ASSIGNMENT OF TASKS ON PARALLEL AND DISTRIBUTED COMPUTER SYSTEMS WITH NETWORK CONTENTION

[1]Hussein EL GHOR, [2]Rafic HAGE CHEHADE

1Lebanese University, IUT Saida, GRIT Department, B.P. 813-Saida, Lebanon

[2] Lebanese University, IUT Saida, GRIT Department, B.P. 813-Saida, Lebanon

E-mail: elghorh@ul.edu.lb, rafichajj@hotmail.com

ABSTRACT

Rapid advances in communications technology and the proliferation of inexpensive PCs and workstations have created a wide avenue for *Distributed Computing Systems* to move into mainstream computing. A Distributed Computing System (DCS) consists of a number of PCs or workstations interconnected through PPP, LAN or WAN. These systems provide a higher performance, better reliability and throughput over centralized mainframe systems.

Clearly, we have a set of M tasks connected in some fashion and a heterogeneous DCS composed of N computers of different capabilities. Tasks of a given application require certain computer resources (memory, processor and communication link). Indeed, computers and communication resources in the system are also capacitated. For these reasons, the issue is how to assign (*allocate/schedule*) the tasks of a given application onto the available computers of the system so as to maximize the system throughput i.e. to minimize the total sum of execution and communication costs.

Keywords: Distributed Systems (DS), Task Allocation, Task Scheduling, Link Contention, Simulated Annealing (SA), Branch and Bound (BB).

1. INTRODUCTION

Applications In this article, we will add an important factor which is the amount of contention on communication operations. The challenge here is to make an algorithm for the allocator/ scheduler of tasks onto machines in a heterogeneous environment taking into consideration the account of network contention.

Many applications would normally take a long time to finish execution on one machine. If these applications could be divided into a number of tasks and executed concurrently on different machines, a tremendous improvement in the performance would occur. But two main problems face this operation, first one include partitioning the application into tasks referred to “task partitioning problem” and the second include assigning the tasks onto computers in the system referred to “task assignment problem”.

As wrote before, the problem here is concerned with allocating and scheduling tasks of a parallel application among computing sites of a DCS. The DCS consists of N computers interconnected through PPP, LAN or WAN. Each computer has its own processor and memory. The interconnection network has a communication capacity and propagation delay. On the other hand, the parallel application is represented by as a collection of M tasks which corresponds to nodes in the graph while the arcs of the graph may represent communication between tasks. We have 2 kinds of graph to represent the problem; directed and undirected graphs G(V,E), where V represents a set of M tasks and E represents a set of edges.

The goal from this process is to find the best location for the application tasks and the order of their execution so as to minimize the schedule length and realize the task dependency.

.

2. SOLUTION STRATEGY

The solution strategy is characterized by the following three phases:

1) In the first phase, some INPUT values must be gathered. The task graph analyses the application to acquire the task attributes (execution time, memory requirements and processing load) and communication edges (data to be communicated network contention and bandwidth requirements). On the other hand, the Distributed Computing

System (DCS) provides information about Computers (processing load and available memory) and Network (unit communication cost and resources capacities).

2) In the second phase, the process of task allocating / scheduling the input values from first phase. It then attempts to map the tasks onto the processors using some optimization criterion (placement constraints, processing load constraints, memory capacity constraints and communication capacity constraints). Finally, it distributes the application tasks onto the available computers in the system by applying one of the proposed algorithms (Exact, Heuristic and Hybrid)

3) In the third phase, we must get the final results on the OUTPUT in one of two cases: in case of TIG or TFG, for each task i, the allocator finds the best location of i, locate(i). In case of TPG or DAG, for each task i, the scheduler finds locate(i), start(i) and finish(i). [1].

3. COST FUNCTIONS

The assignment problem is usually handled based on the optimization of a cost function.
Depending on this context, many components of the cost function may be defined.

- **Accumulative Execution Time**

The processing cost / load at a processor p (EXECp) is the total execution time incurred by tasks running at processor p. Let TCp be the set of tasks such that:

$$TCp = \{i \mid Xip = 1, 1 \leq i \leq M, 1 \leq p \leq N\}$$

Let Cip denotes the cost of processing a task i, then the actual execution cost EXECp will be formulated as:

$$EXECp = \sum_{i \in TCp} Cip = \sum_{i \in TCp} di * ep$$

Where di is the size of the task i, and ep is the average processing time of one instruction on the processor p.

- **Accumulative Communication Time:**

Actual Communication Cost at a processor p is the total time of communicating data between tasks at the processor p with other tasks at processor q. Let Cijpq be the cost of sending data between a task i at the processor p and a task j at processor q, dij is the average quantity of data to be transferred between p and q;

$$COMMp = \sum_{q \neq p} \sum_{i \in TCp} \sum_{j \neq i, j \in TCq} TCq * Cijpq$$

Here, we must take into consideration the amount of loss in network communication that can occur because of several factors like type of media and data.

$$COMMp = \sum_{q \neq p} \sum_{i \in TCp} \sum_{j \neq i, j \in TCq} (Spq + dij * (Cpq - Lpq))$$

Where Spq time necessary for p to set communication with q, Cpq is the average of transferring a data unit between p and q and Lpq is the loss of data in communication link between p and q.

The Accumulative Communication Time (ACT) is the total time for exchanging data between tasks residing at separate computers.

$$ACT = \sum_{p} \sum_{p \neq q} \sum_{i} \sum_{j \neq i} Cijpq\ Xip\ Xjq$$

Allocation Model for Throughput Define Al and Cl as the available communication capacity and the cost of transferring a data unit respectively. Let Yl f be the value of flow through the link l under the flow f. Then the assignment problem will be:

$$\min \sum_{p} \sum_{i} Cip\ Xip + \sum_{f} \sum_{l} C_l\ Y_{lf}$$

Where it is subjected to the following constraints (task redundancy, memory, processing load and communication load where network considered as an important factor).

4. OPTIMAL TASK ASSIGNMENT

Now, we must solve the problem by finding an optimal solution by using the *Hybrid Approach*. But, optimal solutions to the task assignment problem may be found through exhaustive enumeration of all the possible elements plus their cost; this can take a lot of time and a lot of memory. In addition, optimal solution to the task assignment problem is known to be NP-hard.

We are now able to find the optimal solution of the assignment problem by combining the advantage of both the *SA algorithm* and the *Modified BB technique*. This is because we can find the optimal solution with Exact algorithm but it will take a high computation time. On the other hand, we can find quickly a suboptimal solution using SA algorithm but this solution could be far from the optimal one.

To benefit from both of the algorithms above, one can find an initial solution rapidly by using the Heuristic method and then try to improve it by using the Exact method. As a result, two phases are needed to find an optimal solution: *First* phase is the suboptimal one. In it, we quickly find a suboptimal solution by using the Heuristic Method. The efficiency of the hybrid method depends on the initial solution's quality. As the initial solution nears the optimal the computational time of the second phase decreases. *Second* phase is the optimal one. In it, we consider the solution obtained from the first phase as an initial solution. The original minimization problem is converted into a maximization problem by m using duality; the dual problem is then solved by constructing and traversing a search tree considering the cost of the first solution as a lower bound. During the optimal solution search, the algorithm prunes all assignments with costs lower than the first candidate cost.

5. EXPERIMENTS AND EVALUATION

We are now able to find the optimal solution of the assignment problem by combining the advantage of both the SA algorithm and the Modified BB technique. This is because we can find the optimal solution with Exact algorithm but it will take a high computation time. On the other hand, we can find quickly a suboptimal solution using SA algorithm but this solution could be far from the optimal one.

To benefit from both of the algorithms above, one can find an initial solution rapidly by using the Heuristic method and then try to improve it by using the Exact method. As a result, two phases are needed to find an optimal solution: First phase is the suboptimal one. In it, we quickly find a suboptimal solution by using the Heuristic Method. The efficiency of the hybrid method depends on the initial solution's quality. As the initial solution nears the optimal the computational time of the second phase decreases. Second phase is the optimal one. In it, we consider the solution obtained from the first phase as an initial solution. The original minimization problem is converted into a maximization problem by m using duality; the dual problem is then solved by constructing and traversing a search tree considering the cost of the first solution as a lower bound. During the optimal solution search, the algorithm prunes all assignments with costs lower than the first candidate cost.

The above description is very clear and correct in the theoretical part of view, but it still need to do some experiments to ensure that the approach lead to the optimal assignment. For that reason, we must study the cost function of the Hybrid approach (SABB) against the (BB) technique. In this study, a large number of randomly generated task graphs will be allocated onto distributed systems of different topologies are considered. Note that the same procedure will be done as in the previous sections.

The goal is to minimize the total sum of execution and communication costs that may be incurred under any assignment. The system configuration is restricted to LAN of bus topology. First of all, we must present the linear model of such assignment.

$$\min \textstyle\sum_p \sum_t Ctp\, Xtp + \sum_f \sum_l C_l\, Y_{lf}$$

Where:

$$\textstyle\sum_p X_{tp} = 1 \qquad \forall\, tasks\ t$$

$$\textstyle\sum_t p_t X_{tp} \le P_p \qquad \forall\, computers\ p$$

$$\textstyle\sum_t m_t X_{tp} \le M_p \qquad \forall\, computers\ p$$

$$\textstyle\sum_f Y_{lf} \le A_l \qquad \forall\, virtual\ channels\ l$$

$$\textstyle\sum_{lin} Y_{lf} - \sum_{lout} Y_{lf} - Lpq + d_f X_{tp} - d_f X_{fp} - df\,Lpq = 0 \qquad \forall\, flows\ f, nodes\ p$$

The processor graph is constructed, in this paper, by using virtual channel representation between every pairs of computers in the system. In the graph, nodes represent processors while edges represent communication channels between them. Each node has the memory size Mp, the computation load Pp and the failure rate A, of the corresponding processor p, while each channel 1 has the communication capacity Al, the communication cost Cl and the failure rate pl of the corresponding path pq. In the network flow, it is often desirable to transport the maximum amount of flow from a starting point (called source) to a terminal point (called sink) with minimal cost. For a flow to deal with the loss in network communication due to network contention, it must have two characteristics:

$$0 \ll flow\ in\ each\ arc \ll arc\ capacity$$

$$flow\ into\ node\ k = flow\ out\ of\ node\ k - loss\ in\ network\ communication$$

This part shows the throughput of the hybrid algorithm (SABB) against the well known Branch-and-Bound (BB) algorithm. The two algorithms are coded in Matlab and tested for a large number of randomly generated task graphs that being allocated onto a distributed computing system. The simulation program contains two major parts.

The first part reads as input the number of tasks and the number of processors. It then generates a task graph and equivalent parameters. It also generates the system parameters considering a particular topology of a distributed system. In this paper, the system configuration is restricted to LAN of bus topology (case a) and to fully connected topology (case b).

For generating the parameters, the program uses the following test data: The failure rates of processors and communication links are given in the ranges [0.0005-0.000l0] and [0.00015-0.00050] respectively. The costs of processing tasks at different processors are given in the range [15- 25]. The memory requirements of each task are given in the range [l-l0]. The value of data to be communicated between tasks is given in the range [5-l0]. The average number of neighbors to a task is 3. The second part of the simulation program applies each of the algorithms to find an optimal allocation and the associated system reliability.

Results below shows the simulation for allocating tasks onto a distributed systems of bus topology (case a) and to fully connected topology (case b).

The results show that, at a given number of tasks, the average number of generated nodes when using the SABB algorithm is lower than that of using the BB algorithm.

This is because, many branches of the search tree may be pruned as a results of the good starting point of the second phase of the hybrid algorithm.
Also, the average computation time of the SABB algorithm is lower than that of the BB algorithm. This is because, using the SABB algorithm, lower number of nodes may be explored.

This result leads when using the SABB algorithm, the total sum of execution and communication costs will be lower which is a good solution to be published.

6. CONCLUSION

To make a summary of all the work to perform the optimal approach, it has been shown that the standard BB approach leads to an optimal assignment but needs a lot of time and memory, the modifications done to the BB algorithm improves its performance.
While the SA algorithm finds and in a short time a suboptimal task assignment but it can be far from the optimal one. Also, the SA algorithm works efficiently with a high number of nodes. Therefore, the solution compatible for our problem is by using the two algorithms by starting with the solution obtained by the SA algorithm as an initial solution and try to improve it by using the BB approach. This work is done taking into consideration the amount of loss in network communication. As the initial solution approaches the optimal solution the running time of the algorithm decreases. Hence, in developing such hybrid approach, the heuristic technique of the first phase should be done carefully.

7. REFERENCES

[1] Gamal Attiya and Yskandar Hamam, "Static Task Assignment in Distributed Computing Systems ", *21st IFIP tc7 Conference on System Modeling and Optimization*, Sphia Antipolis, Nice, France, 2003.

[2] C.-H Lee, D. Lee and M. Kim, "Optimal Task Assignment in Linear Array Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 8, No. 2, pp. 119-129, Feb 1997.

[3] M. -S. Chern, G.H. Chen and P. Liu, "An LC Branch-and-Bound Algorithm for The Module Assignment Problem", *Information Processing Letters*, Vol. 32, pp. 61-7, 1989.

[4] P.-Y. Chang, D.-Y Chen and K.-M. Kavi, "File Allocation Algorithms to Minimize Data Transmission time in Distributed Computing Systems", *J. of Information Science and Engineering*, vol. 17, pp. 633-646, 2001.

[5] J.B. Sinclair, "Efficient Computation of Optimal Assignments for Distributed Tasks", *Journal of Parallel and Distributed Computing*, vol. 4, pp. 342-362, 1987.

[6] M. Kafil and I. Ahmad, "Optimal Task Assignment in Heterogeneous Distributed Computing Systems", *IEEE Concurrency*, vol. 6, No. 3, pp. 42-51, July-Sept. 1998.

[7] K. Efe, “Heuristic Models of Task Assignment Scheduling in Distributed Systems”, *IEEE Computer*, vol. 15, pp. 50-56, June 1982.

[8] K. Taura and A. Chien, “A Heuristic Algorithm for Mapping Communicating Tasks on Heterogeneous Resources”, *Proceeding of the 9th Heterogeneous Computing Workshop*, pp. 102-115, Cancun, Mexico, May 2000.

[9] Y. Hamam and K.S. Hindi, “Assignment of Program modules to Processors: A Simulated Annealing Approach”, *European Journal of Operational Research*, vol. 122, pp. 509-513, 2000.

[10] A. L. Corcoran and D. A. Schoenefeld, “A Genetic Algorithm for File and Task Placement in Distributed Systems”, *Proceedings of IEEE Conference on Evolutionary Computation*, Orlando, Florida, June 1994.

[11] H. -A. Choi, S. Daknis, B. Narahari and R. Simha, “Algorithms for Mapping Task Graphs to a Network of Heterogeneous Workstations”, *International Conference on High Performance Computing, Chennai*, India, 1997.

[12] Y. Kopidakis, M. Lamari, V. Zissimopoulos, “On the Task Assignment Problem: Two New Efficient Algorithms”, *Journal of Parallel and Distributed Computing*, vol. 42, pp. 21-29, 1997.

[13] A. Radulesco and A. J. C. Gemund, “Low-Cost Task Scheduling for Distributed Memory Machines”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, No. 6 pp. 648-658, 2002.

[14] Gamal Attiya and Yskandar Hamam, “Assignment of Tasks onto Heterogeneous Computers: A Branch-and-Bound Technique”, *Journal of Computers and Operational Research*, 2004.

[15] D. S. Johnson, C. R. Aragon, L. A. McGeogh and C. Schevon, “Optimized by Simulated Annealing: An Experiment Evaluation”, *AT&T Labs*, Murray Hill, NJ, Preprint 1987.

[16] T. -Y. Wang and K.-B. Wu “A parameter Set Design Procedure for the Simulated Annealing Algorithm under the Computational Time Constraint”, *Computers and Operations Research*, vol. 26, No. 7, pp. 665-678, July 1999.

[17] I. Ahmad, Yu-Kwong Kwok, Min-You Wu and Wei Shu, ”Link Contention- Constrained Scheduling and Mapping of Tasks and Messages to a Network of Heterogeneous Processors”, *IEEE*, pp. 113-124, December 2000.

[18] R. Perego and G. De Petris “Minimizing Network Contention for Mapping Tasks onto Massively Parallel Computers” *IEEE*, 1995.

[19] YU-KWONG KWOK, ISHFAQ AHMAD “Link Contention-Constrained Scheduling and Mapping of Tasks and Messages to a Network of Heterogeneous Processors” *Springer*, 2000.

[20] Oliver Sinnen and Leonel A. Sousa “Communication Contention in Task Scheduling” *IEEE Transactions on Parallel and Distributed Systems*” Vol. 16, No. 6, June 2005.