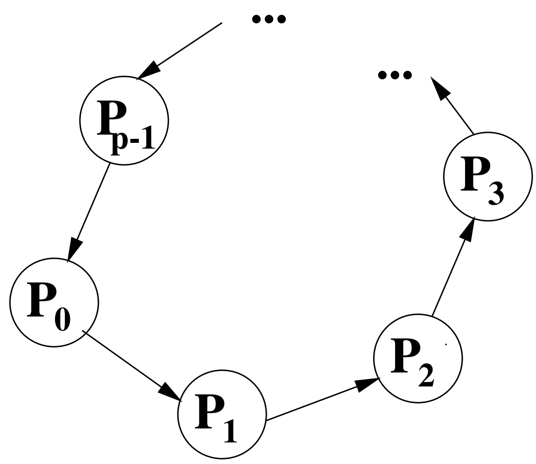
Algorithmique sur anneau de processeurs

Plan

1. Macro-communications sur un anneau
2. Produit matrice-vecteur
3. Algorithmes de balayage d'image

1 Macro-communications sur un anneau

Architecture



– p processeurs arrangés en un anneau orienté, numérotés de 0 à $p-1$
– Numéro logique : q $\leftarrow$ my_num()
– Nombre total de processeurs : tot_proc_num().

Fonctionnement SPMD
– Tous les processeurs exécutent le même programme
– Ils opèrent sur des données présentes dans leur mémoire locale
– Envoyer un message à son successeur :

send(adr, L)

adr adresse mémoire du processeur émetteur de la première valeur
L est la longueur du message
– Recevoir un message : receive(adr, L)

– Il doit y avoir un receive "en face de" chaque send
– Adresses locales :

```
q ← mynum();
if (q=0) then send(adr1, L);
if (q=1) then receive(adr2, L);
```

Sémantique

– Hypothèse restrictive : **send** et **receive** bloquants
– Hypothèse classique : **receive** bloquant, **send** non bloquant
– Hypothèse récente : les deux primitives sont non bloquantes
  $\rightarrow$ envoyer, recevoir et calculer simultanément, si les données ne sont pas partagées en écriture
  $\rightarrow$ trois *threads* de contrôle, une pour le calcul, et deux dédiées aux communications asynchrones

Coût d'une communication

– Difficile à modéliser
– Formule simple : envoyer/recevoir un message de longueur L coûte $\beta + L\tau$
– β coût d'initialisation (configuration du matériel, mise en oeuvre logicielle)
– τ inverse de la bande passante, mesure la vitesse du lien en régime permanent
– Distance de communication : message de P_0 à P_q $\rightarrow$ coût $q(\beta+L\tau)$

Diffusion

– Envoi par P_k d'un message de longueur L (stocké à l'adresse adr) à tous les autres processeurs

– Programme du processeur numéro q :

```
broadcast(k, adr, L) {
q ← my_num();
p ← tot_proc_num();
if q = k then
   send(adr, L);
else
   if q = k − 1 mod p then
     receive(adr, L);
   else
     receive(adr, L);
     send(adr, L);
}
```

– receive bloquant : on transmet le message dès réception

– temps de la diffusion d'un message de longueur L : $(p-1)(\beta+L\tau)$.

Diffusion personnalisée

– Envoi par P_k d'un message différent à tous les autres processeurs
– Message pour P_q à l'adresse adr[q]
– A la fin, chaque processeur a son message à l'adresse adr
– Pipeline des messages sur l'anneau
– Programme du processeur numéro q :

```
scatter(k, adr, L) {
q ← my_num();
p ← tot_proc_num();
if q = k then
   adr ← adr[k];
else
   receive(adr, L);
   for i = 1 to k − 1 − q mod p
     {{ send(adr, L) // receive(temp, L) }};
     adr ← temp;
}
```

– send non bloquant
– receive bloquant (sinon ajouter des points de synchronisation)
– temps total comme pour la diffusion, $(p-1)(\beta + L\tau)$.

2 Produit matrice-vecteur

Données

– Multiplication $y = Ax$
– Matrice A de dimension $n \times n$
– Vecteur x à n composantes (numérotées de 0 à $n-1$)
– Anneau orienté de p processeurs, $r = n/p$

Distribution

– Ensemble de n produits scalaires :

```
for i = 1 to n do
  for j = 1 to n do
    y[i] = y[i] + a[i,j] * x[j];
```

– Distribuer le calcul des produits scalaires aux processeurs.
– Chaque processeur a en mémoire r lignes de la matrice A, rangées dans une matrice a de dimension $r \times n$
– P_q contient les lignes qr à $(q+1)r-1$ de la matrice, et les composantes de même rang des vecteurs x et y :

```
var a : array[0..r-1,0..n-1] of real;
var x, y : array[0..r-1] of real;
```

Principe

$$
\begin{array}{ccc}
P_0 & \begin{pmatrix} A_{00} & A_{01} & A_{02} & A_{03} & A_{04} & A_{05} & A_{06} & A_{07} \\ A_{10} & A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} \end{pmatrix} & \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \\
\hline
P_1 & \begin{pmatrix} A_{20} & A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & A_{27} \\ A_{30} & A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} & A_{37} \end{pmatrix} & \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \\
\hline
P_2 & \begin{pmatrix} A_{40} & A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} & A_{47} \\ A_{50} & A_{51} & A_{52} & A_{53} & A_{54} & A_{55} & A_{56} & A_{57} \end{pmatrix} & \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} \\
\hline
P_3 & \begin{pmatrix} A_{60} & A_{61} & A_{62} & A_{63} & A_{64} & A_{65} & A_{66} & A_{67} \\ A_{70} & A_{71} & A_{72} & A_{73} & A_{74} & A_{75} & A_{76} & A_{77} \end{pmatrix} & \begin{pmatrix} x_6 \\ x_7 \end{pmatrix}
\end{array}
$$

(a) Distribution initiale des données

$$
\begin{array}{ccccc}
P_0 & \begin{pmatrix} A_{00} & A_{01} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ A_{10} & A_{11} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} & \text{temp} \leftarrow & \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} \\
\hline
P_1 & \begin{pmatrix} \bullet & \bullet & A_{22} & A_{23} & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & A_{32} & A_{33} & \bullet & \bullet & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} & \text{temp} \leftarrow & \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \\
\hline
P_2 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & A_{44} & A_{45} & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & A_{54} & A_{55} & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} & \text{temp} \leftarrow & \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \\
\hline
P_3 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{66} & A_{67} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{76} & A_{77} \end{pmatrix} & \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} & \text{temp} \leftarrow & \begin{pmatrix} x_4 \\ x_5 \end{pmatrix}
\end{array}
$$

(b) Première étape

$$
\begin{array}{lccc}
P_0 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{06} & A_{07} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{16} & A_{17} \end{pmatrix} & \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} \\
\hline
P_1 & \begin{pmatrix} A_{20} & A_{21} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ A_{30} & A_{31} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} \\
\hline
P_2 & \begin{pmatrix} \bullet & \bullet & A_{42} & A_{43} & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & A_{52} & A_{53} & \bullet & \bullet & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \\
\hline
P_3 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & A_{64} & A_{65} & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & A_{74} & A_{75} & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_2 \\ x_3 \end{pmatrix}
\end{array}
$$

(c) Deuxième étape

$$
\begin{array}{lccc}
P_0 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & A_{04} & A_{05} & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & A_{14} & A_{15} & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \\
\hline
P_1 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{26} & A_{27} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{36} & A_{37} \end{pmatrix} & \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} \\
\hline
P_2 & \begin{pmatrix} A_{40} & A_{41} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ A_{50} & A_{51} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} \\
\hline
P_3 & \begin{pmatrix} \bullet & \bullet & A_{62} & A_{63} & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & A_{72} & A_{73} & \bullet & \bullet & \bullet & \bullet \end{pmatrix} & \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} & \text{temp} \leftarrow \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}
\end{array}
$$

(c) Troisième étape

$$
\begin{array}{l|l}
P_0 & \begin{pmatrix} \bullet & \bullet & A_{02} & A_{03} & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & A_{12} & A_{13} & \bullet & \bullet & \bullet & \bullet \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \quad \text{temp} \leftarrow \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \\
\hline
P_1 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & A_{24} & A_{25} & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & A_{34} & A_{35} & \bullet & \bullet \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} \quad \text{temp} \leftarrow \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \\
\hline
P_2 & \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{46} & A_{47} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & A_{56} & A_{57} \end{pmatrix} \begin{pmatrix} x_6 \\ x_7 \end{pmatrix} \quad \text{temp} \leftarrow \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} \\
\hline
P_3 & \begin{pmatrix} A_{60} & A_{61} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ A_{70} & A_{71} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \quad \text{temp} \leftarrow \begin{pmatrix} x_6 \\ x_7 \end{pmatrix}
\end{array}
$$

(c) Quatrième étape

Algorithme

```
matrice-vecteur(A, x, y) {
q ← my_num();
p ← tot_proc_num();
for step = 0 to p − 1 do
   {{ send(x, r) // receive(temp, r) //
     for i=0 to r-1 do
       for j=0 to r-1 do
         y[i] ← y[i]+ a[i,(q - step mod p)r+j]*x[j] }};
   x ← temp;
}
```

- Temps d'exécution $T(p) = p \max\{r^2\tau_a, \beta + r\tau_c)$
- Efficacité égale à 1 si n assez grand ($\frac{n^2}{p^2}\tau_a \geq \beta + \frac{n}{p}\tau_c$).

3 Algorithmes de balayage d'image

Algorithme séquentiel

– P une grille de $n \times n$ points

– Voisinage :

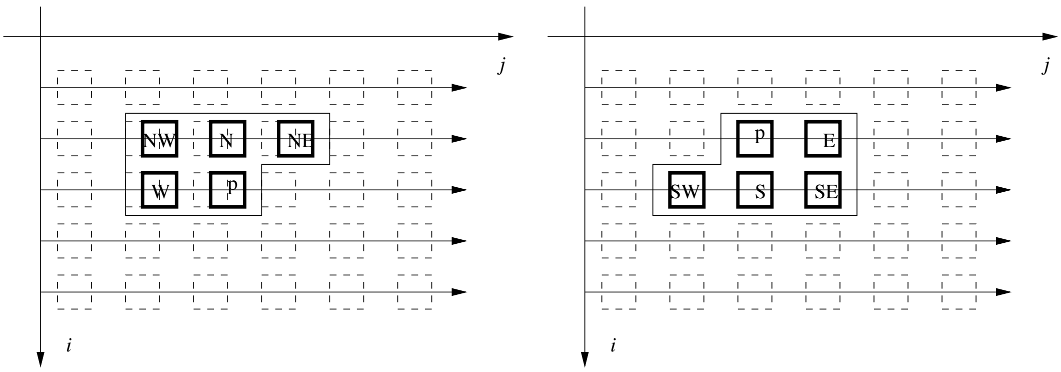
$$\begin{array}{ccc} NW & N & NE \\ W & \boxed{\mathrm{p}} & E \\ SW & S & SE \end{array}$$

– Balayage en passe avant : de haut en bas et de gauche à droite :

(PAV) p := mise_à_jour_avant(p, W, NW, N, NE)

– Balayage en passe arrière : de bas en haut et de droite à gauche :

(PAR) p := mise_à_jour_arrière(p, E, SE, S, SW)



(a) balayage avant **(b) balayage arrière**

Distance au contour

– P image binaire : figure $F = \{+\infty\}$, entièrement entourée par son complémentaire $P \setminus F = \{0\}$
– Calcul de la distance (CD) au contour : image copie de F
chaque pixel $\rightarrow$ sa distance (selon une certaine métrique) à $P \setminus F$
– Axe médian = lieu des centres des disques maximaux

– Balayer l'image deux fois :
(PAV) $\quad p := \min(p, W + t_1, NW + t_2, N + t_1, NE + t_2)$
(PAR) $\quad p := \min(p, E + t_1, SE + t_2, S + t_1, SW + t_2)$
– Coefficients t_1 et t_2 :
$t_1 = 1$ et $t_2 = +\infty \rightarrow$ distance de Manhattan d_4
$t_1 = t_2 = 1 \rightarrow$ distance de l'échiquier d_8
$t_1 = 3$ et $t_2 = 4 \rightarrow$ approximation de la distance Euclidienne.

Calcul de la trajectoire optimale

– Voir polycopié

Parallélisation de PAV : Algorithme glouton

– Cas $p = n$: au top $2i + j$, P_i reçoit le pixel $(i-1, j+1)$ de P_{i-1} calcule (i, j) et le transmet à P_{i+1}

$$\begin{array}{l}
P_0\ \boxed{0}\ \boxed{1}\ \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{8}\ \boxed{9}\ \cdots \\
P_1\ \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{8}\ \boxed{9}\ \cdots \\
P_2\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{8}\ \boxed{9}\ \cdots \\
P_3\ \boxed{6}\ \boxed{7}\ \boxed{8}\ \boxed{9}\ \cdots \\
P_4\ \boxed{8}\ \boxed{9}\ \cdots \\
\cdots\ \cdots
\end{array}$$

– Cas $p < n$: allocation entrelacée
la ligne j est allouée au processeur j modulo p
P_0 a besoin de valeurs calculées par P_{p-1}
P_0 reçoit $(p-1, 0)$ au temps $2p - 1$ et $(p-1, 1)$ au temps $2p$
$\rightarrow n \geq 2p$, sinon P_0 est inactif

– Point fort : le délai entre deux processeurs voisins est petit

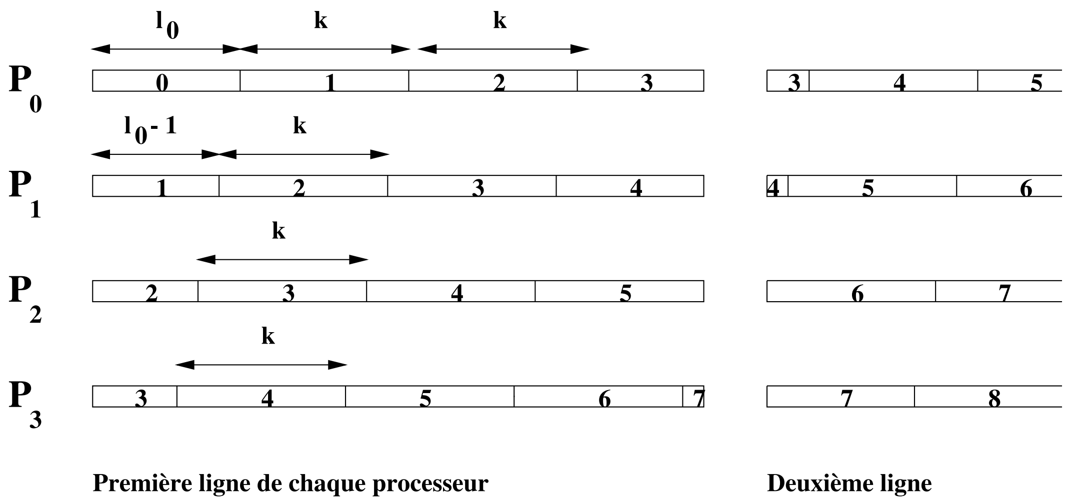
– Point faible : nombreuses communications de petite taille

Augmenter la granularité

– Utiliser de plus longs messages

– Calcul de k pixels consécutifs à chaque étape

– Condition pour le régime permanent : $n \geq (k+1)p$



– Point fort : communications moins coûteuses

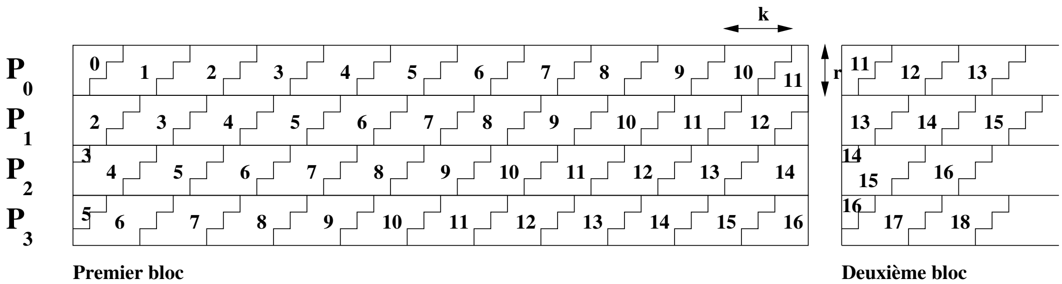
– Point faible : délai d'attente entre deux processeurs adjacents

D'autres stratégies d'allocation

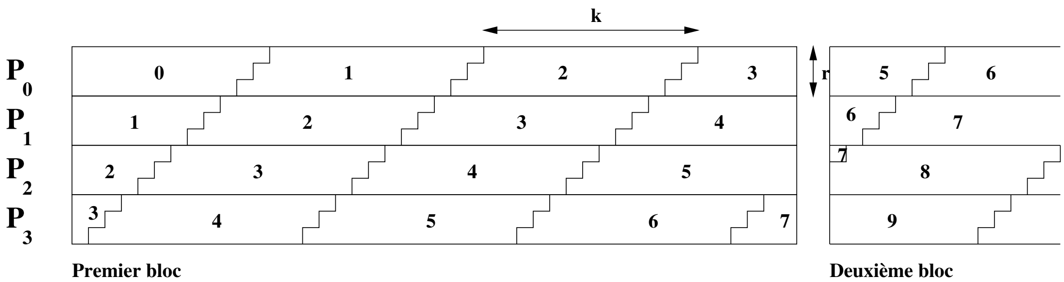
– Allouer des blocs de r lignes consécutives de manière entrelacée :

P_0	P_1	P_2	P_3
$0, 1, 2$	$3, 4, 5$	$6, 7, 8$	$9, 10, 11$
$12, 13, 14$	$15, 16, 17$	$18, 19, 20$	$21, 22, 23$
$24, 25, 26$	$27, 28, 29$	$30, 31, 32$	$33, 34, 35$
$36, 37, 38$	$39, 40, 41$	$42, 43, 44$	

– Condition pour le régime permanent : $n \geq (k + r)p$

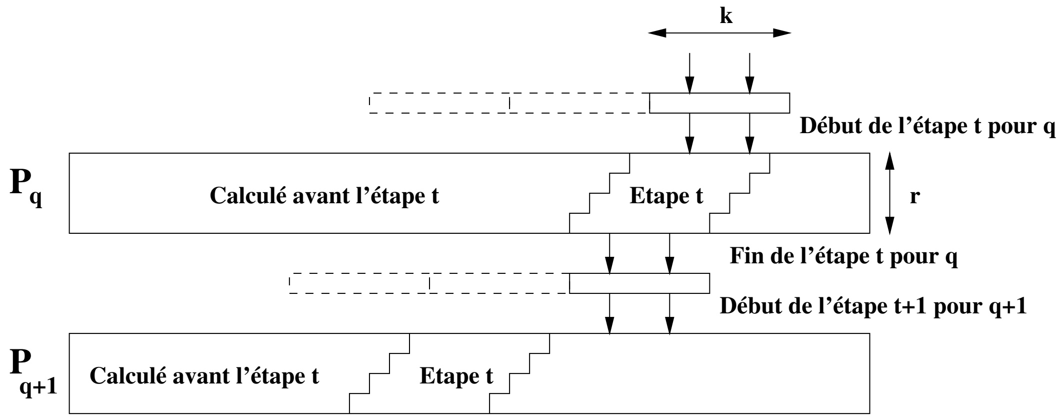


$n = 44$, $p = 4$, $r = 3$ et $k = 4$



$n = 44$, $p = 4$, $r = 3$ et $k = 13$

Analyse de complexité



- Temps séquentiel $T_{seq} = n^2 \tau_a$
- Temps d'exécution d'une étape : $\tau_{step} = \beta + k\tau_c + rk\tau_a$
- Calculer t_q, début du calcul de P_q ?
  1. P_0 calcule l_0 pixels à $t_0 = 0$
  2. P_1 calcule $l_1 = (l_0 - r)\text{mod}k$ pixels à $t_1 = 1 + \lfloor \frac{r-l_0}{k} \rfloor$
  3. P_q calcule $l_q = (l_0 - q * r)\text{mod}k$ pixels à $t_q = q + \lfloor \frac{qr-l_0}{k} \rfloor$
- Nombre total d'étapes $T_p = t_{p-1} + \lfloor (n^2/(p * r) - l_q + r - 1)/k \rfloor$
- Temps d'exécution de l'algorithme $T_{//} = \tau_{step} T_p$.
- Condition de validité ?
  1. P_0 effectue sa première réception au temps t_p
  2. A cet instant, il a déjà calculé $l_0 + k(t_p - 1)$ pixels
  3. Condition : $n - (l_0 + k(t_p - 1)) + l_p \geq k$
  4. On retrouve : $n \geq p(r + k)$

Optimisation

– Problème de taille n
– Anneau de p processeurs
– Allocation des lignes par blocs de taille r, avec $1 \leq r \leq \frac{n}{p}$
– Segments de taille k, avec $1 \leq k \leq \frac{n}{p} - r$

$$T_{//} = (\beta + k\tau_c + rk\tau_a)\left((p-1)(1+\frac{r}{k}) + \frac{n^2}{prk}\right).$$

– $k_{opt}(r) = \min(k_{max}(r), k_{//}(r))$
– $k_{max}(r) = \frac{n}{p} - r$
– $k//(r) = \sqrt{\frac{\beta}{\tau_c + r\tau_a}\left(\frac{n^2}{p(p-1)r} + r\right)}.$