CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

September 13th, 2017

Gaussian Elimination Problem

Gaussian elimination uses the elementary array operations to turn a matrix into row echelon form.

$$\begin{bmatrix} 4 & 3 & 2 & 5 \\ 6 & 7 & 8 & 2 \\ 3 & 1 & 5 & 6 \\ 7 & 8 & 3 & 4 \end{bmatrix}$$

$$\frac{1}{4}R_1 \to R_1$$

$$\begin{bmatrix} 1 & \frac{3}{4} & \frac{2}{4} & \frac{5}{4} \\ 6 & 7 & 8 & 2 \\ 3 & 1 & 5 & 6 \\ 7 & 8 & 3 & 4 \end{bmatrix}$$

$$-6R_1 + R_2 \to R_2$$

$$\begin{bmatrix} 1 & \frac{3}{4} & \frac{2}{4} & \frac{5}{4} \\ 0 & 7 - \frac{18}{4} & 8 - \frac{12}{4} & 2 - \frac{30}{4} \\ 3 & 1 & 5 & 6 \\ 7 & 8 & 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \frac{3}{4} & \frac{2}{4} & \frac{5}{4} \\ 0 & \frac{5}{2} & 5 & \frac{-11}{2} \\ 3 & 1 & 5 & 6 \\ 7 & 8 & 3 & 4 \end{bmatrix}$$

Bitonic Sort

Compare Exchange: With two processors P_i and P_j, this operation causes them to exchange their respective values x_i and x_j with P_i maintaining $max(x_i, x_j)$ and P_j maintaining $min(x_i, x_j)$.
Compare Split: With two processors P_i and P_j, the two processors take $\frac{N}{P}$ elements to sort individually, and exchange them at the end of their respective sorts. In the case where $N >> P$, this has a complexity $O(\frac{N}{P}\log_2\frac{N}{P})$.

Example

Suppose you have two processors P_i and P_j and in the splitting the input, P_i receives [2, 8, 6, 4] and P_j receives [7, 3, 1, 5]. They will each respectively sort their sublists:

$$P_i : [2, 4, 6, 8]$$

$$P_j : [1, 3, 5, 7]$$

And then perform a compare split:

$$P_i : [1, 2, 3, 4]$$

$$P_j : [5, 6, 7, 8]$$

Bitonic Sequence

A bitonic sequence is a sequence containing an ascending sequence followed by a descending sequence.

$$a_1, a_2, a_3, a_4, \ldots, a_{\frac{N}{2}}$$

$$a_{\frac{N}{2}+1}, a_{\frac{N}{2}+2}, \ldots, a_N$$

You can merge elements of a bitonic sequence in $\log_2 n$ steps to get a sorted list. Each step has $\frac{N}{2}$ swaps.

$$[18, 36, 72, 90, 52, 39, 26, 13]$$

In the above example, the data size is 8. $N = 8 = 2^k, k = 3$. We compare element i with element $i + 2^{k-1}$ from 1 to 2^{k-1}. If element i is less than element $i + 2^{k-1}$, we exchange them. This results in:

$$[18, 36, 26, 13][52, 39, 72, 90]$$

Between the two groups on the two processors, we compare element i with $i + 2^{k-2}$, resulting in:

$$[18, 13][26, 36][52, 39][72, 90]$$

We then compare element i with element $i + 2^{k-2}$:

$$[13, 18, 26, 36, 39, 52, 72, 90]$$

Note that this split-comparison process can be done recursively in $\log n$ time. This is better than doing an $O(n)$ merge on the bitonic sequence.

Example

Turning an array of numbers into a bitonic sequence:

$$\begin{array}{lllllll}
98 & \to & 28 & \to & 28 & \to & 28 \\
28 & \to & 98 & \to & 56 & \to & 56 \\
70 & \to & 70 & \to & 70 & \to & 70 \\
56 & \to & 56 & \to & 98 & \to & 98 \\
42 & \to & 42 & \to & 84 & \to & 112 \\
14 & \to & 14 & \to & 112 & \to & 84 \\
112 & \to & 84 & \to & 42 & \to & 42 \\
86 & \to & 112 & \to & 14 & \to & 14
\end{array}$$

Reminders

Professor Mohan Kumar:
`mjkvcs@rit.edu`
`https://cs.rit.edu/~mjk`

Rahul Dashora (TA):
`rd5476@mail.rit.edu`

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech