ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 14, March 2017

DEADLINE CONSTRAINED WORKFLOW SCHEDULING ALGORITHM

Amirthavalli.J, Final year, Department of Computer Science, Kongu Engineering College, Erode.

Abinaya.T, Final year, Department of Computer Science, Kongu Engineering College, Erode.

Dhinesh.E, Final year, Department of Computer Science, Kongu Engineering College, Erode.

Dr.K.Kousalya, Professor, Department of Computer Science, Kongu Engineering College, Erode

ABSTRACT

Cloud computational platforms are very promising for execution of scientific applications since they provide ready to go infrastructure for any task. However, complex tasks, which contain a large number of interconnected applications, which are usually called workflows, require efficient tasks scheduling in order to satisfy user defined QoS, like cost or execution time (makespan). When QoS has some restrictions – execution time and deadline scheduling becomes even more complicated. In this paper we propose heuristic algorithm for scheduling workflows in hard-deadline constrained clouds – Levelwise Deadline Distributed Linewise Scheduling (LDD-LS) – which, in combination with implementation of MIN-MIN algorithm and dependency matrix. Experiment show less execution time for the task completion.

KEYWORDS: Cloud Computing, Workflow Scheduling, LDD-LS Algorithm, MIN-MIN Scheduling.

1. INTRODUCTION

Cloud computing[3] is a technology enabling high performance services for all categories of IT facilities presented to the clients as a service. The term cloud refers to the service provider that organizes all categories of resources like storage, computing, etc. Cloud has the ability to adjust the resources capacity. Scheduling maps and manages the execution of tasks on distributed resources, includes dependent where the tasks in the dependent on each other and independent where the tasks are independent, will execute independently. Workflow scheduling[4] is the problem of mapping each task to a suitable resource and of ordering the tasks on each resource to satisfy some performance criterion. Since task scheduling is a well-known NP-complete problem, many heuristic methods have been proposed for homogeneous and heterogeneous distributed systems. [5] discussed about a Secure system to Anonymous Blacklisting. The secure system adds a layer of accountability to any publicly known anonymizing network is proposed. Servers can blacklist misbehaving users while maintaining their privacy and this system shows that how these properties can be attained in a way that is practical, efficient, and sensitive to the needs of both users and services. The system can also enhanced by supporting for varying time periods. Workflows constitute a common model for describing a wide range of scientific applications in distributed systems [6]. Usually, a workflow is described by a Directed Acyclic Graph (DAG) in which each computational task is represented by a node, and each data or control dependency between tasks is represented by a directed edge between the corresponding nodes. These scheduling methods try to minimize the execution time (makespan) of the workflows. In this work we improved earlier developed Levelwise deadline distributed Linewise scheduling(LDD-LS) using MIN-MIN Scheduling.

2. LITERATURE REVIEW

Due to intensive development of cloud-based computational environments, big number of articles is related to applications execution scheduling in clouds. Unlike deadline-constrained scheduling in Grids[7], for clouds we have potentially unlimited number of resources, but we have to pay for their usage. Thus we have more complex problem of multiobjective optimization, where not only execution time, but also resources cost must be considered. Coevolution genetic algorithm was proposed in [8] for multiple workflows with hard-deadlines, however, this work does not consider the cost optimization. Authors of [9] present metaheuristic algorithm for cost efficient scheduling of long-running applications in cloud environments. Besides cost optimization, Frincu et al. took into account workload balancing between resources, which is also very challenging problem. Authors of [10] make use of multiobjective optimization theory

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 14, March 2017

[11] in order to obtain the best solutions within user-defined constraints. Application of general purpose methods for scheduling problem allows Fard et al. efficiently work with very complex cases of four objective optimization and outperform algorithms under comparison in bi-criteria cases. However, authors show that in some cases time complexity of proposed approach can rise up to $O(m.n^3)$, where m is number of resources and n is number of tasks, which makes it quite difficult to apply this algorithm for very big workflows. Zhang et al. [12] perform extensive experimental evaluation of proposed algorithm and show significant performance and quality increase compared to Monte Carlo and Blind Pick methods. One significant drawback of this article regarding our case is that it does not take into account workflow deadlines, which is crucial for us.

3. LDD-LS ALGORITHM

LDD-LS algorithm is deadline-constrained adaptation of LEFT (Linewise Earliest Finish Time) heuristic algorithm. The execution process of LEFT consists of following steps:1. Identify levels, 2. Schedule according to their relative computing times, and 3. Level wise map the task which provides the earliest finish time. In this algorithm, the deadlines are not identified. there may be a chance of assigning level 2 and 3 before the execution of level 1.So this may cause more waste of resources.

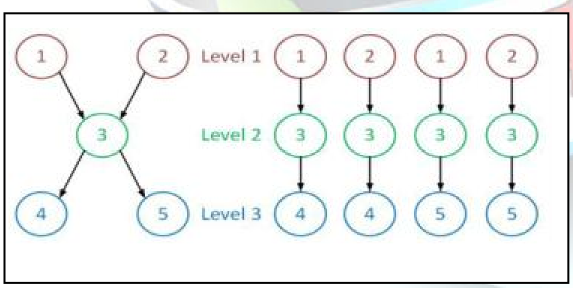

Fig.1 Workflow Linewise Representation

In our improved LDD-LS Algorithm, Directed Acyclic Graph(DAG) has been taken as an input matrix. The levels for each node are identified by the columns and rows. The deadlines are randomly generated for each and every node. The maximum completion time is assigned as deadline for each level. There is chance for wastage of resources (i.e.) unallocated resources. Fig.2 represents the input matrix. The resources will be allocated in the virtual machine corresponding to their node level. The deadline for each level is calculated by execution time and multiples of number of node.

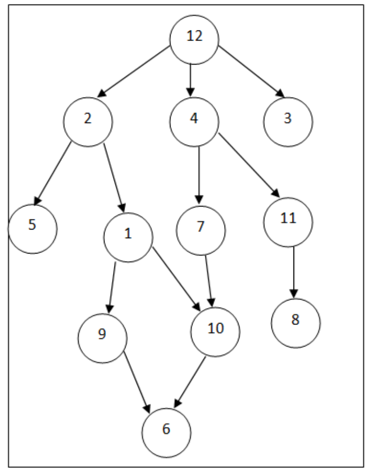

Fig.2 Directed Acyclic Graph.

4. MIN-MIN ALGORITHM

MIN-MIN Algorithm is one of the scheduling algorithms implemented. Min-Min begins with the set of all unassigned tasks(t). Firstly it computes minimum completion time(CT) for all tasks in meta-task(MT) on all resources. Then two main phases of this algorithm begins. In the first phase, the set of minimum expected completion time(ET) for each task is found. In the second phase, the task with the overall minimum expected completion time is chosen and assigned to the corresponding resource(r). Then the process is repeated until all tasks are mapped.

step 1: **for** all tasks t_i in MT
step 2: **for** all machines m_j
step 3: $CT_{ij} = ET_{ij} + r_j$
step 4: **do** until all tasks are mapped
step 5: **for** each task t_i in MT
step 6: Find minimum CT_{ij} and resource that obtains it.
step 7: find the task t_k with the minimum CT_{ij} .
step 8: Assign t_k to resource m_l
step 9: Delete t_k from MT
step 10: Update r_l
step 11: Update CT_{il} for all i.
step 12: **End do**

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 14, March 2017

5. EXPERIMENTAL RESULTS

In order to obtain results of the proposed algorithm, java was implemented. In Our scenario, the proposed algorithm is compared to the existing Levelwise deadline distributed Linewise Scheduling algorithm, for this purpose following illustrative example is taken. We have taken the DAG as input matrix and machines. Machine size ranges from 10000 to 100000.

5.1 MIN-MIN Scheduling With Deadline

The task with lowest completion time will be pushed into the corresponding machine as node as in the respective row. If machine has more than one value, the value will be added with the previous value in the machine. The maximum completion time of the entire task is assigned as makespan. Fig.3 shows the deviations of makespan for different number of tasks using MIN-MIN Algorithm using deadline.

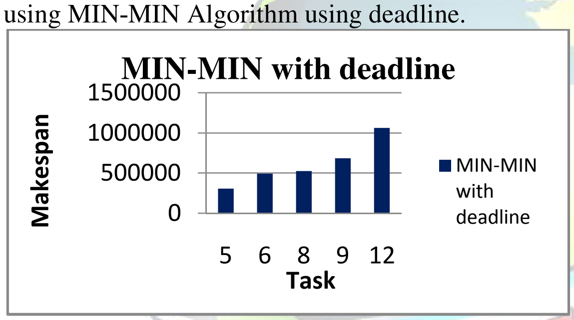

Fig.3 Makespan of MIN-MIN Algorithm using deadline.

5.2 MIN-MIN Scheduling With Dependency

In this work, the dependency matrix is used to execute the nodes. The task with no dependencies will be executed first. The task with dependency will be executed after the task completed.

```
Dependency matrix:
1:      2
2:      12
3:      12
4:      12
5:      2
6:      9       10
7:      4
8:      11
9:      1
10:     1       7
11:     4
12:
```

The dependency matrix for Fig.2

Fig.4 shows the deviation of makespan for different number of tasks using MIN-MIN Algorithm with dependency.

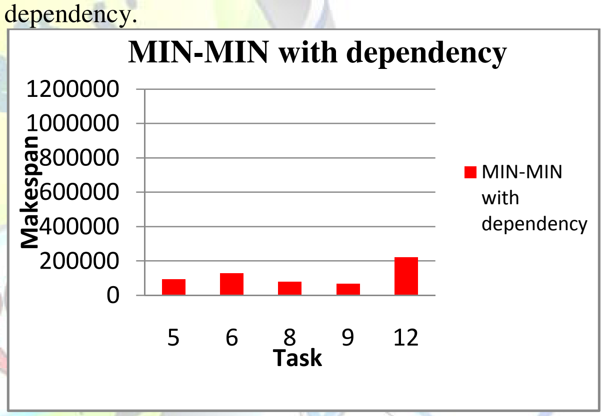

Fig.4 Makespan of MIN-MIN Algorithm using dependency

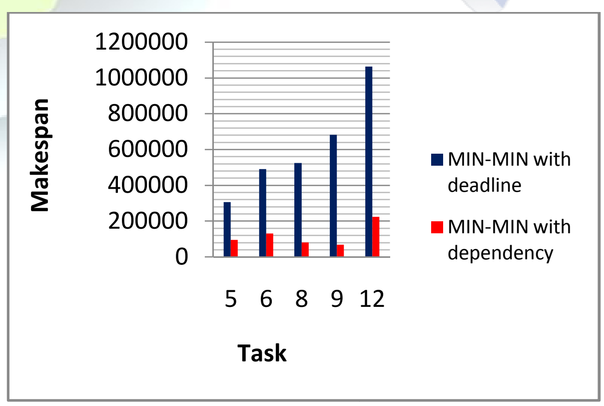

Fig.5 Comparison chart of makespan

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 14, March 2017

6. CONCLUSION

In this article we investigated the mi-min scheduling using deadline and dependency for scheduling scientific workflows. In our work we considered the deadline and makespan. Experimental results show MIN-MIN Scheduling using deadline completes the task in 1023055sec whereas MIN-MIN Scheduling using dependency matrix completes the task in 208479sec. As a result, the makespan is reduced by using MIN-MIN Algorithm.

7. REFERENCE

[1] Alexander A. Visheratin, Mikhail Melnik, Denis Nasonov "Workflow scheduling algorithms for hard-deadline constrained cloud environments"

[2] Gang Liu, Jing Li and Jianchao Xu "An Improved Min-Min Algorithm in cloud computing"

[3] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic "Cloud computing and emerging it platforms: vision, hype and reality for delivering computing as the 5th utility" Future Gener. Comput. Syst., 25 (6) (2009), pp. 599–616

[4] Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds." Future Generation Computer Systems 29.1 (2013): 158-169.

[5] Christo Ananth, A.Regina Mary, V.Poornima, M.Mariammal, N.Persis Saro Bell, "Secure system to Anonymous Blacklisting", International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Issue 4,July 2015,pp:6-9

[6] E. Deelman, D. Gannon, M. Shields, I. Taylor "Workflows and e-science: an overview of workflow system features and capabilities" Future Gener. Comput. Syst., 25 (2009), pp. 528–540

[7] Sakellariou, Rizos, et al. "Scheduling workflows with budget constraints." Integrated research in GRID computing. Springer US, 2007. 189-202.

[8] Visheratin, A., Melnik, M., Butakov, N., & Nasonov, D. (2015). Hard-deadline Constrained Workflows Scheduling Using Metaheuristic Algorithms. Procedia Computer Science, 66, 506-514.

[9] Frincu, Marc E., and Ciprian Craciun. "Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments." Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on. IEEE, 2011.

[10] Fard, Hamid Mohammadi, et al. "A multi-objective approach for workflow scheduling in heterogeneous environments." Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012). IEEE Computer Society, 2012.

[11] Marler, R. Timothy, and Jasbir S. Arora. "Survey of multi-objective optimization methods for engineering." Structural and multidisciplinary optimization 26.6 (2004): 369-395.

[12] Zhang, Fan, et al. "Multi-objective scheduling of many tasks in cloud platforms." Future Generation Computer Systems 37 (2014): 309-320.