Divide and Conquer Algorithms

Sathish Vadhiyar
Introduction

- One of the important parallel algorithm models

- The idea is to
  - decompose the problem into parts
  - solve the problem on smaller parts
  - find the global result using individual results

- Works naturally and works well for parallelization
Introduction

- Various models
  - Recursive sub-division: Has a division and computation phase, then a merge phase. E.g., merge sort
  - Local compute – merge/coordinate – local compute. E.g., sample sort
Recursive sub-division:
- Merge sort (you know already)
- Solving tri-diagonal systems
- FFT
Parallel solution of linear system with special matrices

Tridiagonal Matrices

\[
\begin{pmatrix}
  a_1 & h_1 & & \\
  g_2 & a_2 & h_2 & \\
    & g_3 & a_3 & h_3 \\
    &    & \ddots & \ddots \\
  g_n & a_n & & h_n
\end{pmatrix}
\begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{pmatrix}
=
\begin{pmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_n
\end{pmatrix}
\]

In general:

\[g_i x_{i-1} + a_i x_i + h_i x_{i+1} = b_i\]

Substituting for \(x_{i-1}\) and \(x_{i+1}\) in terms of \(\{x_{i-2}, x_i\}\) and \(\{x_i, x_{i+2}\}\) respectively:

\[G_i x_{i-2} + A_i x_i + H_i x_{i+2} = B_i\]
Tridiagonal Matrices

\[
\begin{align*}
A_1 & \quad H_1 \\
A_2 & \quad H_2 \\
G_3 & \quad A_3 & \quad H_3 \\
G_4 & \quad A_4 & \quad H_4 & \ldots \\
\end{align*}
\]

\[
x_1 \quad B_1 \\
x_2 \quad B_2 \\
x_3 \quad B_3 \\
\ldots \\
x_n \quad B_n
\]

Reordering:
Tridiagonal Matrices

\[
\begin{array}{ccc}
A_2 & H_2 \\
G_4 & A_4 & H_4 \\
& & \\
& & \\
A_1 & H_1 \\
G_3 & A_3 & H_3 \\
& & \\
& & \\
G_{n-3} & A_{n-1}
\end{array}
\]

\[
x_2 \quad x_4 \quad \cdots \quad x_n \\
= \\
B_2 \quad B_4 \quad \cdots \quad B_n
\]
Tridiagonal Systems

- Thus the problem of size $n$ has been split into even and odd equations of size $n/2$
- This is **odd–even** reduction
- For parallelization, each process can divide the problem into subproblems of smaller size and solve the subproblems
- This is **divide-and-conquer** technique
Tridiagonal Systems - Parallelization

- At each stage one representative process of the domain of processes is chosen.
- This representative performs the odd-even reduction of problem $i$ to two problems of size $i/2$.
- The problems are distributed to 2 representatives.

Diagram:

```
  n
 /  \
/    \
 n/2  n/2
 /  \\  /  \\
n/4 n/4 n/4 n/4
 /  \\  /  \\
n/8 n/8 n/8 n/8
```

Index:

1  2  3  4  5  6  7  8
Local compute – merge – local compute

- Prefix Computations
- Sample sort