Understanding
Parallel Computers - Paradigms and Programming Models

Sofien GANNOUNI
Computer Science
E-mail: gnnosf@ksu.edu.sa ; gansof@yahoo.com

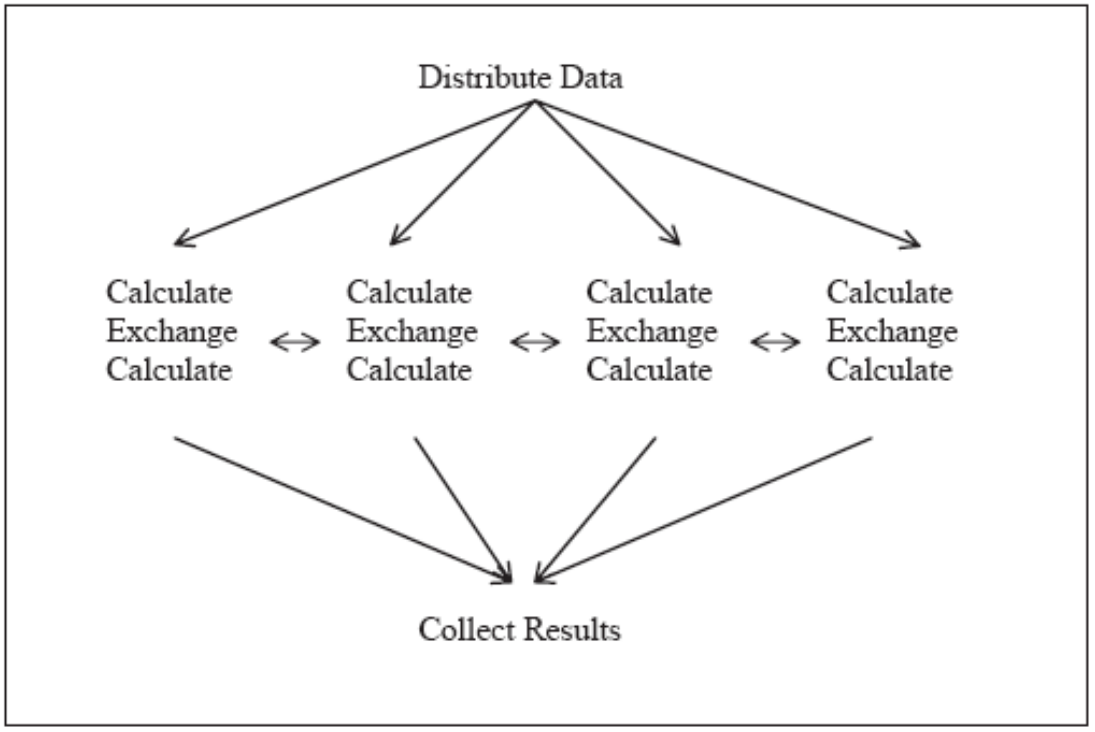
Parallel Programming Models

- SPMD
- Task Farming
- Divide and Conquer
- Dataflow

Single Program Multiple Data (SPMD)

- SPMD model is the dominant pattern to structure parallel programs.
- A single program is loaded into each node of a parallel system.
- Node are executing the code independently but act on multiple data sets including "private" and " shared " data.
- Nodes cooperating in the execution of the program are assigned unique IDs, allowed to self-schedule themselves, and dynamically get assigned to the required tasks under this cooperative execution.

SPMD: The Concept

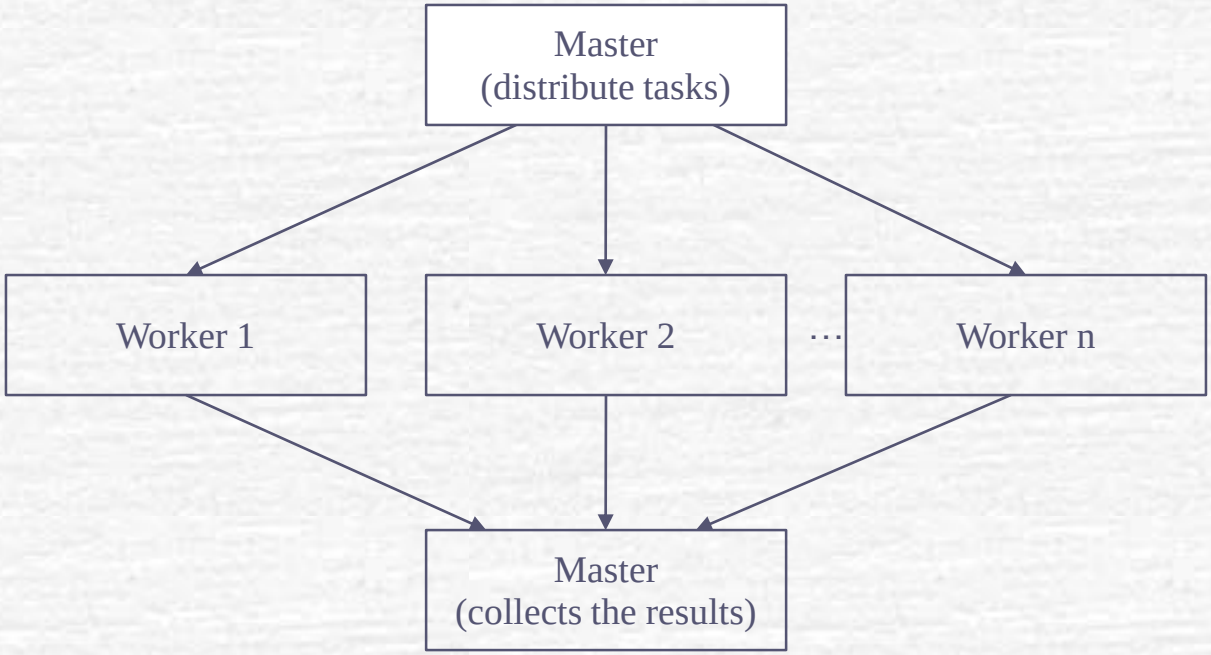


Basic structure of SPMD program

Task Farming Model

- **Task farming (or Master-Worker or Master-Slave or Manager-Worker) consists of two entities:**
  - the master
  - and many workers.
- **The master:**
  - decomposes the problem into small tasks,
  - then distributes/farms these tasks among a farm of workers
  - and finally collects the partial results to produce the final result of the computation.
- **The worker:**
  - receives a task, process it and send the result back to the master.

Basic Task Farming Structure



Simple Task Farming Algorithm

Initialization

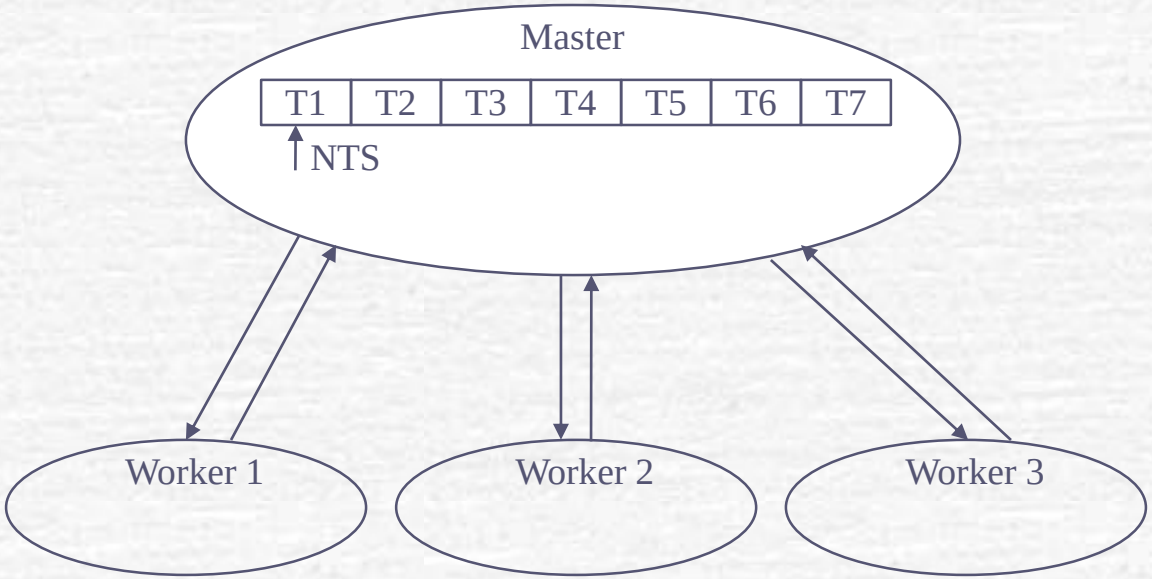
For task = 1 to N

PartialResult = + Function (task) ← Worker Tasks

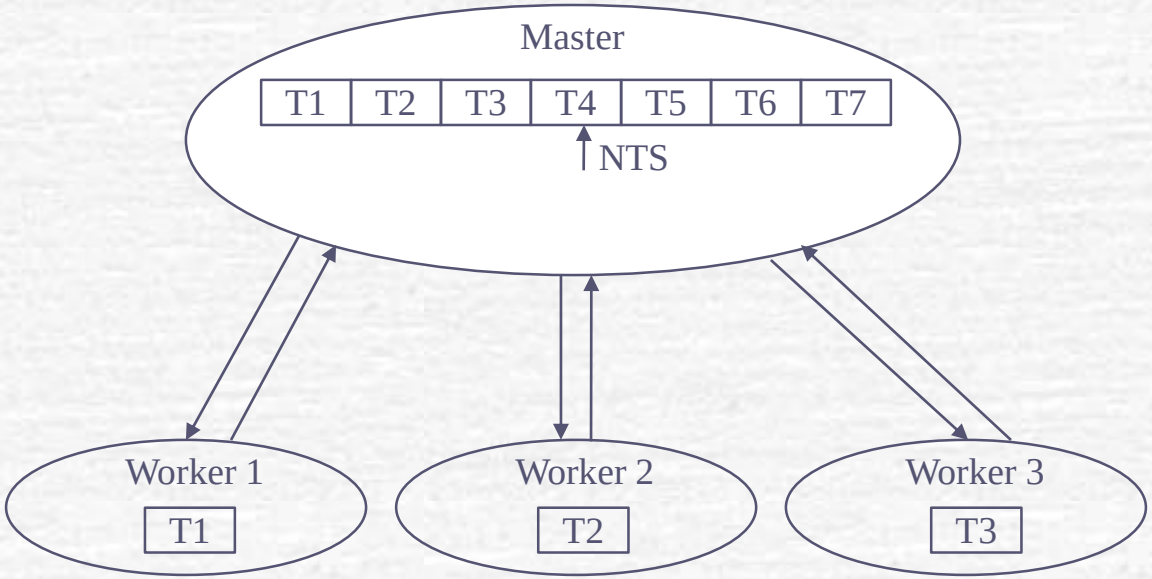
End

act_on_tasks_complete() ← Master Tasks

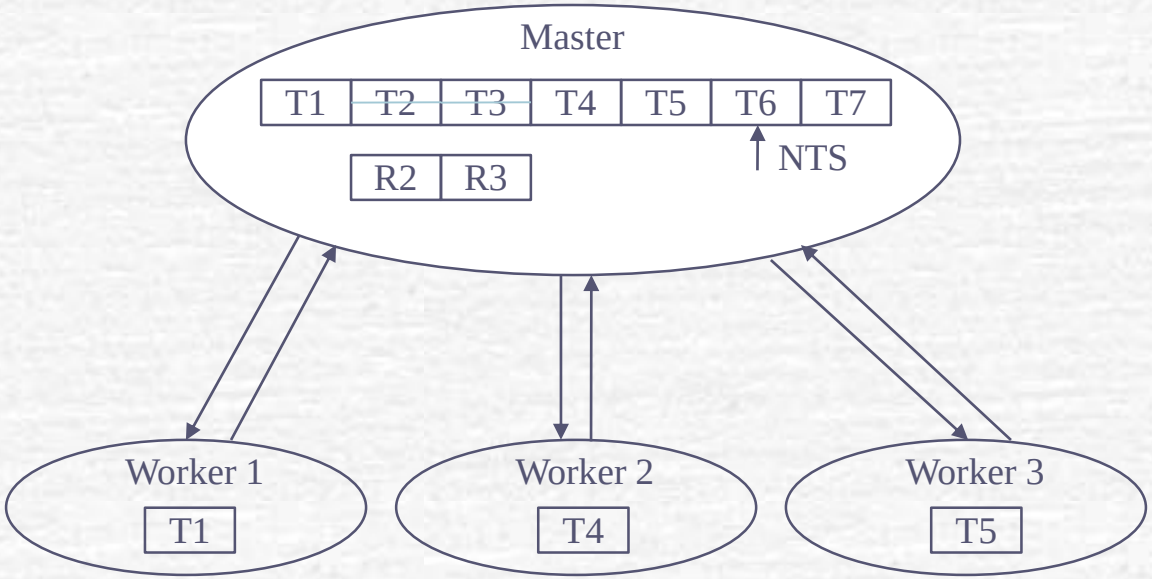
Simple Task Farming Example (a)



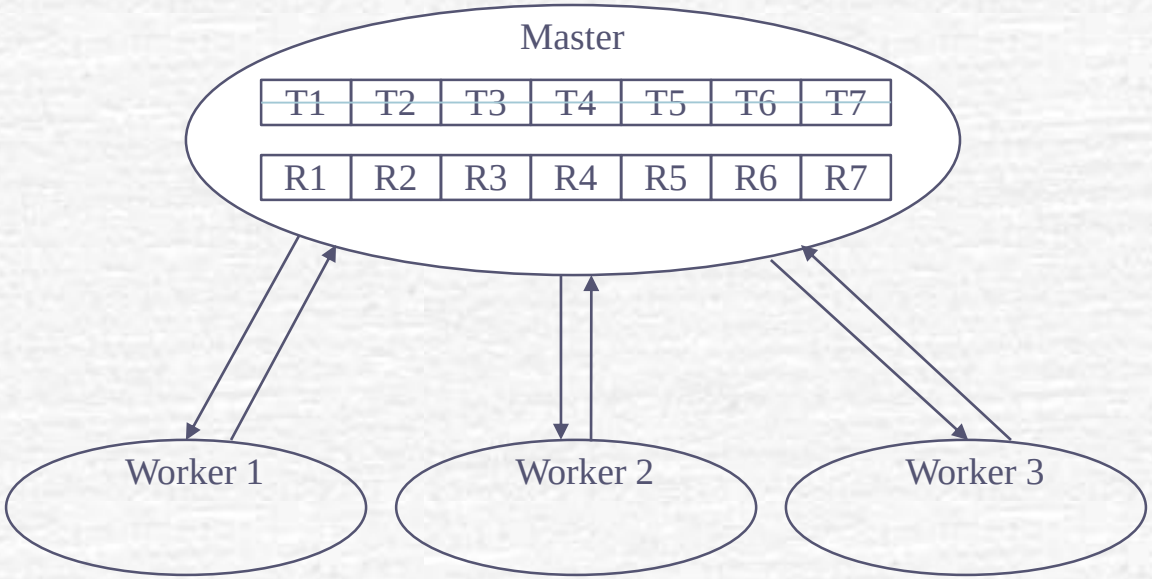
Simple Task Farming Example (b)



Simple Task Farming Example (c)



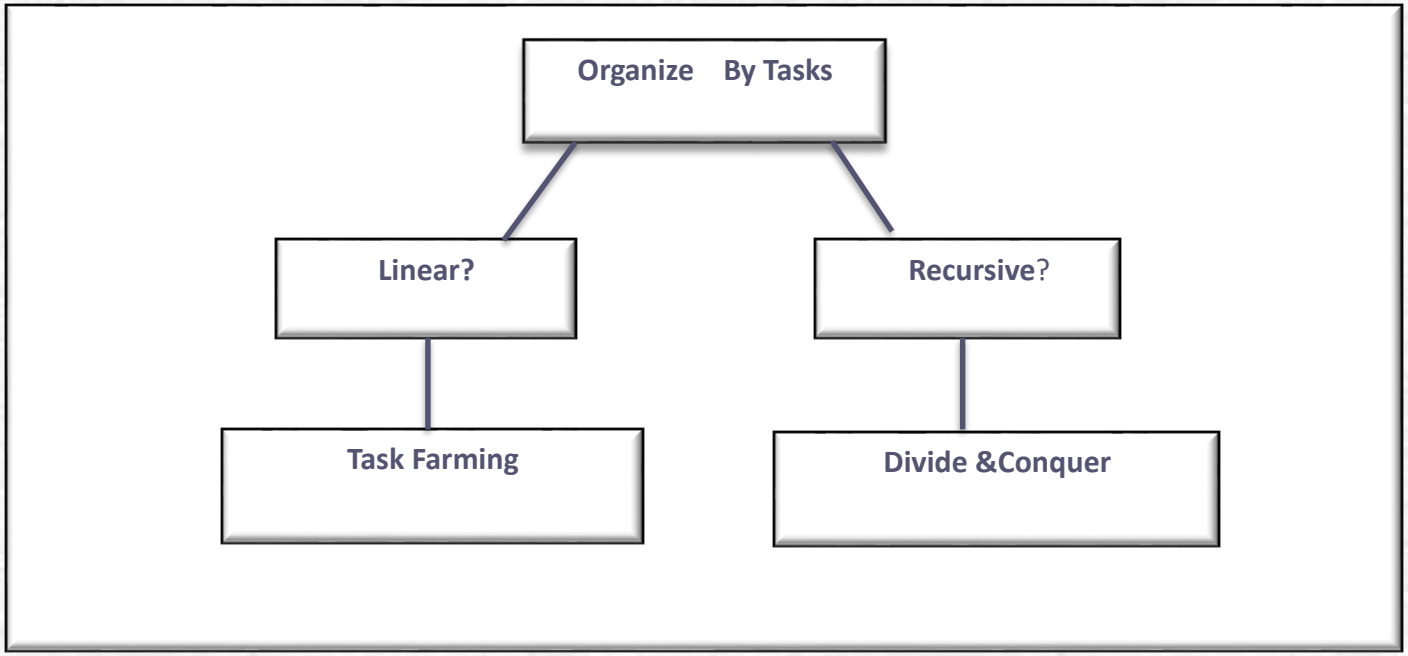
Simple Task Farming Example (d)



Divide and Conquer Model

- This model solves a complex problem by splitting it into smaller easier sub-problems.
- The sub-problems have the same nature as the original one and could be solved by recursively applying the same algorithm.
- The recursion stops at the base case in which the sub-problem is solved directly.
- The results of all sub-problems are then joined to produce the final overall solution.
- The sub-problems are usually solved independently and concurrently

Task Organization

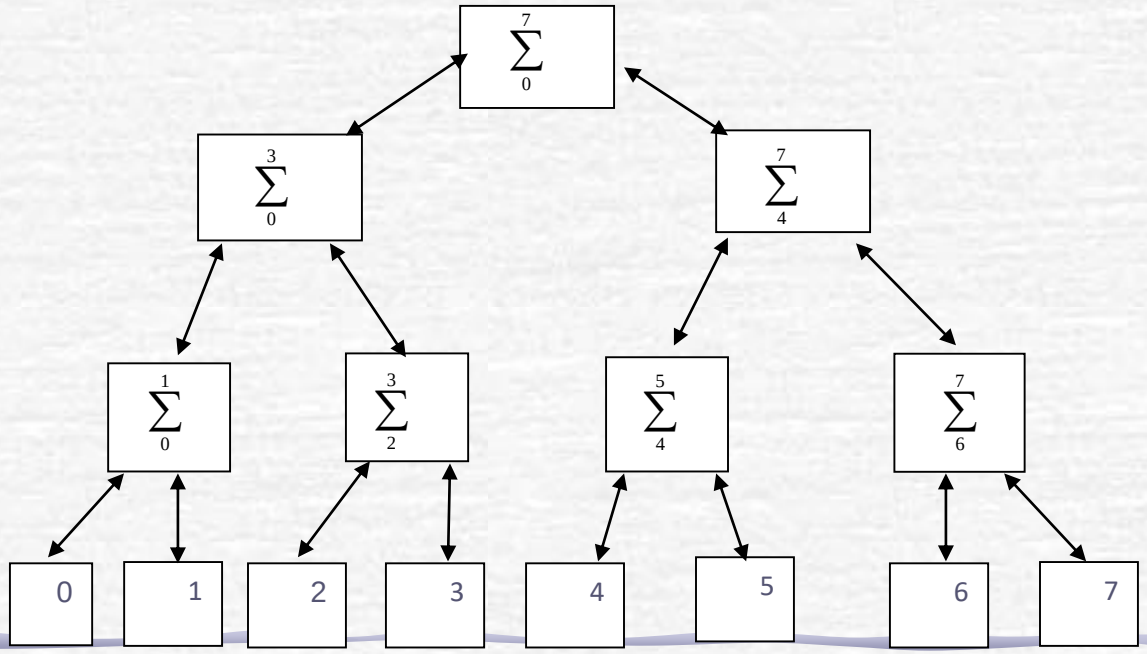


- The key point in Divide and Conquer is that the same task is recursively performed on different data.

Parallel Divide and Conquer Example

- Problem : Summing N numbers .
- Solution : Divide into two subproblems each of size N/2

  Using the feature $\sum_{i=0}^{2^n-1} = \sum_{i=0}^{2^{n-1}-1} + \sum_{i=2^{n-1}}^{2^n-1}$

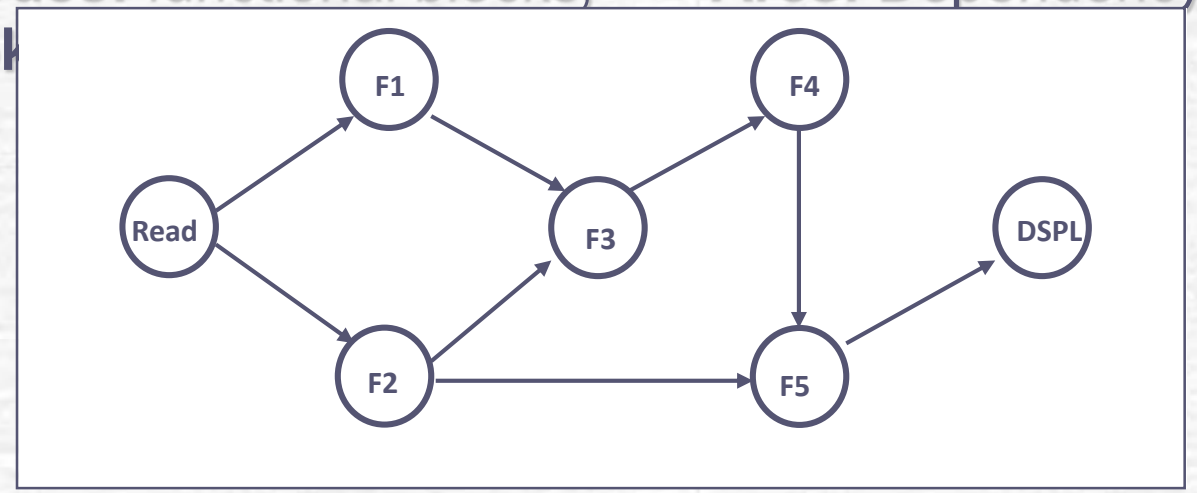


Dataflow

- **The main concept of dataflow programming is to divide the computation problem into multiple disjoint functional blocks.**
- **Each block solves part of the problem.**
- **Blocks are connected to each other to :**
  - show the dependency between them.
  - *express the logical execution flow*, and they can be used to easily express parallelism.
- **Data-pipelining is a specialized case of dataflow model.**

Principles of Dataflow Model

- **In dataflow model, computation is modeled as a directed graph.**
  - **Nodes:** functional blocks, **Arcs:** Dependency, **Tok**



There may be many nodes that are ready to fire (execute) at a given time.