Parallel Program Design: Decomposition Techniques

Ned Nedialkov

McMaster University
Canada

CS/SE 4F03
February 2014

Outline

Data Decomposition

1. Decompose the data
2. The decomposition is used to induce computational tasks

- Partitioning output data: each part of the output can be computed independently of the rest
  - Each task computes parts of the output
- Partitioning input data: each tasks works on part of the input data
  - may not be possible to partition based on the output; e.g sorting a sequence, finding largest number

Example: partitioning output data

Multiplying two matrices

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$C_{11} = A_{11}B_{11} + A_{12}B_{21} \quad (1)$$

$$C_{12} = A_{11}B_{12} + A_{12}B_{22} \quad (2)$$

$$C_{21} = A_{21}B_{11} + A_{22}B_{21} \quad (3)$$

$$C_{22} = A_{21}B_{12} + A_{22}B_{22} \quad (4)$$

- The numbers on the right denote task numbers
- We can compute the result with 4 tasks
- Each tasks does two multiplications in $O(2n^3/8) = O(n^3/4)$ and one addition in $O(n^2/4)$
- Total in parallel is $O(n^3/4 + n^2/4) = O(n^3/4)$

Partitioning intermediate results

Consider

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

We can compute

$$\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \end{bmatrix} = \begin{bmatrix} D_{11}^{(1)} & D_{12}^{(1)} \\ D_{21}^{(1)} & D_{22}^{(1)} \end{bmatrix}$$

$$\begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} D_{11}^{(2)} & D_{12}^{(2)} \\ D_{21}^{(2)} & D_{22}^{(2)} \end{bmatrix}$$

and then

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} D_{11}^{(1)} & D_{12}^{(1)} \\ D_{21}^{(1)} & D_{22}^{(1)} \end{bmatrix} + \begin{bmatrix} D_{11}^{(2)} & D_{12}^{(2)} \\ D_{21}^{(2)} & D_{22}^{(2)} \end{bmatrix}$$

- 8 tasks can run in parallel to multiply $n/2 \times n/2$ matrices
  - Each tasks does $O(n^3/8)$ work
- 4 tasks can add them in parallel
  - Each task does $n^2/4$ work
- Total in parallel is $O(n^3/8 + n^2/4) = O(n^3/8)$
- About 2 times faster than if partitioning the output

Recursive decomposition

- Recursively subdivide the problem
- Solve subproblems in parallel
- Combine the results
- E.g. parallel quicksort

Exploratory decomposition

- Assume we have a large solution space, and we need to search for a solution in it
- Decompose this space into smaller parts
- Search in parallel in these smaller parts to find a solution
- E.g. solving the 15-puzzle problem