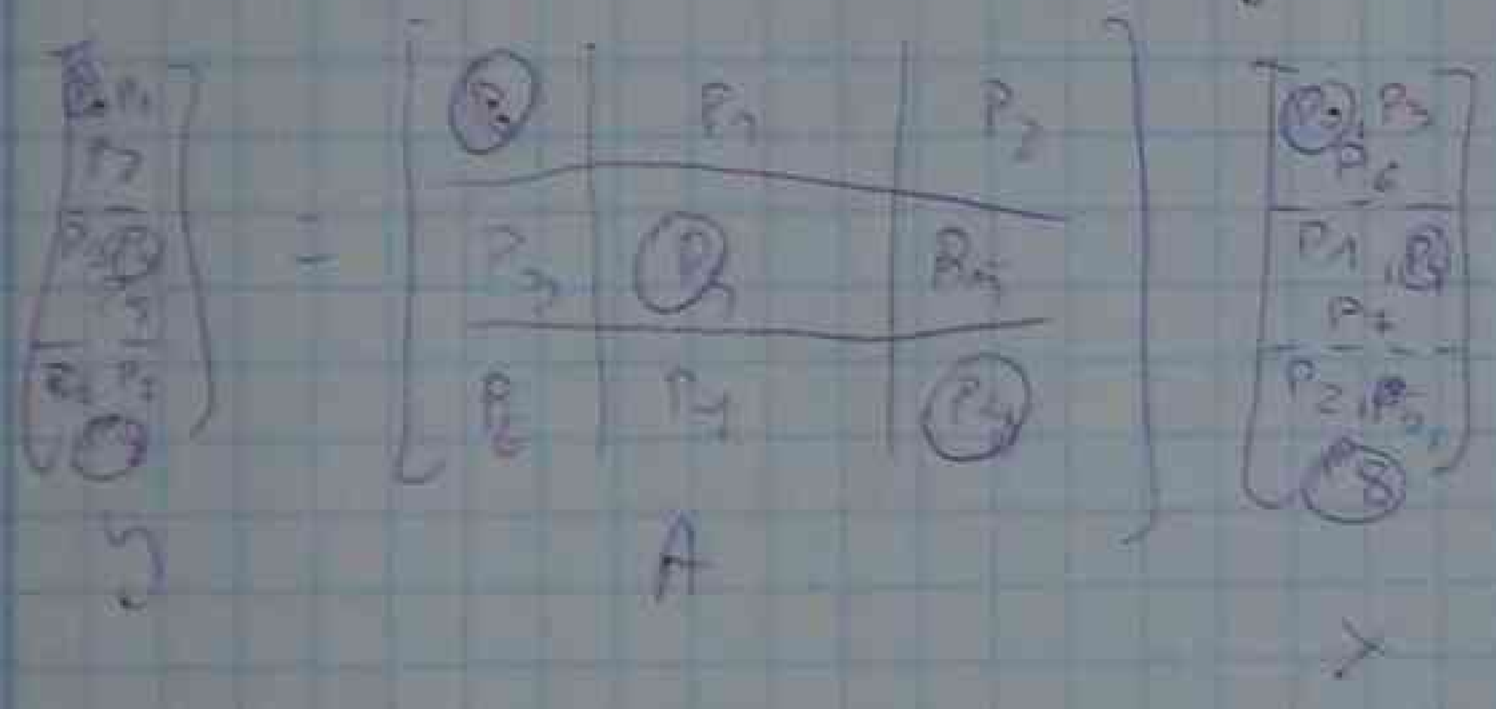
2. Podział kolumnowy, procesor dostaje grupę kolumn



Algorytm

1) obliczenia $\frac{2n^2}{p}$ operacji / procesor
2) Redukcja (sumowanie wartości y każdego procesora), $\frac{n}{p} t_w \log p + t_s \log p$

3. Podział szachownicowy



Procesy na przekątnej głównej dostają części wektora X

Algorytm

1) Każdy proc musi zebrać zbiorowo od wszystkich procesów w kolumnie, Allgather, $\frac{n}{\sqrt{p}}$ to rozmiar komunikatu

$t_s \cdot \log \sqrt{p} + \frac{n}{\sqrt{p}} \cdot t_w (\sqrt{p} - 1)$

2) obliczenia $\frac{n}{\sqrt{p}} \cdot \frac{n}{\sqrt{p}} = \frac{n^2}{p} \Rightarrow 2\frac{n^2}{p} \cdot t_c$

3. Redukcja $\frac{n}{\sqrt{p}}$ · $t_w \log\sqrt{p} + t_s \log\sqrt{p}$

Gdy macierz będzie mała, a zmiany są niewielkie. Zwiększa się liczba operacji w stosunku do komunikacji.

Gdy macierz jest pasmowa, rośnie liczba obliczeń, $\frac{np}{2m}$ to liczba sąsiadów, z którymi wymienia komunikaty. Ważnym argumentem jest szerokość pasma.