Name: Shivam Pandey	
Roll No.: 28	
Experiment No.8	
Implement Task Assignment	
Date of Performance:27/03/2024	
Date of Submission:03/04/2024	

Aim: To implement task assignment approach

Objective: Implement task assignment approach

Theory:

In task assignment approach, process is viewed as a collection of related tasks and these tasks are scheduled to suitable nodes so as to improve performance.

A process is split into pieces called tasks. This split occurs along natural boundaries, so that data transfers among the tasks is minimized. The amount of computation required by each task and the speed of each processor are known. The cost of processing each task on every node of the system is also known. The inter-process communication (IPC) costs between every pair of tasks are known.
Other constraints, such as resource requirements of the tasks and the available resources at each node, precedence relationships among the tasks, and so on, are also known.

Consider the following example:

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

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

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

The serial assignment cost can be calculated as follows:

Serial assignment execution cost $(x) = x_{11} + x_{21} + x_{31} + x_{42} + x_{52} + x_{62}$
$= 5 + 2 + 4 + 3 + 2 + 4 = 20$

Serial assignment communication cost $(c) = c_{14} + c_{15} + c_{16} + c_{24} + c_{25} + c_{26} + c_{34} + c_{35} + c_{36}$
$= 0 + 0 + 12 + 12 + 3 + 0 + 0 + 11 + 0 = 38$

Serial assignment total cost $= x + c = 20 + 38 = 58$

And the optimal assignment cost can be calculated as follows:

Optimal assignment execution cost $(x) = x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{62}$
$= 5 + 2 + 4 + 6 + 5 + 4 = 26$

Optimal assignment communication cost $(c) = c_{16} + c_{26} + c_{36} + c_{46} + c_{56}$
$= 12 + 0 + 0 + 0 + 0 = 12$

Optimal assignment total cost $= x + c = 26 + 12 = 38$

For implementing task assignment approach, take from the user all the parameters and then calculate the cost.

Code and output:

```
#include <iostream>
#include <vector>
#include <climits>

int execCost[6][2] = {
    {5, 10},
    {2, INT_MAX},  // INT_MAX to represent infinity where execution on this node is
not possible
    {4, 4},
    {6, 3},
    {5, 2},
    {INT_MAX, 4}
};

int commCost[6][6] = {
    {0, 6, 4, 0, 0, 12},
    {6, 0, 8, 12, 3, 0},
    {4, 8, 0, 0, 11, 0},
    {0, 12, 0, 0, 5, 0},
    {0, 3, 11, 5, 0, 0},
    {12, 0, 0, 0, 0, 0}
};
// Function to calculate the total cost for a given assignment
int calculateTotalCost(const std::vector<int>& assignment) {
    int executionCost = 0;
    int communicationCost = 0;

    // Calculate execution cost
    for (int i = 0; i < 6; ++i) {
        if (execCost[i][assignment[i]] == INT_MAX) {
            executionCost += 1000; // Assume a high cost for infeasible operations, should
not actually occur in optimal settings
        } else {
            executionCost += execCost[i][assignment[i]];
        }
    }

    // Calculate communication cost
    for (int i = 0; i < 6; ++i) {
        for (int j = i + 1; j < 6; ++j) {
            if (assignment[i] != assignment[j]) {
                communicationCost += commCost[i][j];
            }
        }
    }
    return executionCost + communicationCost;}
```

```
int main() {
    // Define serial and optimal assignments based on the corrected understanding
    std::vector<int> serialAssignment = {0, 0, 0, 1, 1, 1};   // Serial assignment configuration
    std::vector<int> optimalAssignment = {0, 0, 0, 0, 0, 1}; // Optimal assignment configuration

    // Calculate total costs
    int totalCostSerial = calculateTotalCost(serialAssignment);
    int totalCostOptimal = calculateTotalCost(optimalAssignment);

    // Output the results
    std::cout << "Total Cost in Serial Assignment: " << totalCostSerial << std::endl;
    std::cout << "Total Cost in Optimal Assignment: " << totalCostOptimal << std::endl;

    return 0;
}
```

Conclusion:
In this exploration, we have demonstrated the task assignment approach in a distributed system by calculating the total execution and communication costs for both serial and optimal task assignments. Program utilized given execution and communication costs matrices. The serial assignment attempted to minimize the execution cost by strategically placing certain tasks on specific nodes but lacked the holistic approach seen in the optimal assignment.

In the optimal assignment, tasks were distributed across nodes in a manner that both minimized execution cost and significantly reduced communication costs, showcasing how careful task distribution can enhance overall system performance and efficiency. This approach highlighted the critical importance of considering both inter-process communication and execution costs in distributed systems to achieve the most cost-effective and efficient configuration.