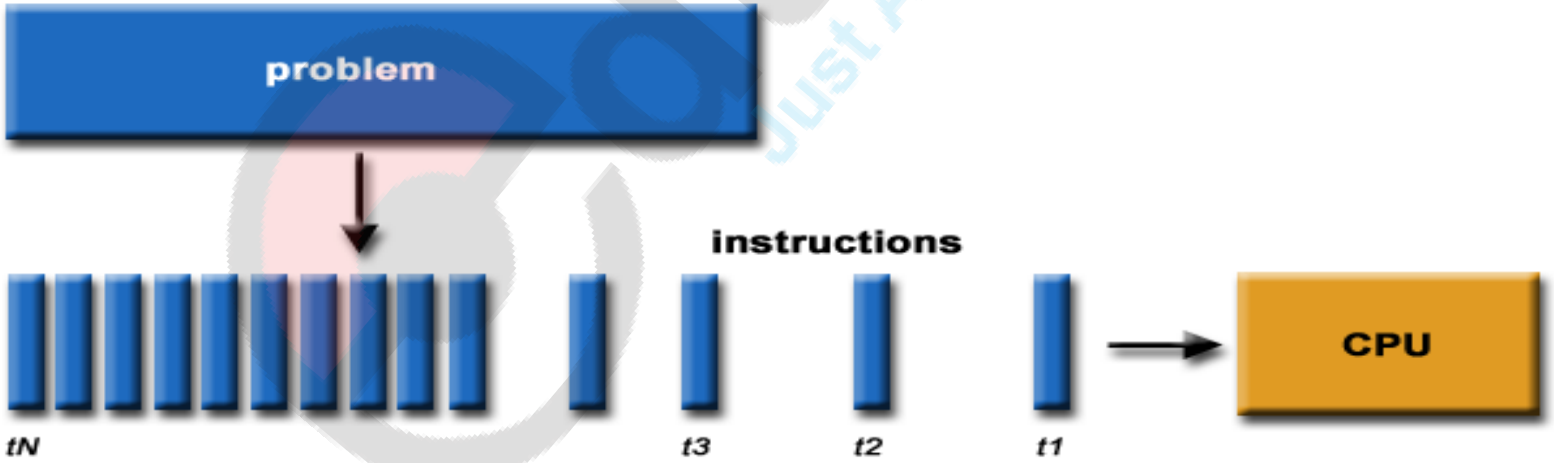
Parallel computing

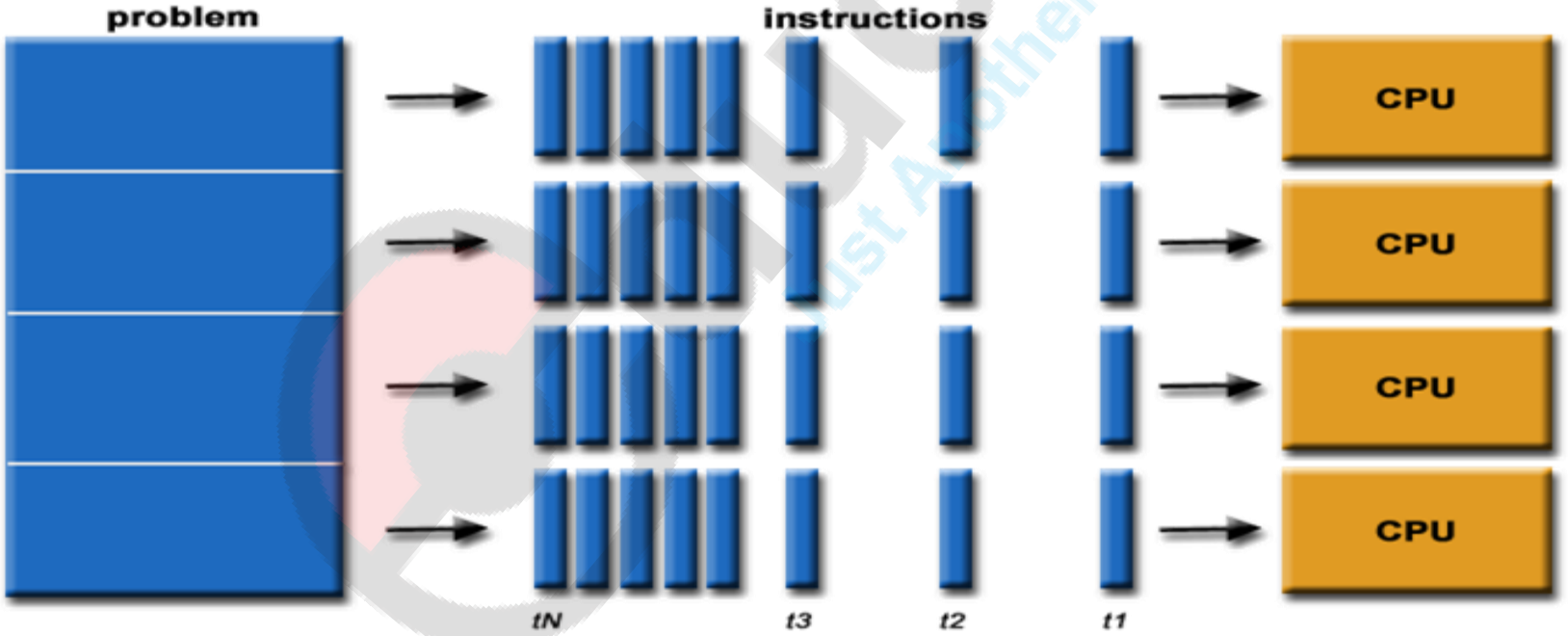
What is parallel computing?

- Traditionally, software has been written for ***serial*** computation:
  - To be run on a single computer having a single Central Processing Unit (CPU);
  - A problem is broken into a discrete series of instructions.
  - Instructions are executed one after another.
  - Only one instruction may execute at any moment in time.



Cont...

- In the simplest sense, ***parallel computing*** is the simultaneous use of multiple compute resources to solve a computational problem.
  - To be run using multiple CPUs
  - A problem is broken into discrete parts that can be solved concurrently
  - Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different CPUs



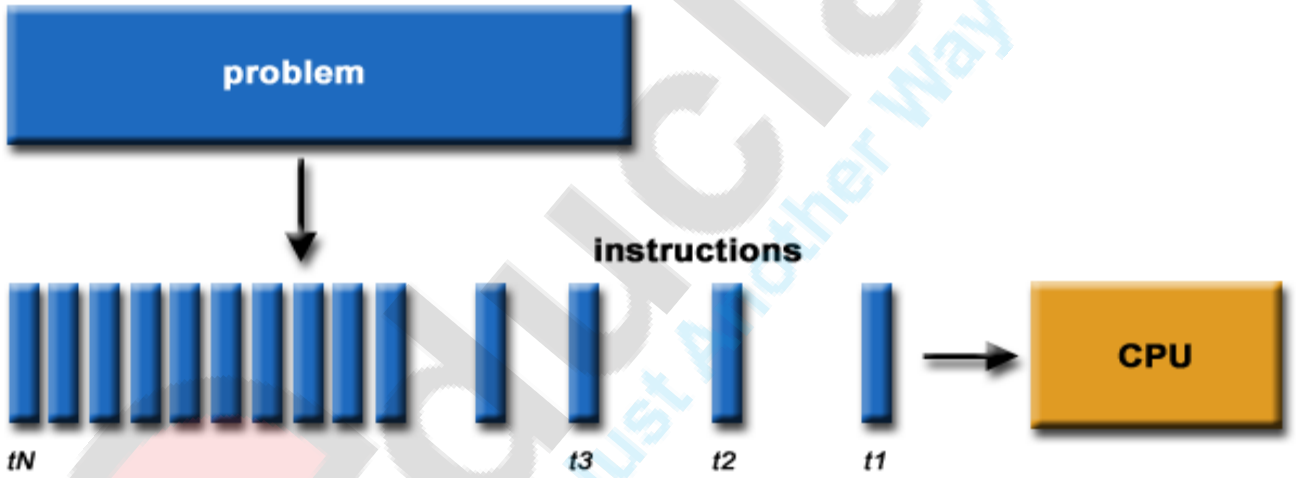
Resources required for Parallel Computing

- A single computer with multiple processors;
- A single computer with (multiple) processor(s) and some specialized computer resources (GPU, FPGA …)
- An arbitrary number of computers connected by a network

Parallel Computing : Scope

- Application in Engineering and Design
- Scientific Application
- Commercial Application
- Application for Computer System.

Serial Computation



Serial Computing Cont...

- Problem is broken down into a discrete series of instructions.
- Instruction are executed on a single processor sequentially.
- Only one instruction is executed at a time.
- Mainly used in monolithic application on single machine that do not have time constraints.

Advantages

- Fast execution of application with smaller tasks.
- Ease of implementation.
- Best suited for monolithic application.

Limitation

- Poses significant constraints to build faster serial computers.
- Hardware limitation on the transmission speed.
- Limits to miniaturization because of the processor technology.
- Expensive considering its performance.
- Processor in the serial computing consumes unacceptable power.

Parallel computing

- Speed up real time applications to achieve high performance.
- Provide a cost effective solutions by increasing the number of CPU's in a computer and by adding an efficient communication system between them.
- Workload can now be shared between different processors
- Results in much higher computing power and performance.
- Solves complex computation problems.

Advantages

- 1)Reduce time for task completion
- Solving large and complex real-world
- Provide concurrency
- Better resource utilization
- Fault tolerance

Disadvantage

- Require complex hardware
- Expensive than serial computing
- No tasks can be perfectly parallelizable, so shared resource have to be used serially.
- Task interdependencies must be consider before design.
- Communication overhead exists as multiple processors are involved in computing.

What is pipeline???

- The fundamental principle of pipelining is to increase the throughput or maximize the rate at which the instruction are executed.
- New inputs are accepted at one end before previously accepted input appears as output at other end.
- Organize concurrent activity in a computer system.

Implementation of Pipelining

- Synchronous pipeline: different subtasks are performed synchronously by different hardware blocks known as stage.
- Number of subtasks depend upon number of stages.
- The stages will perform different operations and result produced by each stage is temporarily buffered in latches and then passed on the next stage.
- Clocked latches are used to interface between stages as shown in the figure . After the arrival of clock pulse ,all the latches will transfer the data to the next stage simultaneously.
- All the stages are logical circuits having some delay.So the maximum delay time of stage will define the clock period and thus speed of the pipeline.

- The pipeline will be more efficient if the stages are divided in such a way that each stakes almost equal time to complete the subtask given to it.
- The clock period t of a pipeline is equal to sum of maximum stage delay time ts and the latch delay time d.
- T=max(ts)+d
- The pipeline frequency is defined as the inverse of the clock period.
- F=1/t.

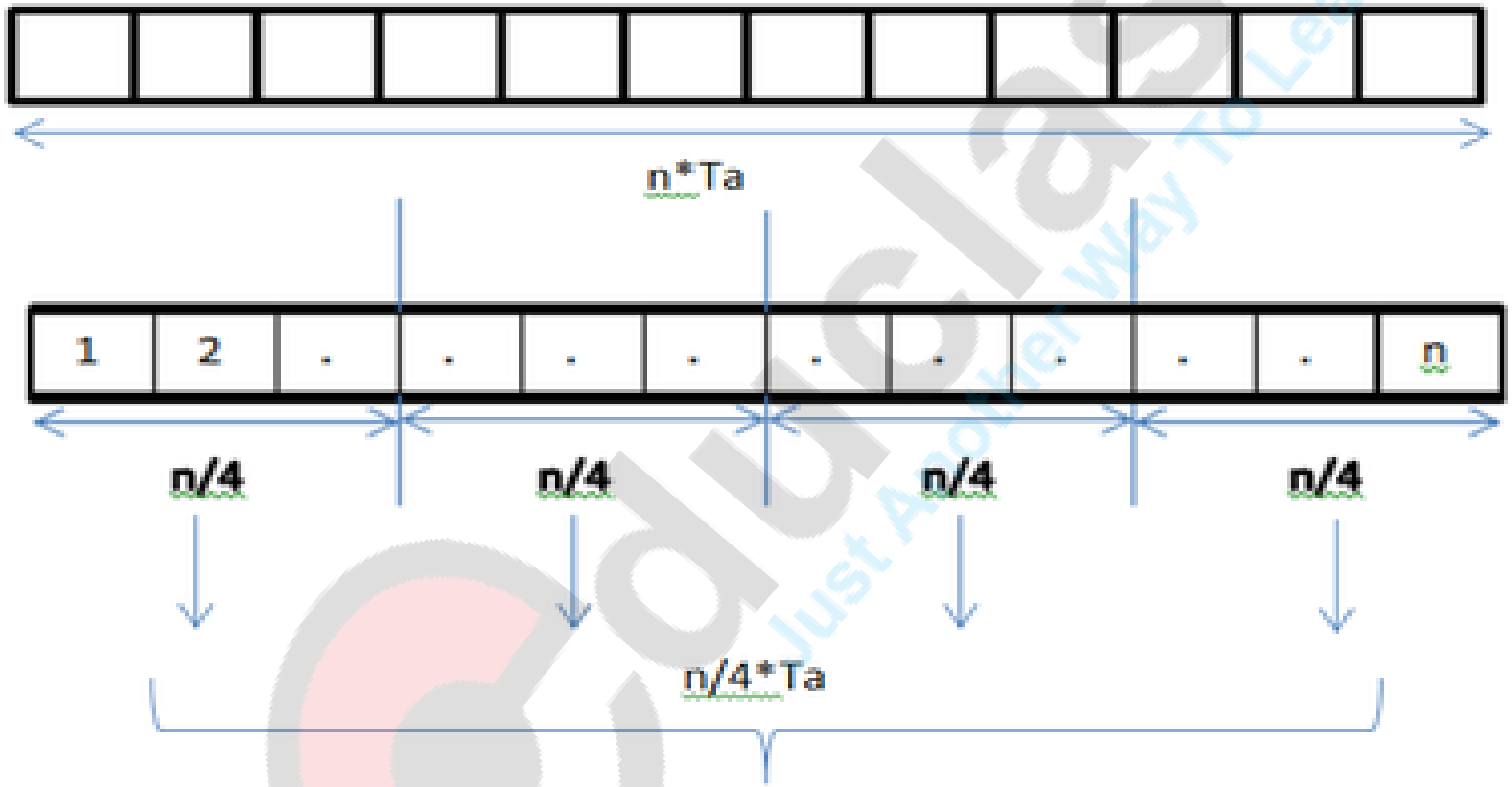
Asynchronous pipeline

- The k-stage asynchronous pipeline model is shown in the fig.The dataflow between adjacent stage in asynchronous pipeline is controlled by handshaking protocol.When a syage is ready to transmit the data,it sends ready signalto its next stage

Data Parallelism

- Parallelization across multiple processor in parallel computing environment.
- Can be applied on regular data structure like array and metrics
- Works on each elements in parallel manner.
- Let us take an example that we have “n” elements and the “ta” is the time take for single element operation if the system process serial execution then the execution sum of whole operation in serial computing would be (n*ta).

- Where as in case of parallel computing the count willbe ((n/4*)ta) as per the diagram shown below


n*Ta
1
2
.
.
.
.
.
.
.
.
.
n
n/4
n/4
n/4
n/4
n/4*Ta

Data parallelism	Task parallelism
Same operations are performed on different subsets of same data.	Different operations are performed on the same or different data.
Synchronous computation	Asynchronous computation
Speedup is more as there is only one execution thread operating on all sets of data.	Speedup is less as each processor will execute a different thread or process on the same or different set of data.
Amount of parallelization is proportional to the input data size.	Amount of parallelization is proportional to the number of independent tasks to be performed.
Designed for optimum load balance on multi processor system.	Load balancing depends on the availability of the hardware and scheduling algorithms like static and dynamic scheduling.

Control Parallelism

- Also known as task or function parallelism
- Multiple processor are involved
- Task parallelism focuses on distributing tasks—concurrently performed by processes or threads—across different processors
- In contrast to data parallelism which involves running the same task on different components of data, task parallelism is distinguished by running many different tasks at the same time on the same data.
- A common type of task parallelism is pipelining

Scalibility

- Increase number of processors --> decrease efficiency
- Increase problem size --> increase efficiency
- Can a parallel system keep efficiency by increasing the number of processors and the problem size simultaneously???

Yes: --> scalable parallel system

No: --> non-scalable parallel system

A scalable parallel system can always be made cost-optimal by adjusting the number of processors and the problem size.

Topologies

- Classification are as follows:

1)One to all

2)All to one

3)All to all

One to all and all to one

- 1)Ring
- 2)mesh
- 3)hypercube
- 4)balanced tree

All to all

- 1)Linear
- 2)Mesh
- 3)Hyper cube

Parallel algorithm models

- The Data-Parallel Model
- The Task Graph Model
- The Work Pool Model
- The Master-Slave Model
- The Pipeline or Producer-Consumer Model
- Hybrid Models

The Data-Parallel Model

- Simplest form of algorithm
- Task are statically mapped onto process
- Each task perform similar operations on different data.
- applied concurrently on different data items is called data parallelism.
- Work is done in phases
- Data operated upon the different phases may be different.

- the decomposition of the problem into tasks is usually based on data partitioning because a uniform partitioning of data followed by a static mapping is sufficient to guarantee load balance.
- Data parallel can be applied on both Shared address space and message passing.
- Interaction overheads in the data-parallel model can be minimized by choosing a locality preserving decomposition

Task Graph Model

- In the task graph model, the interrelationships among the tasks are utilized to promote locality or to reduce interaction costs.
- This model is typically employed to solve problems in which the amount of data associated with the tasks is large relative to the amount of computation associated with them.
- Ex:quicksort

Work Pool Model

- Dynamic mapping of tasks onto processes for load balancing in which any task may potentially be performed by any process.
- The mapping may be centralized or decentralized

Master Slave Model

- Also known as manager-worker model
- One or more manager generate work and allocate to the worker
- Task can be allocate prior if the manager estimate the size of the task.
- The manager-worker model can be generalized to the hierarchical or multi-level manager-worker model in which the top-level manager feeds large chunks of tasks to second-level managers

Pipeline Model

- Simultaneous execution of different programs on a data stream is called stream parallelism.
- It is a chain of producer and consumer
- Also known as producer and consumer.
- The pipeline does not need to be linear
- Usually involve static mapping of task.
- Load balancing i is the task granularity.

Hybrid Model

- More than one model is applicable to solve the problem
- Multiple model can be applied sequentially or hierarchically.

Classification parallel computing

- Distributed computing
- Cluster Computing
- Grid Computing

Parallel Architecture

- The main components are as follows:
- 1)Processors
- 2)Memory
- 3)Communication
- 4)Control

Issues in Parallel computing

- Design of parallel computing
- Design of efficient parallel algorithms
- Parallel Programming models
- Parallel computing language
- Methods for evaluating parallel algorithm.
- Parallel Programming tools.