

OPERASI MATRIKS

Topik yang akan dibahas

- transpose
- perkalian

TRANSPOSE

Definisi:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}$$

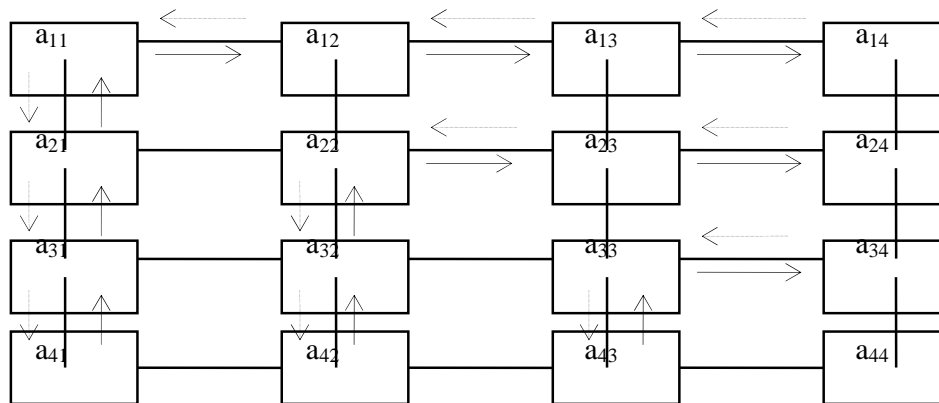
Prosedur sekuensial:

```

procedure TRANSPOSE (A)
  for i = 2 to n do
    for j = 1 to i-1 do
      aij ↔ aji
    end for
  end for.
  
```

Waktu: $O(n^2)$; jumlah langkah: $\Omega(n^2)$

MESH TRANSPOSE



Awalnya: prosesor $P(i,j)$ menyimpan elemen data a_{ij} .

Pada akhir komputasi: $P(i,j)$ menyimpan a_{ji} .

Ide algoritma:

- Elemen diagonal tetap stasioner.
- Elemen di bawah diagonal dipindah ke posisi simetris di atas diagonal (tanda panah biasa).
- Elemen di atas diagonal dipindah ke posisi simetris di bawah diagonal (tanda panah terputus-putus).
- Setiap prosesor $P(i,j)$ memiliki tiga register:
 1. $A(i,j)$ digunakan untuk menyimpan a_{ij} pd awalnya dan a_{ji} ketika algoritma berakhir.
 2. $B(i,j)$ digunakan untuk menyimpan data yg diterima dari $P(i, j+1)$ atau $P(i-1, j)$, yaitu, dari tetangga di kanan atau atasnya.
 3. $C(i,j)$ digunakan untuk menyimpan data yg diterima dari $P(i, j-1)$ atau $P(i+1, j)$, yaitu, dari tetangga di kiri atau bawahnya.

procedure MESH TRANSPOSE (A)

Step 1: do steps 1.1 and 1.2 in parallel

- (1.1) for $i=2$ to n do in parallel
for $j=1$ to $i-1$ do in parallel
 $C(i-1,j) \leftarrow (a_{ij}, j, i)$
end for
end for
- (1.2) for $i=1$ to $n-1$ do in parallel
for $j=i+1$ to n do in parallel
 $B(i,j-1) \leftarrow (a_{ij}, j, i)$
end for
end for.

Step 2: do steps 2.1, 2.2, and 1.2 in parallel

- (2.1) for $i=2$ to n do in parallel
for $j=1$ to $i-1$ do in parallel
while $P(i,j)$ receives input form its neighbours do
(i) if (a_{km}, m, k) is received from $P(i+1, j)$
then send it to $P(i-1, j)$
end if
(ii) if (a_{km}, m, k) is received from $P(i-1, j)$
then if $i=m$ and $j=k$
then $A(i,j) \leftarrow a_{km}$ { destination reached}
else send (a_{km}, m, k) to $P(i+1, j)$
end if
end if
end while
end for
end for
- (2.2) for $i=1$ to n do in parallel
while $P(i,i)$ receives input form its neighbours do
(i) if (a_{km}, m, k) is received from $P(i+1, i)$
then send it to $P(i, i+1)$
end if

```

(ii) if ( $a_{km,m,k}$ ) is received from  $P(i, i+1)$ 
    then send it to  $P(i+1, i)$ 
    end if
end while
end for

(2.3) for  $i=1$  to  $n-1$  do in parallel
    for  $j=i+1$  to  $n$  do in parallel
        while  $P(i,j)$  receives input from its neighbours do
            (i) if ( $a_{km,m,k}$ ) is received from  $P(i, j+1)$ 
                then send it to  $P(i, j-1)$ 
            end if
            (ii) if ( $a_{km,m,k}$ ) is received from  $P(i, j-1)$ 
                then if  $i=m$  and  $j=k$ 
                    then  $A(i,j) \leftarrow a_{km}$  { destination reached }
                    else send ( $a_{km,m,k}$ ) to  $P(i, j+1)$ 
                end if
            end if
        end while
    end for
end for

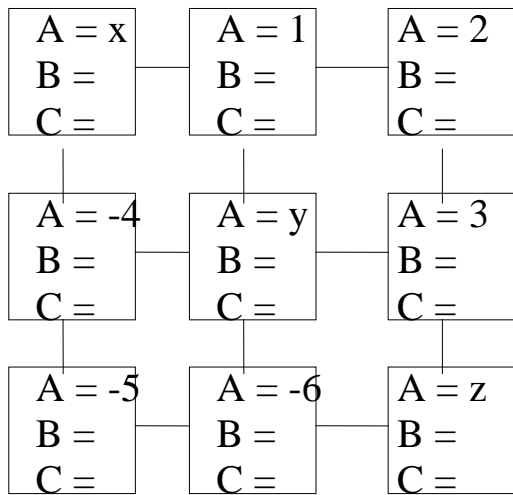
```

Analisis:

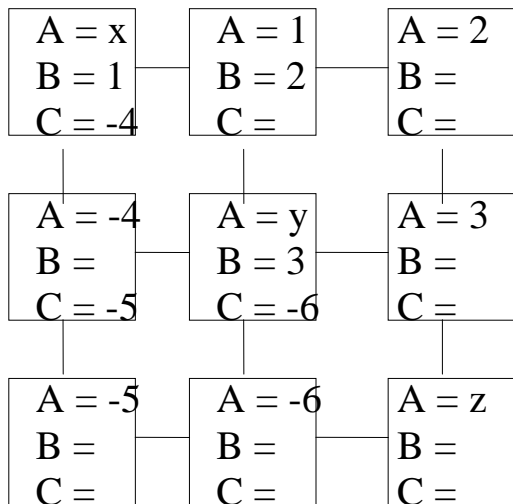
- Setiap elemen a_{ij} , $i > j$, harus menelusuri kolomnya ke atas sampai mencapai $P(j,j)$ dan kemudian berjalan sepanjang baris hingga sampai di $P(j,i)$. Hal yg sama berlaku untuk a_{ij} , $j > i$.
- Jalur terpanjang adalah yg dijalani oleh a_{n1} (atau a_{1n}) yg terdiri dari $2(n-1)$ langkah.
- Running time prosedur MESH TRANSPOSE: $t(n) = O(n)$
- $p(n) = n^2$
- Cost: $O(n^3) \rightarrow$ tidak optimal

Contoh:

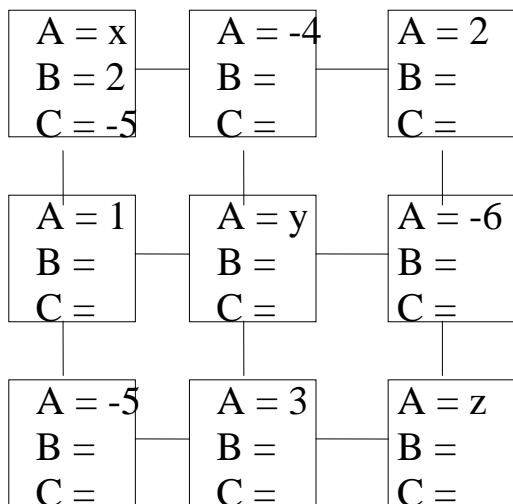
$$A = \begin{bmatrix} x & 1 & 2 \\ -4 & y & 3 \\ -5 & -6 & z \end{bmatrix}$$



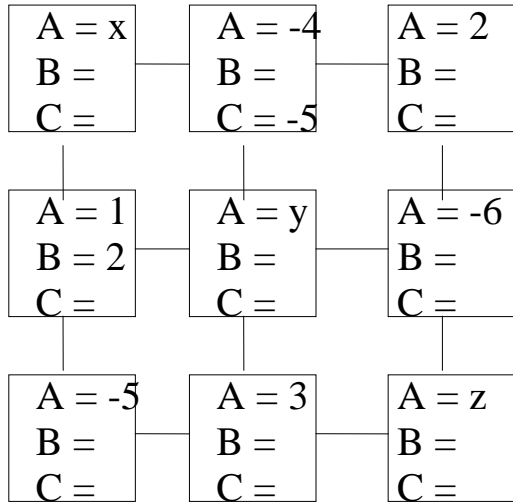
Keadaan awal



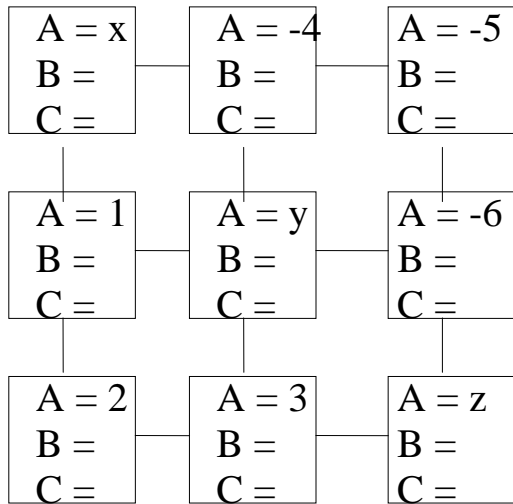
Step 1



Iterasi pertama step 2



Iterasi kedua step 2



Iterasi ketiga step 2

SHUFFLE TRANSPOSE

Pertimbangan:

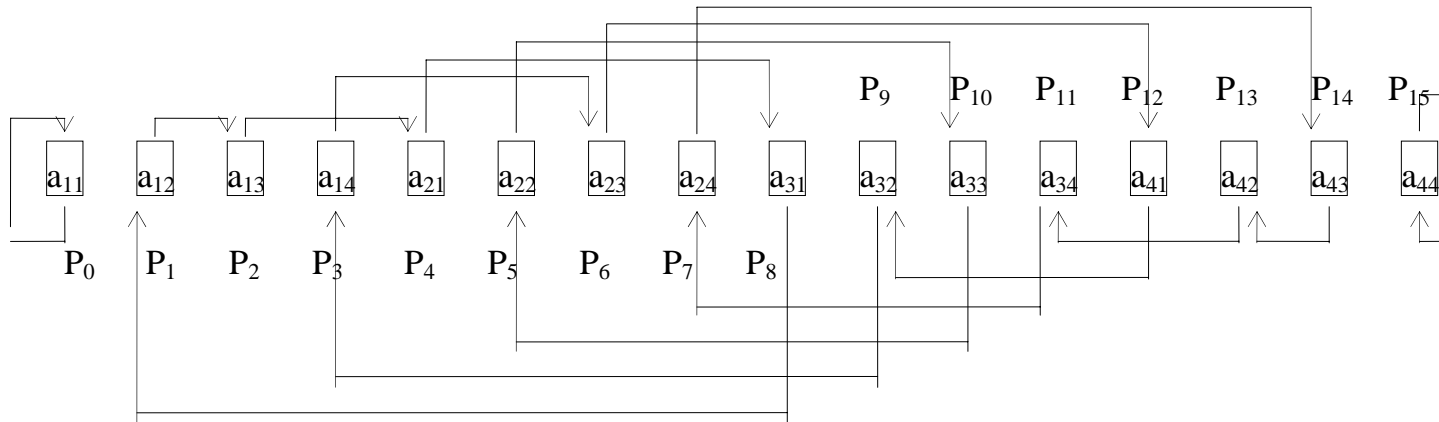
Speedup MESH TRANSPOSE hanya linier, dianggap kecil karena jumlah prosesor kuadratik.

Geometri yg berbeda dpt men-transpose matriks dalam waktu logaritmik.

$n = 2^q$; matrik $A(n \times n)$

Digunakan interkoneksi perfect shuffle dgn n^2 prosesor: P_0, P_1, P_{n-1} .

a_{ij} disimpan pada awalnya tersimpan di prosesor P_k , dengan $k = 2^q(i-1) + (j-1)$.



Setelah q operasi shuffle, prosesor P_k berisi elemen aji.

P_k dihubungkan ke P_m ; m didapat dengan menggeser bit k ke kiri siklis.

Indeks prosesor k terdiri dari $2q$ bit. q most significant bit merepresentasikan $i-1$, dan q least significant bit menyatakan $j-1$.

Setelah q shuffle (yaitu q pergeseran siklis ke kiri), elemen yang tadinya ada pada P_k akan berada di prosesor yang indeksnya adalah:

$$s = 2^q(j-1) + (i-1)$$

procedure SHUFFLE TRANSPOSE (A)

for $i=1$ to q do

for $k=1$ to $2^{2q}-2$ do in parallel

P_k sends the element of A it currently holds to $P_{2^{k \bmod (2^{2q}-1)}}$

end for

end for.

Analisis:

- Iterasi: q waktu konstan
- Waktu: $t(n) = O(\log n)$
- $p(n) = n^2$; $c(n) = O(n^2 \log n) \rightarrow$ tdk optimal
- Tetapi interkoneksi shuffle lebih cepat dari mesh.

Jalannya elemen:

a_{11} : P0 \rightarrow P0 \rightarrow P0

a_{12} : P1 \rightarrow P2 \rightarrow P4

a_{13} : P2 \rightarrow P4 \rightarrow P8

a_{14} : P3 \rightarrow P6 \rightarrow P12

a_{21} : P4 \rightarrow P8 \rightarrow P1

a_{22} : P5 \rightarrow P10 \rightarrow P5

a_{23} : P6 \rightarrow P12 \rightarrow P9

a_{24} : P7 \rightarrow P14 \rightarrow P13

a_{31} : P8 \rightarrow P1 \rightarrow P2

a_{32} : P9 \rightarrow P3 \rightarrow P6

a_{33} : P10 \rightarrow P5 \rightarrow P10

a_{34} : P11 \rightarrow P7 \rightarrow P14

a_{41} : P12 \rightarrow P9 \rightarrow P3

a_{42} : P13 \rightarrow P11 \rightarrow P7

a_{43} : P14 \rightarrow P13 \rightarrow P11

a_{44} : P15 \rightarrow P15 \rightarrow P15

EREW TRANSPOSE

\rightarrow merupakan algoritma yang cost-optimal.

- Algoritma menggunakan $(n^2-n)/2$ prosesor dan berjalan pada komputer EREW SM SIMD.
- Matriks A berada di shared memory.
- Setiap prosesor memiliki dua indeks i dan j , dengan $2 \leq i \leq n$ dan $1 \leq j \leq i-1$.
- Semua prosesor beroperasi paralel dan prosesor P_{ij} menukar dua elemen A , yaitu a_{ij} dan a_{ji} .

procedure EREW TRANSPOSE (A)

for $i=2$ to n do in parallel

for $j=1$ to $i-1$ do in parallel

$a_{ij} \leftrightarrow a_{ji}$

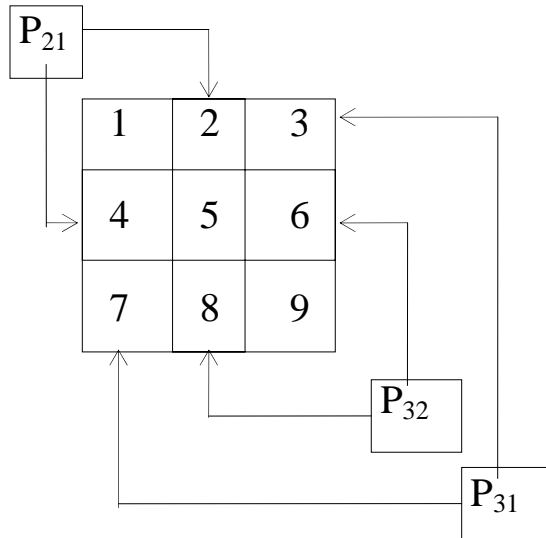
end for

end for.

Analisis:

- Running time: $t(n) = O(1)$
- $p(n) = O(n^2)$; $c(n) = O(n^2) \rightarrow$ optimal

Contoh:



PERKALIAN MATRIKS DENGAN MATRIKS

Definisi:

$$A(m \times n) \times B(n \times k) = C(m \times k)$$

$$c_{ij} = \sum_{s=1}^n a_{is} \times b_{sj}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq k$$

Algoritma sekuensial:

procedure MATRIX MULTIPLICATION (A, B, C)

 for $i=1$ to m do

 for $j=1$ to k do

 (1) $c_{ij} \leftarrow 0$

 (2) for $s=1$ to n do

$c_{ij} \leftarrow c_{ij} + (a_{is} \times b_{sj})$

 end for

 end for

 end for

Waktu: $O(n^3)$

PERKALIAN MESH

Digunakan $m \times k$ prosesor untuk mengalikan $A(m \times n)$ dengan $B(n \times k)$.

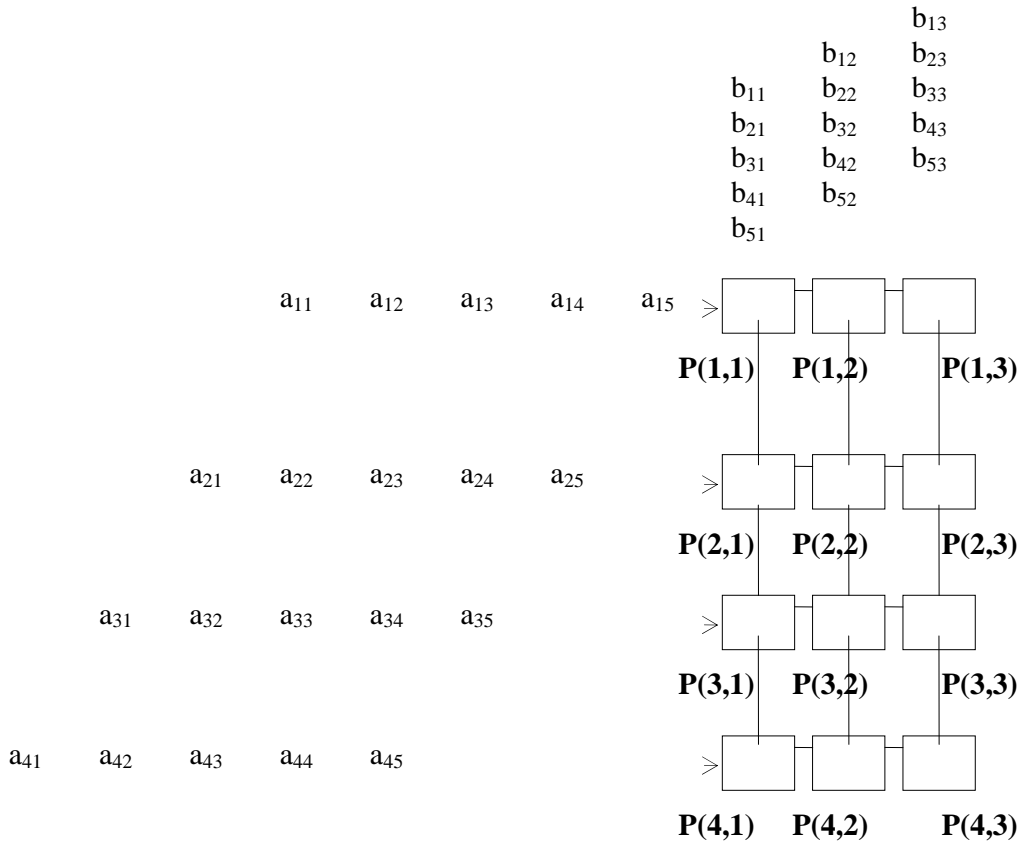
1. Matriks A dan B dimasukkan ke *boundary processor* di kolom 1 dan baris 1, berturut-turut.
2. Baris i matriks A ketinggalan satu satuan waktu dari baris $i-1$ untuk $2 \leq i \leq m$.

3. Kolom j matriks B ketinggalan satu satuan waktu dari kolom $j-1$ untuk $2 \leq j \leq k$.
4. Langkah 2 dan 3 untuk menjamin bahwa a_{is} bertemu dengan b_{sj} pada prosesor $P(i, j)$ pada waktu yang tepat.
5. Di akhir algoritma, elemen hasil perkalian, c_{ij} , berada di prosesor $P(i, j)$.

Pada awalnya, c_{ij} sama dengan nol.

Ketika $P(i, j)$ menerima dua input a dan b , prosesor:

- (i) mengalikan keduanya,
- (ii) menambahkan hasilnya ke c_{ij} ,
- (iii) mengirimkan a ke $P(i, j+1)$ kecuali $j = k$, dan
- (iv) mengirim b ke $P(i+1, j)$ kecuali $i=m$.



procedure MESH MATRIX MULTIPLICATION (A, B, C)

```

for  $i=1$  to  $m$  do in parallel
  for  $j=1$  to  $k$  do in parallel
    (1)  $c_{ij} \leftarrow 0$ 
    (2) while  $P(i, j)$  receives two inputs  $a$  and  $b$  do
      (i)  $c_{ij} \leftarrow c_{ij} + (a \times b)$ 
      (ii) if  $i < m$  then send  $b$  to  $P(i+1, j)$ 
            end if
      (iii) if  $j < k$  then send  $a$  to  $P(i, j+1)$ 
            end if
    end while
  end for
end for

```

Analisis:

- Dengan menganggap $m \leq n$ dan $k \leq n$, run time: $t(n) = O(n)$.
- $p(n) = O(n^2)$, $c(n) = O(n^3) \rightarrow$ sama dengan prosedur sekuensial.

PERKALIAN MATRIKS SHUFFLE-EXCHANGE

Jika tersedia $n^3 = 2^{3q}$ prosesor pada model SIMD shuffle-exchange, dua matriks $n \times n$ dapat dikalikan dalam waktu $\Theta(\log n)$.