iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** X **Month of publication:** October 2018

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue X, Oct 2018- Available at www.ijraset.com

An Enhanced Kuhn Munkres Algorithm for Improving Computational Performance

S. U. Nivethitha[1], P. Shanthi[2]
[1]Research Scholar, Department of Computer Science, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-641005, India
[2]Assistant Professor, Department of Information Technology, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-641005, India

Abstract: Fog computing (FC) and edge computing can serve as the middle layer of end user devices and the powerful cloud. As an execution of subtype of FC, mobile cloud computing (MCC) objectives at leveraging limited resources available at network edge to enrich mobile applications and promote end user's experience. So, an efficient resource allocation or scheduling technique is very important to ensure the effectiveness of MCC. The dynamic tasks scheduling mechanism using weighted bi-graph model have been presented for ensuring the performance of scheduling technique. This model was taken the dynamics of both tasks and providers into consideration. The scheduling problem was translated to be a maximum weighted bi-graph matching problem and an integer programming model was formulated. The matching issue was solved through this model that consists of state information collection of offloaded tasks and service providers, mapping relationship establishment, profit matrix determination and optimal matching using Kuhn Munkres (KM). However, this model has low computational performance. Hence, in this paper, an enhanced the KM algorithm is proposed by using the sparsity based KM and parallel based KM for improving the computational performance. The proposed approach is used for solving assignment problem with considerable accuracy loss. The experimental results show that the proposed approach provides better results in terms of total profit and guarantee ratio.
Keywords: Kuhn Munkres, Sparsity based KM, Parallel based KM, Guarantee ratio, Total profit

I. INTRODUCTION

As the development of cloud computing (CC) and mobile internet, the wireless devices are become more and more intelligent. Rich Mobile Applications (RMAs) [1], [2] is provided richer experience for people and enhance the quality of people's life. Because of the limited resources and energy supply, single mobile device cannot meet the entire needs of RMAs. CC is put forward as a novel available paradigm that can enable the RMAs to be done with the help of Cloud [3]. The mobile cloud or fog systems are for using available resources to handle offloaded tasks.

To decrease the computation trouble, a dynamic tasks scheduling algorithm using weighted bi-graph model [4] was developed. The scheduling issue was translated into a maximum weighted bi-graph matching problem. An integer programming model was formulated by using the matching issue. The task scheduling procedure consists of four phases: information collection of offloaded tasks and service providers, establishment of mapping relationship, determination of weight matrix and generation of optimal matching approach using KM algorithm. But, computational performance is low in this approach. In this paper, to enhance the computational performance, an enhanced KM algorithm based on the sparsity based KM and parallel based KM is proposed. This approach is used to solve the assignment problem with considerable accuracy loss.

The remainder of the article is organized as follows: Section 2 describes about the scheduling techniques. Section 3 describes about the proposed methods. Section 4 illustrates the performance evaluation of the proposed techniques. Section 5 concludes the research work.

II. RELATED WORK

Taxonomy [5] was presented for scheduling in distributed systems and a taxonomy extension was proposed for covering cloud computing schedulers. The literature has been classified in this taxonomy. Relevant future directions were identified to schedule in distributed systems.

The grouped tasks scheduling technique [6] was discussed that was utilized for scheduling tasks in cloud computing network by applying quality of service. This technique was combined many types of task attributes such as task priority, latency of task, task size, user type that were utilized for measuring priority of tasks.

A Discrete Symbiotic Organism Search method [7] was designed for optimal scheduling of tasks on cloud resources. Symbiotic organism search was a newly developed metaheuristic optimization method to solve numerical optimization issues. This method was imitated the symbiotic relationships shown using organisms in an ecosystem. This approach was converged faster once the search was got larger that was made it suitable for large-scale scheduling issues.

A real-time dynamic scheduling method [8] was developed for implementing task-based applications on distributed computing platforms to decrease the energy consumption. A polynomial-time mechanism was presented that was combined a set of heuristic rules and a resource allocation approach for getting best solutions on an affordable time scale. This approach was minimized a multi-objective function.

The task scheduling method [9] was explored by a hybrid technique that was combined desirable characteristics of two of the most widely utilized biologically-inspired heuristic techniques, the genetic algorithms and the bacterial foraging methods in the computing cloud. Initially, the scheduling technique was used to minimize the make span and it was decreased the energy consumption, both economic and ecological perspectives.

Mathematical model [10] was designed by Load balancing mutation a particle swarm optimization based schedule and allocation for cloud computing which was taken into account reliability, transmission time, make span, execution time, transmission cost, round trip time and load balancing among virtual machine and tasks. This approach was taken the load balancing once distributing tasks to available resources, tasks assign as earlier as possible, finished as earlier as possible and reschedule failure tasks. It was utilized for any number of tasks and resources.

III.PROPOSED METHODOLOGY

In this section, sparsity based Kuhn-Munkres (sKM) and parallel KM (pKM) by using original KM algorithm are described for improving the computational performance.

A. Assignment Problem

Assume that there are n tasks to be completed with m mobile devices available. Without loss of generality, consider $n \leq m$. Any mobile devices might be assigned for performing one of these tasks, every pair (task, mobile device) is a time cost for that to perform the task. The objective is to done the entire tasks while reducing total time cost, while assigning every mobile device to exactly one task and vice versa. The assignment issue is formulated to the following optimization issue:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} * a_{ij},$$
$$s.t \sum_{i=1}^{n} a_{ij} = 1, \quad 0 \leq \sum_{j=1}^{m} a_{ij} \leq 1, \ a_{ij} \in \{0,1\} \tag{1}$$

In the above equation, $C \in R^{n \times m}$ denotes the cost matrix, c_{ij} represents the cost of mobile device i to perform task j, $A \in R^{n \times m}$ denotes the resulting binary matrix, $a_{ij} = 1$ if and only if the $i-th$ mobile device is assigned to the $j-th$ task, otherwise $a_{ij} = 0$.

B. Kuhn-Munkres Algorithm

There are many implantation versions of KM algorithm with time complexity $O(n^3)$: graph, matrix. Considering a complete bipartite graph $G = (V, E)$ where $V = X \cup Y, E \in X \times Y, and\ X \cap Y = \emptyset$. The weight of edge $(x, y)\ as\ w(x, y)$. A vertex $v's$ neighborhood is the set $J_G(v)$ with the entire vertices which share an edge with v and a set $S's$ neighborhood is the set $J_G(S)$ with the entire vertices that share an edge with a vertex in S. Describe the labels $l(x)$ for every vertex in graph that is seen as dual variables of the problem, with every label of a vertices corresponding to its only matching constraint. A feasible labeling is a function $l: V \to R$ satisfies the following condition $l(x) + l(y) \geq w(x, y), \forall x \in X, \forall y \in Y$. Assuming a matching $M (M \subseteq E)$, vertex v is named matched if it is a vertex in M, otherwise it is named as exposed (unmatched, free). We represent by G_l the sub-graph of G that contains those edges where $l(x) + l(y) = w(x, y)$. G_l is a spanning sub-graph of G and integrates the entire vertices from G. G_l only incorporates those edges from the bipartite matching that allow the vertices to be perfectly feasible.

If M^* is a perfect matching in the equality sub-graph G_l, after that M^* is a maximum –weighted matching in G. We show that there are no perfect matching with greater weight than M^*. M is any given perfect matching, after that

$$\begin{aligned} w(M) &= \sum_{x,y \in M} w(x, y) \leq \sum_{x,y \in M} \{l(x) + l(y)\} \\ &= \sum_x l(x) + \sum_y l(y) = \sum_{x,y \in M^*} \{l(x) + l(y)\} \\ &= \sum_{x,y \in M^*} w(x, y) \\ &= w(M^*) \end{aligned} \tag{2}$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue X, Oct 2018- Available at www.ijraset.com

So, M^* denotes a maximal perfect matching and the KM algorithm is assured for reaching the global optimal. The time complexity of this algorithm is $O(mn^2)$, where m denotes the number of elements in X and n represents the number of elements in Y.

C. Sparse and Parallel KM Algorithms

Initially, an $O(mn^2)$ version of KM algorithm is implemented for dense matrix and analyse the memory access operations in the algorithm. We discover which these operations can be equally transformed to a row-wise and column-wise manner that would take benefit of efficient by a cross-linker data structure. This linker is utilized for saving the data and the data is manipulated along the links between the nodes. The cross-linker is relatively efficient for manipulating data with sparsity property, particularly the operations are in column-wise or row-wise way.

The complexity analysis of sKM is analyzed as given below. Describe the number of non-zero elements,

$$\tau = \#\, of\ non - zero\ elements$$

By the sparse assumption, the time complexity of algorithm sKM is $O(m\tau)$. Represent a iteration as discovering an augmenting path. Since at least one more match can be found every iteration, the total iteration number is $O(m)$.

In pKM, there are m tasks in X and n mobile devices in Y. Then, without loss of generality, we consider that $m \leq n$.

In the following algorithm, for every sub-problem in step 3, it remains that tasks are less than mobile devices. Then, if the reformulated data has very strong band structure, or even block diagonal, then we can divide the original problem in k sub-problems and solve every sub-problem through the following algorithm with t computation processing unit. The time complexity of this algorithm is $O(m(n/t)^2)$. There are in total t sub-problems computed in parallel, for every sub-problem, there are m tasks, n/t mobile devices, so every sub-problem costs $O(m(n/t)^2)$.

1) *Algorithm:* The parallel KM algorithm pKM with t threads
2) *Input:* The weight matrix $w \in R^{m\times n}\ with\ m \leq n$
3) *Output:* The maximum perfect assignment for m mobile devices

a) Reformulate the distance W into its band structure style matrix A
b) Divide the entire columns of A into t blocks uniformly
c) Distribute the entire rows into t blocks corresponding to the rows
d) Solve the sub-problems in parallel based on the KM algorithm
e) Check for uniqueness of every task, if more than two mobile devices share the same task, select a mobile device that has the maximal profit; output the results

4) *Algorithm spKM:* Assuming a sort of cases with great many tasks and relatively few mobile devices, $m << n$. There is high parallel potential for solving this type of assignment problems.

Sparsity based parallel KM (spKM) is described that is achieved through replacing KM algorithm by sKM in step 3 of the above algorithm. It is used the sparse and band structure of the assignment problem at most. The paralleled sparse KM consists of three phases: split, sub-problem solving by sparse KM and results merge. Thus, the proposed algorithm is used to enhance the computational performance.

IV. EXPERIMENTAL RESULTS

In this section, the performance comparison of the proposed approach is analysed with the existing approach. The comparison is made among proposed and existing techniques in terms of total profit and guarantee ratio.

A. Total Profit

The scenario assumed is that the client is willing to pay for its offloaded tasks. The total profits increase as the increasing number of tasks which successful to be scheduled. The profits are the balance among costs and incomes.

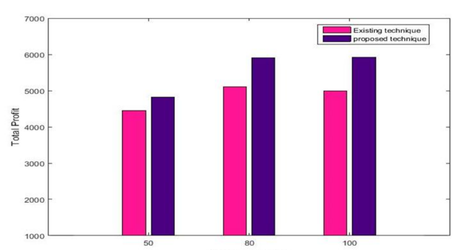


Fig. 1 Comparison of total profit

Fig 1 shows that the comparison of proposed and existing approaches in terms of total profit. From this graph, the number of tasks is represented in X-axis and total profit value is denoted in Y-axis. In this analysis, total profit is increased for proposed approach compared to existing approach.

B. Guarantee Ratio

Higher guarantee ratio brings higher profits and higher quality of service of the system. It is described as the ratio among the number of tasks that successful to be scheduled and total count of tasks.

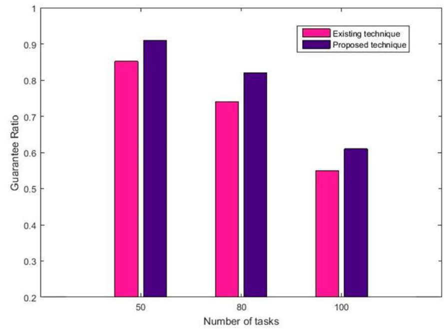


Fig. 2 Comparison of guarantee ratio

Fig 2 shows that the comparison of proposed and existing approaches in terms of guarantee ratio. From this graph, the number of tasks is represented in X-axis and guarantee ratio value is denoted in Y-axis. In this analysis, guarantee ratio is increased for proposed approach compared to existing approach.

V. CONCLUSIONS

In this work, by using sparsity based KM and parallel based KM is proposed to improve the computational performance. This approach is utilized to solve assignment problem with considerable accuracy loss. The experimental results show that the proposed approach provides better results in terms of guarantee ratio and total profit.

REFERENCES

[1] March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, & B. S. Lee, "mcloud: towards a new paradigm of rich mobile applications". Procedia Computer Science, 5, 618-624, 2011.

[2] S. Abolfazli, Z. Sanaei, A. Gani, F. Xia, & L. T. Yang, "Rich mobile applications: genesis, taxonomy, and open issues". Journal of Network and Computer Applications, 40, 345-362, 2014.

[3] K. Zkik, G. Orhanou, & S. El Hajji, "Secure mobile multi cloud architecture for authentication and data storage". International Journal of Cloud Applications and Computing (IJCAC), 7(2), 62-76, 2017.

[4] T. Wang, X. Wei, T. Liang, & J. Fan, "Dynamic Tasks Scheduling Based on Weighted Bi-graph in Mobile Cloud Computing". Sustainable Computing: Informatics and Systems, 2018.

[5] L. F. Bittencourt, A. Goldman, E. R. Madeira, N. L. da Fonseca, & R. Sakellariou, "Scheduling in distributed systems: A cloud computing perspective". Computer Science Review, 30, 31-54, 2018.

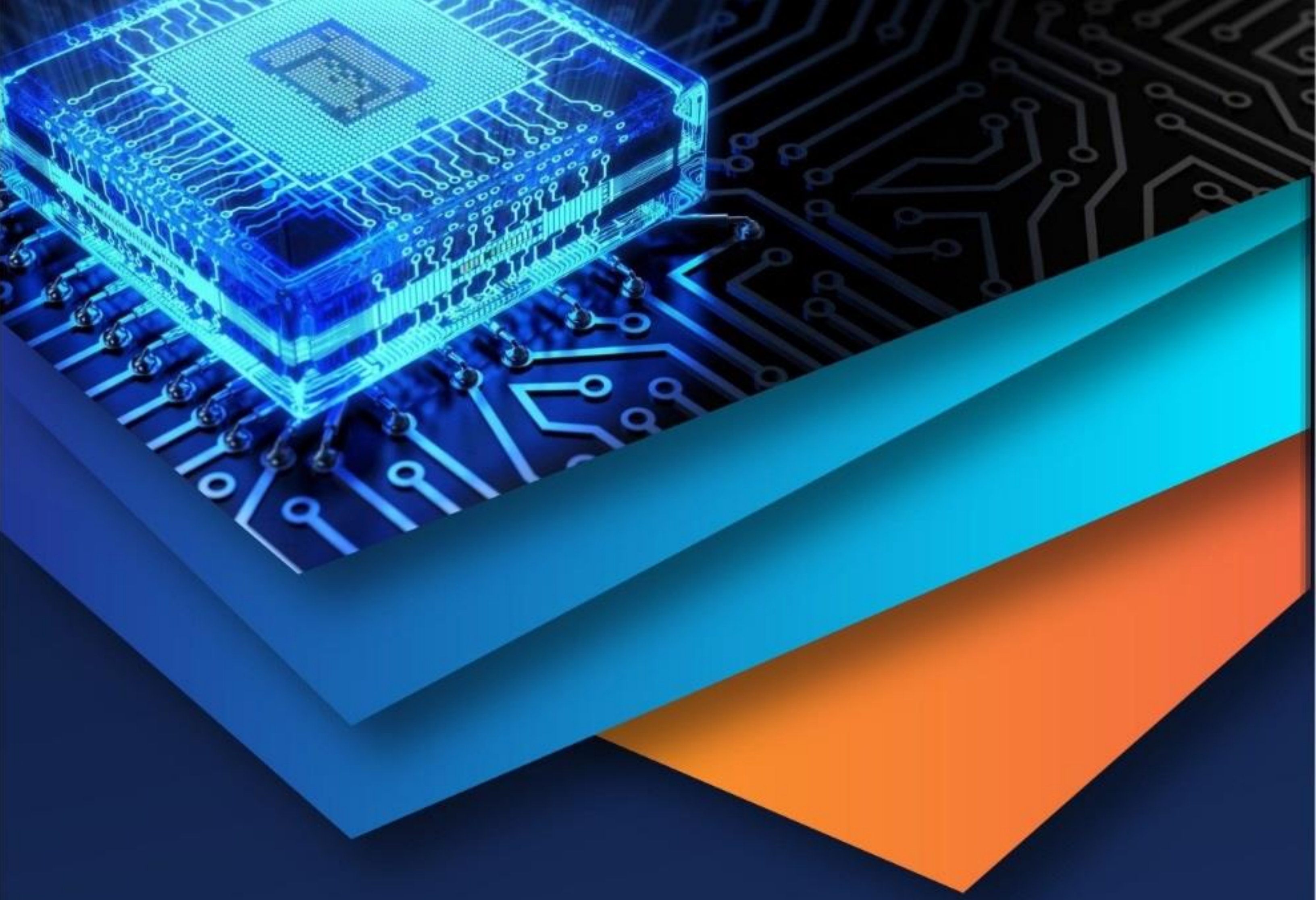
[6] H. G. E. D. H. Ali, I. A. Saroit, & A. M. Kotb, "Grouped tasks scheduling algorithm based on QoS in cloud computing network". Egyptian Informatics Journal, 18(1), 11-19, 2017.

[7] M. Abdullahi, & M. A. Ngadi, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment". Future Generation Computer Systems, 56, 640-650, 2016.

[8] F. Juarez, J. Ejarque, & R. M. Badia, "Dynamic energy-aware scheduling for parallel task-based application in cloud computing". Future Generation Computer Systems, 78, 257-271, 2018.

[9] S. Sobhanayak, A. K. Turuk, & B. Sahoo, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm". Future Computing and Informatics Journal, 2018.

[10] A. I. Awad, N. A. El-Hefnawy, & H. M. Abdel_kader, "Enhanced particle swarm optimization for task scheduling in cloud computing environments". Procedia Computer Science,65, 920-929, 2015.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY