Data and Work Partitioning

Lecture-12

Parallel & Distributed Computing

Course Outlines

Course Name: Parallel and Distributed Computing
Credit Hours: 3(3-0)
Prerequisites: Data Communications and Computer Networks

Course Outlines:

Why use parallel and distributed systems? Why not use them? Speedup and Amdahl's Law, Hardware architectures: multiprocessors (shared memory), networks of workstations (distributed memory), clusters (latest variation). Software architectures: threads and shared memory, processes and message passing, distributed shared memory (DSM), distributed shared data (DSD). Possible research and project topics, Parallel Algorithms, Concurrency and synchronization, Data and work partitioning, Common parallelization strategies, Granularity, Load balancing, Examples: parallel search, parallel sorting, etc. Shared-Memory Programming: Threads, Pthreads, Locks and semaphores, Distributed-Memory Programming: Message Passing, MPI, PVM. Other Parallel Programming Systems, Distributed shared memory, Aurora: Scoped behaviour and abstract data types, Enterprise: Process templates. Research Topics.

Decomposition (Partitioning)

- One of the fundamental steps that we need to undertake to solve a problem in parallel is to split the computations to be performed into a set of tasks for concurrent execution defined by the ***task-dependency graph***.

Decomposition (Partitioning)

Decomposition Techniques are broadly classified as:

1. **Recursive decomposition**

2. **Data decomposition**

3. **Exploratory decomposition**

4. **Speculative decomposition**.

Decomposition (Partitioning) cont…

- The ***recursive*** and ***data decomposition*** techniques are relatively ***general purpose*** as they can be used to decompose a ***wide variety of problems***.
- On the other hand, ***speculative*** and ***exploratory decomposition*** techniques are more of a ***special purpose*** nature because they apply to ***specific classes of problems***.

Decomposition (Partitioning) cont…
Recursive Decomposition

- Recursive decomposition is a method for inducing concurrency in problems that can be solved using the ***divide-and-conquer strategy***.
- In this technique, a problem is solved by first dividing it into a set of independent sub-problems.

Decomposition (Partitioning) cont… Recursive Decomposition

- Each one of these sub-problems is solved by ***recursively*** applying a ***similar division into smaller subproblems*** followed by a ***combination of their results***.
- The divide-and-conquer strategy results in natural concurrency, as different sub-problems can be solved concurrently.

Decomposition (Partitioning) cont…
Recursive Decomposition (Quicksort)

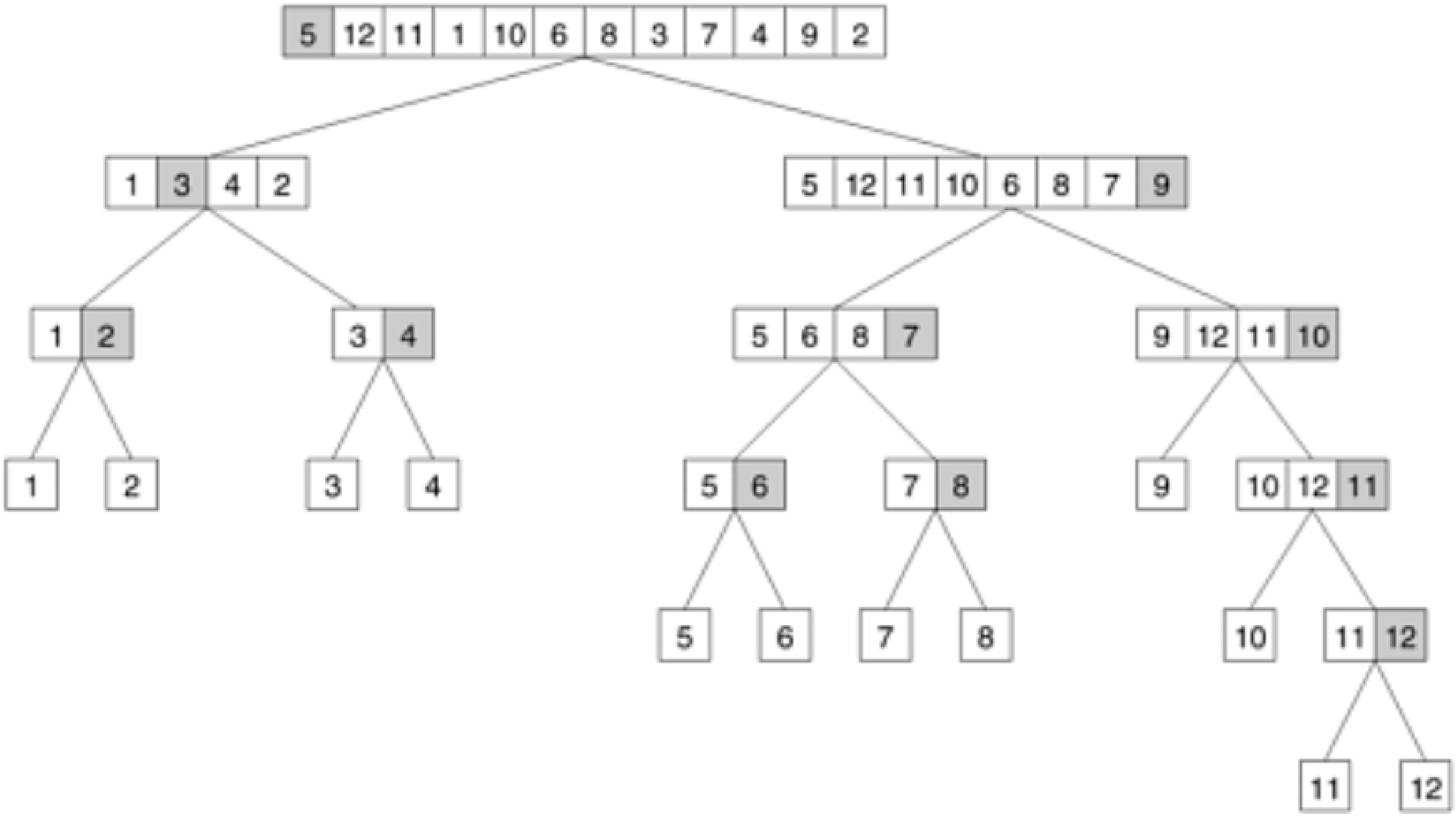
- Consider the sorting a sequence A of n elements using quicksort algorithm.

- Quicksort is a divide and conquer algorithm that starts by selecting a ***pivot element x*** and then ***partitions the sequence A*** into ***two subsequences A_0*** and A_1 such that all the elements in A_0 are **smaller than *x*** and all the elements in A_1 are **greater than or equal to *x***.

Decomposition (Partitioning) cont… Recursive Decomposition (Quicksort)

- This partitioning step forms the ***divide* step** of the algorithm.
- Each one of the subsequences A_0 and A_1 is sorted by **recursively calling quicksort**.
- Each one of these recursive calls further partitions the sequences.
- This is illustrated in following Figure for a sequence of 12 numbers.

Decomposition (Partitioning) cont…
Recursive Decomposition (Quicksort)



Decomposition (Partitioning) cont…
Recursive Decomposition (Quicksort)

- The recursion terminates when each subsequence contains only a single element.
- Then the results will be combined to form a sorted list.

Decomposition (Partitioning) cont…
Data Decomposition

- **Data decomposition** is a powerful and commonly used method for deriving concurrency in algorithms that operate on **large data** structures.
- In this method, the decomposition of computations is done in two steps:

Decomposition (Partitioning) cont… Data Decomposition

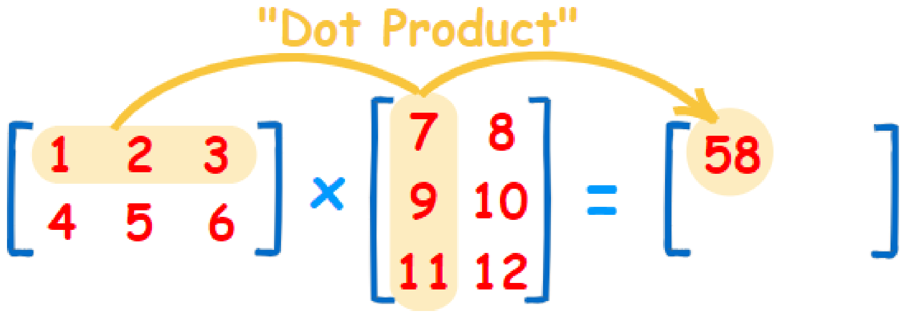
1. In the first step, the data on which the computations are performed is **partitioned**, and
2. In the second step, this data partitioning is used to induce a partitioning of the computations into tasks.

Decomposition (Partitioning) cont… Data Decomposition

- The operations that these tasks perform on different data partitions are usually similar.
- The partitioning of data can be performed in many possible ways.
- But we are going to discuss **matrix-multiplication**.

Decomposition (Partitioning) cont…
Data Decomposition (Matrix Multiplication)

- To multiply a matrix by another matrix we need to do "dot product" of rows and columns ... what does that mean?
- Let see with example: **1st Row X 1st Column:**
- (1, 2, 3) • (7, 9, 11) = 1×7 + 2×9 + 3×11 = **58**



Decomposition (Partitioning) cont…
Data Decomposition (Matrix Multiplication)

- (1, 2, 3) • (7, 9, 11) = 1×7 + 2×9 + 3×11 = **58**
- (1, 2, 3) • (8, 10, 12) = 1×8 + 2×10 + 3×12= **64**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ & \end{bmatrix}$$

Decomposition (Partitioning) cont…
Data Decomposition (Matrix Multiplication)

- **1st Row X 1st Column:** (1, 2, 3) • (7, 9, 11) = 1×7 + 2×9 + 3×11 = **58**
- **1st row X 2nd column**: (1, 2, 3) • (8, 10, 12) = 1×8 + 2×10 + 3×12= **64**
- **2nd row** X **1st column**: (4, 5, 6) • (7, 9, 11) = 4×7 + 5×9 + 6×11= **139**
- **2nd row** X **2nd column**: (4, 5, 6) • (8, 10, 12) = 4×8 + 5×10 + 6×12= **154**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

Decomposition Techniques cont…
Exploratory Decomposition

- ***Exploratory decomposition*** is used to decompose problems whose underlying computations correspond to a searching of a solution from search space.
- In exploratory decomposition, we partition the ***search space*** into ***smaller parts***, and search each one of these parts concurrently, until the desired solutions are found.

Decomposition Techniques cont… Speculative Decomposition

- ***Speculative decomposition*** is used when a program may take one of many possible computationally significant branches depending on the output of other computations that precede it.
- In this situation, while one task is performing the computation whose output is used in deciding the next computation, other tasks can concurrently start the computations of the next stage.

Decomposition Techniques cont… Speculative Decomposition

- This scenario is similar to evaluating one or more of the branches of a *switch* statement in C in parallel before the input for the *switch* is available.