Parallel FFT

Sathish Vadhiyar
Sequential FFT – Quick Review

Y[i] = \sum_{k=0}^{n-1} X[k] \omega^{ki}, 0 \leq i < n

\omega = e^{2\pi \sqrt{-1}/n}

Twiddle factor – primitive n\textsuperscript{th} root of unity in complex plane -

Y[i] = \sum_{k=0}^{(n/2)-1} X[2k] \omega^{2ki} + \sum_{k=0}^{(n/2)-1} X[2k + 1] \omega^{(2k+1)i}

= \sum_{k=0}^{(n/2)-1} X[2k] e^{2\pi \sqrt{-1}/(n/2)} + \sum_{k=0}^{(n/2)-1} X[2k + 1] \omega^i e^{2\pi \sqrt{-1}/(n/2) ki}

= \sum_{k=0}^{(n/2)-1} X[2k] e^{2\pi \sqrt{-1} ki/(n/2)} + \omega^i \sum_{k=0}^{(n/2)-1} X[2k + 1] e^{2\pi \sqrt{-1} ki/(n/2)}
Sequential FFT – Quick Review

- $(n/2)^{th}$ root of unity

\[ \omega = e^{2\pi \sqrt{-1}/(n/2)} = \omega^2 \]

- $2$ $(n/2)$-point DFTs

\[ Y[i] = \sum_{k=0}^{(n/2)-1} X[2k] \omega^{ki} + \omega^i \sum_{k=0}^{(n/2)-1} X[2k + 1] \omega^{ki} \]
Sequential FFT – quick review
Sequential FFT – recursive solution

1. procedure R_FFT(X, Y, n, w)
2. if (n=1) then Y[0] := X[0] else
3. begin
4.   R_FFT(<X(0), X(2), ..., X[n-2]>,
6.       <Q[0], Q[1], ..., Q[n/2]>, n/2, w^2)
5.   R_FFT(<X(1), X(3), ..., X[n-1]>,
6.       <T[0], T[1], ..., T[n/2]>, n/2, w^2)
6. for i := 0 to n-1 do
7.   Y[i] := Q[i mod (n/2)] + w^iT(i mod (n/2));
8. end R_FFT
Sequential FFT – iterative solution

1. procedure ITERATIVE_FFT(X, Y, n)
2. begin
3.     r := log n;
4.     for i := 0 to n-1 do R[i] := X[i];
5.     for m := 0 to r-1 do
6.         begin
7.             for i := 0 to n-1 do S[i] := R[i];
8.             for i := 0 to n-1 do
9.                 begin
/* Let \((b_0, b_1, b_2, \ldots, b_{r-1})\) be the binary representation of \(i\) */
10.                j := \((b_0 \ldots b_{m-1} 0b_{m+1} \ldots b_{r-1})\);
11.                k := \((b_0 \ldots b_{m-1} 1b_{m+1} \ldots b_{r-1})\);
12.                R[i] := S[j] + S[k] \times w^{(b_{m-1}\ldots b_0 0\ldots 0)} ;
13.             endfor;
14.         endfor;
15.     for i := 0 to n-1 do Y[i] := R[i];
16. end ITERATIVE_FFT
Example of $w$ calculation

For a given $m$ and $i$, the power of $w$ is computed by reversing the order of the $m+1$ most significant bits of $i$ and padding them by 0’s to the right.

<table>
<thead>
<tr>
<th>m/ i</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>000</td>
<td>000</td>
<td>100</td>
<td>100</td>
<td>010</td>
<td>010</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>000</td>
<td>100</td>
<td>010</td>
<td>110</td>
<td>001</td>
<td>101</td>
<td>011</td>
<td>111</td>
</tr>
</tbody>
</table>
Parallel FFT – Binary exchange

000\textsuperscript{X(0)} \rightarrow Y(0) \quad P_0

001\textsuperscript{X(1)} \rightarrow Y(1)

010\textsuperscript{X(2)} \rightarrow Y(2) \quad P_1

011\textsuperscript{X(3)} \rightarrow Y(3)

100\textsuperscript{X(4)} \rightarrow Y(4)

101\textsuperscript{X(5)} \rightarrow Y(5) \quad P_2

110\textsuperscript{X(3)} \rightarrow Y(6)

111\textsuperscript{X(7)} \rightarrow Y(7) \quad P_3
Binary Exchange

- $d$ – number of bits for representing processes; $r$ – number of bits representing the elements
- The $d$ most significant bits of element $i$ indicate the process that the element belongs to.
- Only the first $d$ of the $r$ iterations require communication.
- In a given iteration, $m$, a process $i$ communicates with only one other process obtained by flipping the $(m+1)$th MSB of $i$.
- Total execution time: $\frac{n}{P} \log N + \log P(l) + \left(\frac{n}{P}\right) \log P (b)$
Parallel FFT – 2D Transpose

Phase 1 – FFTs along columns
Parallel FFT – 2D Transpose

Phase 2 – Transpose
Parallel FFT – 2D Transpose

Phase 3 – FFTs along columns
2D Transpose

- In general, n elements arranged as $\sqrt{n} \times \sqrt{n}$
- p processes arranged along columns. Each process owns $\sqrt{n}/p$ columns
- Each process does $\sqrt{n}/p$ FFTs of size $\sqrt{n}$ each
- Parallel runtime $= 2(\sqrt{n}/p)\sqrt{n}\log\sqrt{n} + (p-1)(l) + n/p(b)$
3D Transpose

- $n^{1/3} \times n^{1/3} \times n^{1/3}$ elements
- $\sqrt{p} \times \sqrt{p}$ processes
- Steps ?
- Parallel runtime –

$\frac{n}{p}\log n(c) + 2(\sqrt{p}-1)(l) + 2\left(\frac{n}{p}\right)(b)$
In general

- For q dimensions:
- Parallel runtime –
  \[(n/p)\log n + (q-1)(p^{1/(q-1)} - 1) [l] + (q-1)(n/p) [b]\]
- Time due to latency decreases; due to bandwidth increases
- For implementation – only 2D and 3D transposes are feasible. Moreover, there are restrictions on n and p in terms of q.
Choice of algorithm

- Binary exchange – small latency, large bandwidth
- 2D transpose – large latency, small bandwidth
- Other transposes lie between binary exchange and 2D transpose
- For a given parallel computer, based on l and b, different algorithms can give different performances for different problem sizes