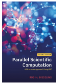
Parallel Fast Fourier Transform

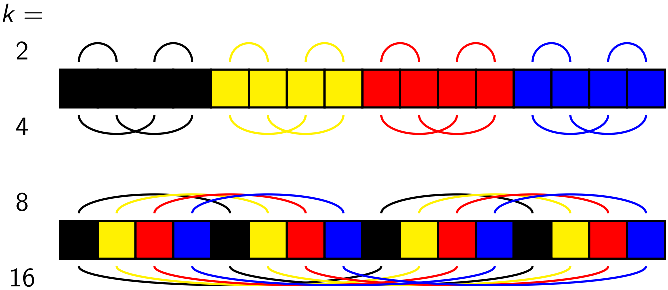
Section 3.4 of Parallel Scientific Computation, 2nd edition

Rob H. Bisseling

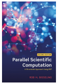
Utrecht University



Data distributions for butterflies of FFT



- n, p must be powers of two with $p < n$. Here: $p = 4$, $n = 16$.
- In stage k, component pair $(x_j, x_{j+k/2})$ at distance $k/2$ is combined.
- Block distribution works for $k = 2, 4, \ldots, n/p$.
- Cyclic distribution works for $k = 2p, 4p, \ldots, n$.

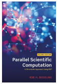


Block distribution works for small butterflies

Let $n = 8, p = 2$. In stage $k = 2$, the vector $\mathbf{x}$ is multiplied by

$$I_4 \otimes B_2 = \begin{bmatrix} 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 \end{bmatrix}.$$

- The first two butterfly blocks $x(0:1)$, $x(2:3)$ are contained in processor block $x(0:3)$.
- The last two butterfly blocks $x(4:5)$, $x(6:7)$ are contained in processor block $x(4:7)$.



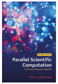
Cyclic distribution works for large butterflies

In stage $k = 8$, the vector $\mathbf{x}$ is multiplied by

$$I_1 \otimes B_8 = B_8 = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \omega & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \omega^2 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \omega^3 \\ 1 & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & -\omega & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -\omega^2 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -\omega^3 \end{bmatrix},$$

where $\omega = \omega_8 = \mathrm{e}^{-\pi i/4} = \frac{1}{2}\sqrt{2} - \frac{1}{2}\sqrt{2}i$.

- The pairs (x_0, x_4) and (x_2, x_6) are combined on $P(0)$.
- The pairs (x_1, x_5) and (x_3, x_7) are combined on $P(1)$.

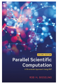


Parallelization strategy: use different distributions

- At the start, for $k \leq n/p$, we use the block distribution.
- Near the end, for $k \geq 2p$, we use the cyclic distribution.
- These two distributions suffice if the block distribution can reach at least up to $k = p$, i.e.,

$$p \leq \frac{n}{p},$$

  which means $p \leq \sqrt{n}$. For example: $p \leq 32$ for $n = 1024$.
- If $p > \sqrt{n}$, we need an additional intermediate distribution, a generalization of the block and cyclic distribution.
- Split the vector into blocks. Each block is owned by a group of processors and is distributed by the cyclic distribution over the processors of that group.



Group-cyclic distribution

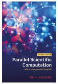
- Let c be fixed such that $1 \leq c \leq p$ and $p \bmod c = 0$. The group-cyclic distribution with cycle c is defined by

$$x_j \longmapsto P\left(\left(j \operatorname{div} \left\lceil\frac{cn}{p}\right\rceil\right) c + \left(j \bmod \left\lceil\frac{cn}{p}\right\rceil\right) \bmod c\right).$$

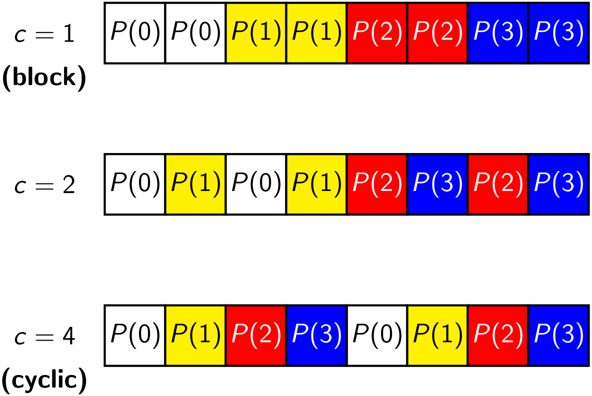
- c is the number of processors in a group and $\left\lceil\frac{cn}{p}\right\rceil = \left\lceil\frac{n}{p/c}\right\rceil$ is the size of a block owned by a group.
- If $n \bmod p = 0$, as happens in the FFT, this reduces to

$$x_j \longmapsto P\left(\left(j \operatorname{div} \frac{cn}{p}\right) c + j \bmod c\right).$$

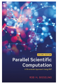
- For $c = 1$, we get the block distribution.
  For $c = p$, we get the cyclic distribution.



From block to cyclic distribution



- Here $n = 8$ and $p = 4$, so that $p > \sqrt{n}$.
- For $c = 2$, we have $p/c = 2$ groups of two processors.



Global and local indices

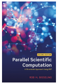
- n, p, and hence c are powers of two, with $1 \leq c < \frac{cn}{p}$.
- Thus, we can write the global index j as

$$j = j_2 \frac{cn}{p} + j_1 c + j_0,$$

  where $0 \leq j_0 < c$ and $0 \leq j_1 < n/p$.
- The processor that owns component x_j is

$$P\left(\left(j \text{ div } \frac{cn}{p}\right) c + j \bmod c\right) = P(j_2 c + j_0).$$

- Processors in the same group have the same j_2, but different j_0.
- We obtain the local index $\mathtt{j}$ by ordering the local components by increasing global index j, so that $\mathtt{j} = j_1$.



Which operations are local?

Butterfly operation on $(x_j, x_{j+k/2})$ is local if

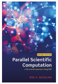
- $x_j, x_{j+k/2}$ are in the same group: $k \leq \frac{cn}{p}$;
- distance $k/2$ is a multiple of c: $k \geq 2c$.

We can use the group-cyclic distribution with cycle c for

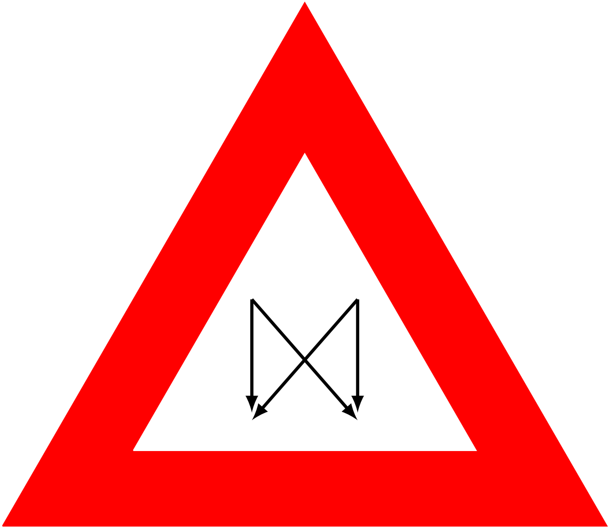
$$2c \leq k \leq \frac{n}{p}c.$$

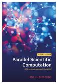
Outline of algorithm:

- start with $c = 1$, perform stages $k = 2, 4, \ldots, n/p$;
- multiply c by n/p, and perform stages $k = 2n/p, 4n/p, \ldots, (n/p)^2$;
- multiply c again by n/p, and so on;
- finish with $c = p$, instead of $c = (n/p)^t \geq p$.



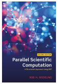
Warning: difficult slides ahead





Parallel unordered FFT: from block to cyclic

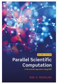
$k := 2$; $c := 1$;
while $k \leq n$ **do**
    $j_0 := s \bmod c$; $j_2 := s \operatorname{div} c$;   ▷ Superstep (0)
    while $k \leq \frac{n}{p} c$ **do**
        $b := \frac{nc}{kp}$;
        for $r := j_2 b$ **to** $(j_2 + 1)b - 1$ **do**
            { Compute local part of $B_k x(rk : (r+1)k - 1)$ }
            for $j := j_0$ **to** $\frac{k}{2} - 1$ **step** c **do**
                $\tau := \omega_k^j x_{rk+j+k/2}$;
                $x_{rk+j+k/2} := x_{rk+j} - \tau$;
                $x_{rk+j} := x_{rk+j} + \tau$;
        $k := 2k$;



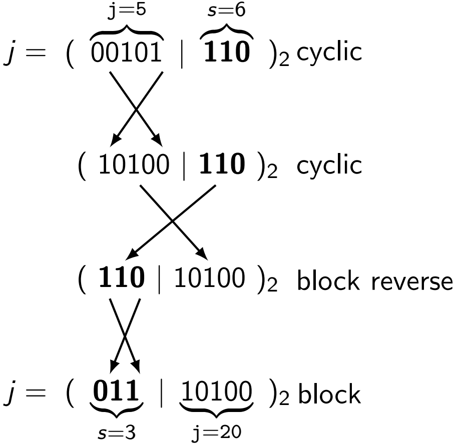
Parallel unordered FFT: from block to cyclic

$k := 2;\ c := 1;$
while $k \le n$ **do**
  $j_0 := s \bmod c;\ j_2 := s \operatorname{div} c;$ ▷ Superstep (0)
  while $k \le \frac{n}{p} c$ **do**
    $b := \frac{nc}{kp};$
    for $r := j_2 b$ **to** $(j_2+1)b - 1$ **do**
      { Compute local part of $B_k x(rk : (r+1)k - 1)$ }
      for $j := j_0$ **to** $\frac{k}{2} - 1$ **step** c **do**
        $\tau := \omega_k^j x_{rk+j+k/2};$
        $x_{rk+j+k/2} := x_{rk+j} - \tau;$
        $x_{rk+j} := x_{rk+j} + \tau;$
    $k := 2k;$

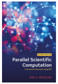
  if $c < p$ **then** ▷ Superstep (1)
    $c_0 := c;\ c := \min(\frac{n}{p} c, p);$
    Redistr$(\mathbf{x}, n, p, c_0, c, rev);$



Parallel bit reversal: from cyclic to block

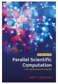


- Example with $p = 8$, $n = 256$, for $j = 46 = (00101110)_2$.
- Start in the cyclic distribution with a local bit reversal.
- Then swap the data between $P(s)$ and $P(\rho_p(s))$.
- We end in the block distribution, with $j = 116 = (01110100)_2$.



Postponing the data swaps

- The distribution just before the swaps is the block distribution with bit-reversed processor numbering.
- All processors perform the same operations in FFT stages $k = 2, 4, \ldots, n/p$, multiplying local blocks of $\mathbf{x}$ by B_k.
- I'll scratch your back if you scratch mine: processors perform the work of their partner.
- The data swaps can be postponed until the first redistribution, immediately after stage $k = n/p$.
- Buy 2, Pay 1: two permutations can be done at the cost of one by combining them. Hence no extra communication is incurred by the data swaps.

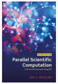


Redistribution from cycle c_0 to cycle c

function $\textsc{Redistr}(\mathbf{x}, n, p, c_0, c, rev)$

if *rev* **then**
{ Reverse the processor numbering }
$j_0 := \rho_p(s) \bmod c_0$;
$j_2 := \rho_p(s) \operatorname{div} c_0$;
else
$j_0 := s \bmod c_0$;
$j_2 := s \operatorname{div} c_0$;

for $j := j_2 \frac{c_0 n}{p} + j_0$ **to** $(j_2 + 1)\frac{c_0 n}{p} - 1$ **step** c_0 **do**
$dest := (j \operatorname{div} \frac{cn}{p})c + j \bmod c$;
put x_j in $P(dest)$;



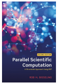
Last iteration of main loop

- The last iteration is determined by the smallest integer t such that $(n/p)^t \geq p$.
- The cycles of the iterations are $c = (n/p)^0, (n/p)^1, \ldots, (n/p)^{t-1}, p$.
- The total number of iterations is therefore $t + 1$.
- For $n = 2^m$ and $p = 2^q$, we have

$$\begin{aligned}(n/p)^t \geq p &\iff n^t \geq p^{t+1} \iff 2^{mt} \geq 2^{q(t+1)} \\ &\iff mt \geq q(t+1) \iff mt - qt \geq q \\ &\iff t \geq \frac{q}{m-q}.\end{aligned}$$

- It follows that

$$t = \left\lceil \frac{q}{m-q} \right\rceil.$$



BSP cost

- Every iteration, except the last, has a computation superstep and a communication superstep that redistributes the data.
- The last iteration has no data redistribution.
- The total synchronization time is therefore

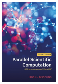
$$T_{\mathrm{sync}} = (2t+1)l = \left(2\left\lceil \frac{q}{m-q} \right\rceil + 1\right) l.$$

- Every redistribution moves at most all the local data in and out, i.e., n/p complex numbers, or $2n/p$ real data words.
- The total communication time is therefore

$$T_{\mathrm{comm}} = t \cdot \frac{2n}{p} g = \left\lceil \frac{q}{m-q} \right\rceil \cdot \frac{2n}{p} g.$$

- Look mama, without counting! The total computation time is

$$T_{\mathrm{comp}} = \frac{5n \log_2 n}{p}.$$



Summary

- We have used different distributions in different parts of the algorithm, trying to make our operations local.
- The algorithm starts and finishes in the cyclic distribution.
- If we split a vector into p/c blocks and distribute each block over c processors by the cyclic distribution, then we obtain the group-cyclic distribution with cycle c.
- The total BSP cost of the parallel FFT algorithm is

$$T_{\mathrm{FFT}} = \frac{5n\log_2 n}{p} + 2 \cdot \left\lceil \frac{\log_2 p}{\log_2(n/p)} \right\rceil \cdot \frac{n}{p} g + \left(2 \left\lceil \frac{\log_2 p}{\log_2(n/p)} \right\rceil + 1 \right) l.$$

- For practical p, we only need one data redistribution:

$$T_{\mathrm{FFT},\ 1<p\leq\sqrt{n}} = \frac{5n\log_2 n}{p} + 2\frac{n}{p} g + 3l.$$