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

$$\begin{bmatrix} a1 & h1 & & & \\ g2 & a2 & h2 & & \\ & g3 & a3 & h3 & \\ & & \ddots & \ddots & \ddots \\ & & & gn & an \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \\ . \\ . \\ xn \end{bmatrix} = \begin{bmatrix} b1 \\ b2 \\ b3 \\ . \\ . \\ bn \end{bmatrix}$$

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

$$
\begin{bmatrix}
A1 & & H1 & & & & \\
 & A2 & & H2 & & & \\
G3 & & A3 & & H3 & & \\
 & G4 & & A4 & & H4 & \\
 & & \ddots & & \ddots & & \ddots \\
 & & & & Gn-2 & & An
\end{bmatrix}
\begin{bmatrix} x1 \\ x2 \\ x3 \\ . \\ . \\ xn \end{bmatrix}
=
\begin{bmatrix} B1 \\ B2 \\ B3 \\ . \\ . \\ Bn \end{bmatrix}
$$

Reordering:

Tridiagonal Matrices

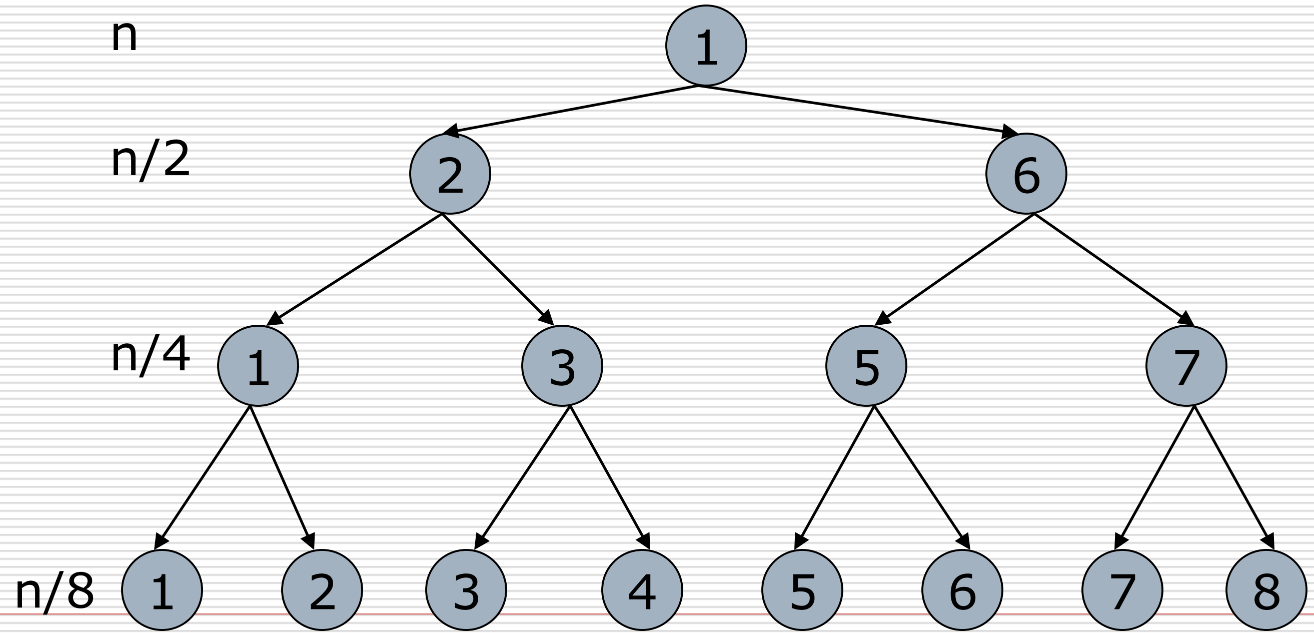
$$
\left[\begin{array}{cccc}
A2 & H2 & & \\
G4 & A4 & H4 & \\
 & \ddots & \ddots & \ddots \\
 & & Gn & An \\
\hline
A1 & H1 & & \\
G3 & A3 & H3 & \\
 & \ddots & \ddots & \ddots \\
 & & Gn\text{-}3 & An\text{-}1
\end{array}\right]
\left[\begin{array}{c}
x2 \\ x4 \\ . \\ xn \\ \hline x1 \\ x3 \\ . \\ xn\text{-}1
\end{array}\right]
=
\left[\begin{array}{c}
B2 \\ B4 \\ . \\ Bn \\ \hline B1 \\ B3 \\ . \\ Bn\text{-}1
\end{array}\right]
$$

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



- Local compute – merge – local compute
  - Prefix Computations
  - Sample sort