- # Task assignment approach

A Distributed System is a Network of Machines that can exchange information with each other through Message-passing. It can be very useful as it helps in resource sharing. In this article, we will see the concept of the Task Assignment Approach in Distributed systems.

Resource Management:

One of the functions of system management in distributed systems is Resource Management. When a user requests the execution of the process, the resource manager performs the allocation of resources to the process submitted by the user for execution. In addition, the resource manager routes processes to appropriate nodes (processors) based on assignments.

Multiple resources are available in the distributed system so there is a need for system transparency for the user. There can be a logical or a physical resource in the system. For example, data files in sharing mode, Central Processing Unit (CPU), etc.

As the name implies, the task assignment approach is based on the division of the process into multiple tasks. These tasks are assigned to appropriate processors to improve performance and efficiency. This approach has a major setback in that it needs prior knowledge about the features of all the participating processes. Furthermore, it does not take into account the dynamically changing state of the system. This approach's major objective is to allocate tasks of a single process in the best possible manner as it is based on the division of tasks in a system. For that, there is a need to identify the optimal policy for its implementation.

Working of Task Assignment Approach:

In the working of the Task Assignment Approach, the following are the assumptions:

The division of an individual process into tasks.

- Each task's computing requirements and the performance in terms of the speed of each processor are known.

- The cost incurred in the processing of each task performed on every node of the system is known.
- The IPC (Inter-Process Communication) cost is known for every pair of tasks performed between nodes.
- Other limitations are also familiar, such as job resource requirements and available resources at each node, task priority connections, and so on.

Goals of Task Assignment Algorithms:

- Reducing Inter-Process Communication (IPC) Cost
- Quick Turnaround Time or Response Time for the whole process
- A high degree of Parallelism
- Utilization of System Resources in an effective manner

The above-mentioned goals time and again conflict. To exemplify, let us consider the goal-1 using which all the tasks of a process need to be allocated to a single node for reducing the Inter-Process Communication (IPC) Cost. If we consider goal-4 which is based on the efficient utilization of system resources that implies all the tasks of a process to be divided and processed by appropriate nodes in a system.

Note: The possible number of assignments of tasks to nodes:

For m tasks and n nodes= m x n

But in practice, the possible number of assignments of tasks to nodes < m x n because of the constraint for allocation of tasks to the appropriate nodes in a system due to their particular requirements like memory space, etc.

Need for Task Assignment in a Distributed System:

The need for task management in distributed systems was raised for achieving the set performance goals. For that optimal assignments should be carried out concerning cost and time functions such as task assignment to minimize the total execution and communication costs, completion task time, total cost of 3 (execution, communication, and interference), total execution and communication costs with the limit imposed on the

number of tasks assigned to each processor, and a weighted product of cost functions of total execution and communication costs and completion task time. All these factors are countable in task allocation and turn, resulting in the best outcome of the system.

Example of Task Assignment Approach:

Let us suppose, there are two nodes namely n1 and n2, and six tasks namely t1, t2, t3, t4, t5, and t6. The two task assignment parameters are:

- execution cost: x_{ab} refers to the cost of executing a task an on node b.
- inter-task communication cost: cij refers to inter-task communication cost between tasks i and j.

Tasks	t1	t2	t3	t4	t5	t6
t1	0	6	4	0	0	12
t2	6	0	8	12	3	0
t3	4	8	0	0	11	0
t4	0	12	0	0	5	0
t5	0	3	11	5	0	0
t6	12	0	0	0	0	0

Execution Cost		
Tasks	Nodes	
	n1	n2
t1	5	10
t2	2	infinity
t3	4	4
t4	6	3
t5	5	2
t6	infinity	4

Note: The execution of the task (t2) on the node (n2) and the execution of the task (t6) on the node (n1) is not possible as it can be seen from the above table of Execution costs that resources are not available.

Case1: Serial Assignment

Task	Node
t1	n1
t2	n1
t3	n1
t4	n2
t5	n2
t6	n2

Cost of Execution in Serial Assignment:

t11 + t21 + t31 + t42 + t52 + t62 = 5 + 2+ 4 + 3 + 2 + 4

= 20 (Refer Execution Cost table)

Cost of Communication in Serial Assignment:

= c14 + c15 + c16 + c24 + c25 + c26 + c34 + c35 + c36

= 0 + 0+ 12 + 12 + 3 + 0 + 0 + 11 + 0

= 38 (Refer Inter-task Communication Cost table)

Hence, Total Cost in Serial Assignment

= 20 + 38

= 58

Types of Task Assignment Approaches

Centralized Task Assignment Approach

The centralized task assignment approach is a method where there is a single point of control for the entire distributed system. In this approach, all the tasks are assigned from a central server, which allocates tasks to different nodes in the network.

The central server monitors the performance of each node and re−assigns tasks as needed. This approach requires that each node in the network communicates with the central server frequently to request task assignments or report on their current status.

One advantage of this approach is that it provides better control over task assignments and resource allocation, as all assignments are managed centrally. However, it also has some disadvantages such as high communication overhead since all systems communicate with a centralized entity which can increase latency and reduce response time especially if there is a large number of nodes in the system.

Decentralized Task Assignment Approach

The decentralized task assignment approach is a method where there is no central point of control in the distributed system. In this approach, every node in the network has equal responsibility for assigning and executing tasks. Each node decides what tasks to execute based on its current status and available resources without any interaction with other nodes or central servers.

The advantage of this approach is that it reduces communication overhead by eliminating frequent communications between nodes and central servers. It also provides better fault tolerance since if one node fails, other nodes in the system can continue working without disruption.

Factors Affecting Task Assignment Approach in Distributed Systems

Distributed systems are complex systems that operate in a network of interconnected computers. These systems are designed to handle a large amount of data and computation by distributing the tasks across multiple machines.

The task assignment approach plays a crucial role in the efficient operation of these distributed systems. Here, we discuss the factors that affect the task assignment approach in distributed systems.

Network Latency: The Barrier to Efficient Task Assignment

Network latency refers to how long it takes for data to travel from one point on a network to another. It is one of the primary factors affecting task assignment approaches in distributed systems.

High network latency can significantly slow down the process of task execution. For instance, if data has to be shuffled between different nodes frequently, it can cause significant delays and affect overall system performance.

A practical solution to address network latency is to employ techniques like caching or replication so that critical data is available locally for faster access. Another option is using algorithms that consider network latency as a factor while assigning tasks so that tasks are assigned closer together geographically where possible.

Load Balancing: The Challenge of Distributing Workload Equitably

In distributed computing, load balancing refers to distributing workloads evenly among different nodes for better utilization of resources and efficient task execution. In other words, load balancing ensures that no single node is overloaded with more tasks than it can handle while others remain underutilized.

The challenge with load balancing lies in identifying how much workload each node can handle, especially when dealing with heterogeneous infrastructure with varying capabilities such as CPU power or memory capacity. To address this challenge, several algorithms have been developed such as round−robin or least−loaded which distribute workload evenly among available nodes based on their capacity for handling tasks.

Resource Availability: Ensuring Adequate Resources for Task Execution

The availability of resources like CPU, memory, or storage is another factor affecting the task assignment approach in distributed systems. Inadequate resources can cause delays or system crashes if a task requires more resources than available on a node. For example, if a node running a task runs out of memory, the task cannot be completed.

To prevent such issues, task assignment algorithms must consider resource availability and allocate tasks only to machines with adequate resources to complete them. Additionally, monitoring tools can be used to track resource utilization and identify overutilized nodes that may need additional support or maintenance.

Network latency, load balancing and resource availability are critical factors affecting the

performance of distributed systems. To ensure efficient execution of tasks in these systems, it is necessary to employ algorithms that consider these factors while assigning tasks among multiple available nodes.

Algorithms for Task Assignment in Distributed Systems

Round Robin Algorithm

The Round Robin Algorithm is a popular task assignment approach used in distributed systems. It involves assigning tasks to nodes in a circular manner, with each node receiving an equal share of tasks.

The algorithm is simple and easy to implement, making it a preferred choice for many applications. In this approach, the system assigns tasks to the first available node, and then moves on to the next node in the list.

Least Loaded Algorithm

Another popular task assignment approach for distributed systems is Least Loaded Algorithm. This approach assigns new tasks to the least loaded node in the network at any given time. In other words, it selects a node that currently has fewer assigned tasks than others.

The Least Loaded Algorithm also helps maintain balanced workload distribution across all available resources and reduces processing delays caused by overburdened resources. One advantage of using this algorithm is that it automatically adjusts to changes in resource availability and processing capabilities by dynamically reassigning tasks as needed.