International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 5, September- October 2017 ISSN 2278-6856

Efficient Directed Acyclic Graph Scheduling In Order To Balance Load At Cloud

Sandeep Kaur[1], Pooja Nagpal[2]

[1]Research scholar (CSE) Rayat Group Of Institutions, Railmajra, Punjab, India

[2]Associate Professor (CSE) Rayat Group Of Institutions, Railmajra, Punjab, India

Abstract

Cloud computing has offered services related to utility aligned IT services. Reducing the schedule length is considered as one of the significant QoS need of the cloud provider for the satisfaction of budget constraints of an application. Scheduling computing tasks in the processor is the key issue of Advanced Computing. Scheduling list become a topic of conversation for the developers because it helps to solve high complexity with minimum complexity and thus estimates additional scheduling of application matrix problem. The main aim is to decrease the overall time to complete a task. For reducing the computation cost and earlier finish time of the system prioritization of the jobs will be done. Task scheduling in a parallel environment is one of the NP (Non deterministic polynomial) problems, which deals with the optimal assignment of a task. To deal with the favorable assignment of some task, task scheduling is considered as one of the NP problem. In this research, for balancing the load algorithm DAG and HEFT will be used. To reduce the load and total execution time jobs will be executed in an order. For the optimization of the traditional scheduling and balancing algorithm, an algorithm has been designed for reducing the delay. DVFS is used to perform the tasks in less time. The job placement also has a great impact on the cost computation. Here, the placement is done by using Optimization algorithm that is Bee colony optimization (BCO) algorithm for generating high-quality solutions for optimizing and searching the problems by depending on bio-inspired operator, namely mutation, crossover and selection. Metrics namely, make span, CCR (Computation Cost Ratio) and Energy consumption are used for the evaluation of the proposed work. All the simulations will be carried out in CLOUDSIM environment.

Keywords: CLOUDSIM, Computation cost ratio (CCR), Bee colony optimization (BCO),Dynamic voltage and frequency scaling (DVFS),Make span, energy consumption, DAG, HEFT

1. INTRODUCTION

Cloud computing is a service for the network resources and these days, cloud computing is taken as well developed. The system of cloud computing develops the resources shared network with the user services. Because of the dynamic, diversified and flexible nature, varied services with the resources are provided to variety of users considered as a benefit to the cloud computing. For solving the problem of task scheduling, few researchers has given many feasible and effective algorithms being integrated into algorithms of scheduling algorithms and list scheduling on the basis of clustering, duplication and random search.

In an effective form of static task scheduling, an application consists of number of tasks with the data dependencies among them. It is represented by DAG (Directed acyclic Graph). DAGs can model many various kinds of information. A spreadsheet can be modelled as a DAG, with a vertex for each cell and an edge whenever the principle in one cell uses the value from another; a topological ordering of this DAG can be utilized to modify all cell values when the spreadsheet is varied. Computational trouble on DAGs consists of topological sorting (finding a topological ordering), construction of the transitive closure and transitive reduction, and the closure problem, in which the aim is to find a minimum-weight subset of vertices with no edges link between them to the rest of the graph.

2. MATERIALS AND METHODS

In this research, for balancing the load algorithm DAG and HEFT can be used. To reduce the load and total execution time jobs will be executed in an order. For the optimization of the traditional scheduling and balancing algorithm, an algorithm has been designed for reducing the delay. For this, DVFS (dynamic voltage and frequency scaling) mode is applied that permits the devices for performing the required tasks with the less amount of required power. It also scale upwards for increasing the performance. The job placement also has a great impact on the cost computation. A job selection algorithm has been designed and implemented for the minimization of the cost and energy consumption. BCO (Bee Colony Optimization) algorithm has been used for the Optimization of Processor selection in order to balance the load.

A. Task model

Computational problem on DAGs consists of topological sorting (finding a topological ordering), construction of the transitive closure and transitive reduction, and the closure problem, in which the aim is to find a minimum-weight subset of vertices with no edges link between them to the rest of the graph.

In this model, a set of node $(n_1, n_2, n_3 \ldots \ldots . n_n)$ are connected by a set of a directed edges, which are represented by (n_i, n_j) where $n_i$is called the Parent node and n_j is called the child node. A node without parent is called an Entry node and a node with no child is an Exit node. The weight of a node is denoted by w (n_i) which represents the process execution time of a process. Since each edge corresponds to a message transfer from one process to another, therefore, the weight of an edge which is equal to the message transmission time from node n_i to n_j. Thus, w (n_i, n_j) becomes zero when n_i and n_j are scheduled to the same processor because intra processor communication time is negligible compared with the inter processor communication time.

The granularity of DAG is

$$f(F) = \min_{x=1,..w}\{ f(x)\} \qquad (1)$$

The DAG coarse grain is

$$f(L) \geq 1 \qquad (2)$$

The grain of the task can be written as

$$F_x = \max \{F(k_x)F(L_x)\} \qquad (3)$$

And the DAG's Granularity is

$$F(F) = \max_{x=1,...v}\{F_x\} \qquad (4)$$

The DAG's fine grain is introduced as

$$F(F) \leq 1$$

B. HEFT (Heterogeneous Earliest Finish Time)

HEFT calculates the rank value for each task nodes on the basis of task nodes mean computational cost and the communication cost among the task nodes. The task node order series is produced by sorting all task nodes as per the decreasing order of rank value. The selection of task nodes can be executed one by one from the order list and transfers to the equivalent server that reduces the finish time of task node. The algorithm is viable and simple, and its time complexity is "O ($n^2 * k$)", in which the number of task nodes is n; and k is the definite number of servers.

C. BCO(Bee Colony Optimization) Algorithm

The Bee Colony Optimization is taken from the bee's behaviour in nature. The actual objective for BCO is the creation of multi agent system for solving the different combinatorial optimization issues. The system of bee is known as an example for the organized team work performing well with the coordinated interaction, labour division, coordination, specialized individuals, simultaneous task performance, and well-knit communication [43]. Bee colony consists of different types of bees as defined below:

- The Queen's responsibility is with the laying eggs for the novel colonies that can be produced.
- The Drones are the males of the hive with the responsibility for mating with Queen. It is the alone role presented in hives and they are later discarded from the colony with their down fall.
- The worker bees are the females in the hive. They are considered as the main building blocks for the hives. The workers built the honey bee comb with cleaning, maintaining, guarding, feeding the drones and queen.

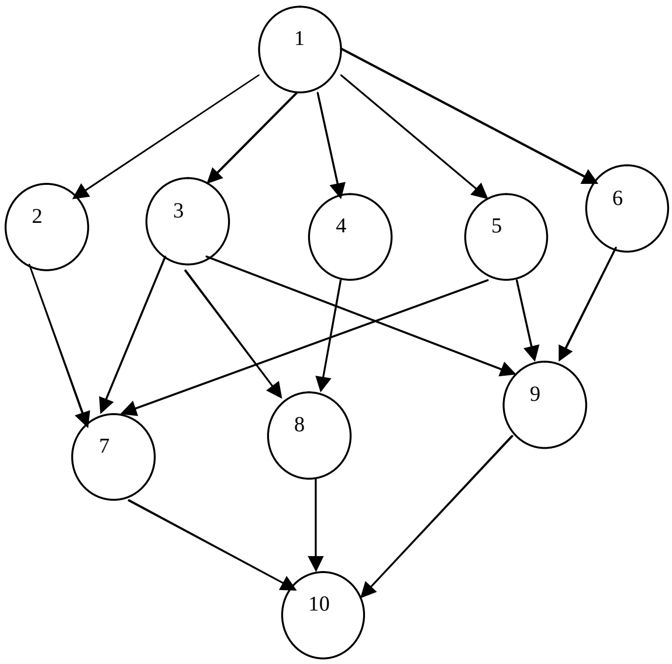

Figure 1. Traditional load balancing problem

3. SIMULATION MODEL

This section explains the work flow of the proposed algorithm in the form of flowchart.

The diagram 1 shows a tradition problem of load balancing and the cost computation. ------>arrow represents a child node.

Multiple ----> arrows represents multiple parents. As for example, Node 7 has three parents namely 2, 5, 3. Placing node 7 above of node 8 who has only 2 parents 3 and 4 will produce more delay into the architecture. The steps required to complete the proposed work are listed below:

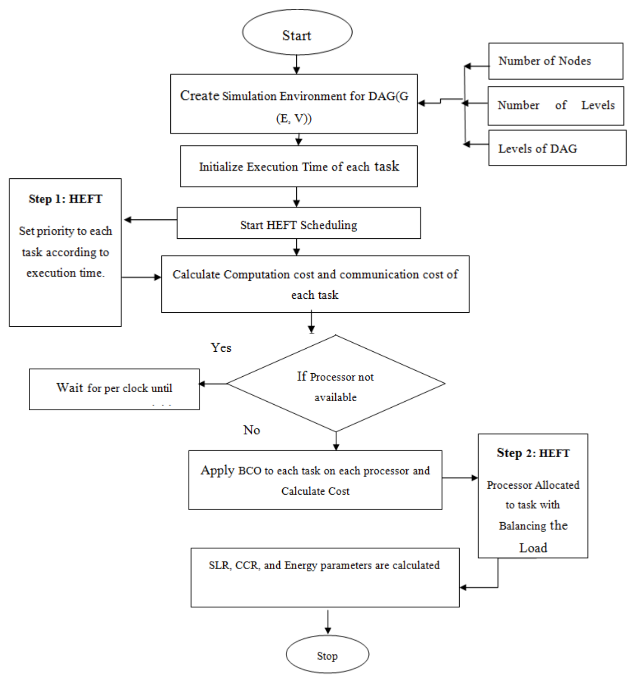

Figure 2. Proposed Methodology Flowchart

Step 1: A simulation environment for directed acyclic graph (DAG) has been created by using number of node and number of levels.
Step 2: Execution time for each task has been assign. It is time required by the system to execute its task.
Step3: Use HEFT scheduling to set the priority to each task according to their execution time.
Step4: Calculate the performance parameters like computation cost and communication cost of each task.
Step5: If process is available then apply Bee colony optimization (BCO) algorithm to each task on each processor and thus calculate performance parameters like Cost. Otherwise wait for per clock until processor is not available.
Step 6: After applying BCO algorithm processor allocate the task with load balancing and thus QoS parameters like SLA, CCR and Energy have been calculated.

4.SIMULATION RESULTS

This section explains the results obtained after the simulation of the whole work.

TABLE 1 JOB/NODE AND PROCESSORS

P1	P2	P3
13	10	11
11	6	8
9	23	14
11	12	16
15	13	18
19	16	15
8	7	9
11	13	15
15	12	9
1	4	6

Costing in terms of milli joules have been set to signify the energy consumption concept. The table says that if NODE 1 is executed on Processor P1 at 13 mille Joules if executed on Processor 2 than it takes 10 joules. If represented graphically, the following graph would be attained. First of all the order of the jobs has to be decided. The decision making algorithm has been discussed in the methodology section. The following order is gained for one set of values.

TABLE 2 JOB ORDER

Job Order					
Level 1	1				
Level 2	2	3	4	5	6
Level 3	7	8		9	
Level 4	10				

There is only one Job at level 1, 5 Jobs at level 2, 3 Jobs at level 3 and one exit Job at level 4.
The data can be represented diagrammatically as follows:

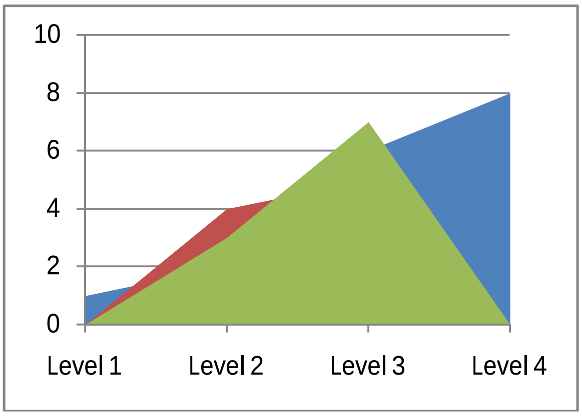

Figure 3 Precedence order

Now for scheduling the tasks/nodes we have two parameters Earliest Start Time(EST) and Earliest Finish time (EFT) and for calculating Earliest Finish Time we have formula.
EFT= EST + Communication Cost + Computation Cost
When we applied BCO on tasks more tasks were scheduled on the Processors than the normal task scheduler within the time frame. When we used BCO algorithm the Energy consumption was reduced and the tasks were scheduled within the Time frame. More time taken means more energy consumption. So figures represent EFE (Earliest finish Energy Consumption) for the Energy consumed for the proposed that is optimized by BCO and the task scheduled without optimization.

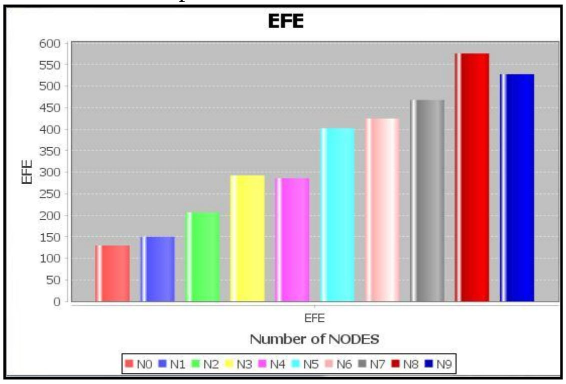

Figure 4 Earliest Finishing Energy Consumption DVFS inspired

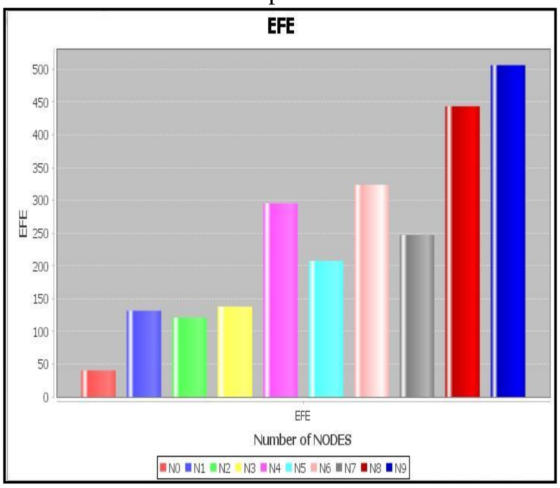

Figure 5 Earliest Finishing Energy Consumption BCO inspired Now, the processor Average Energy Cost is

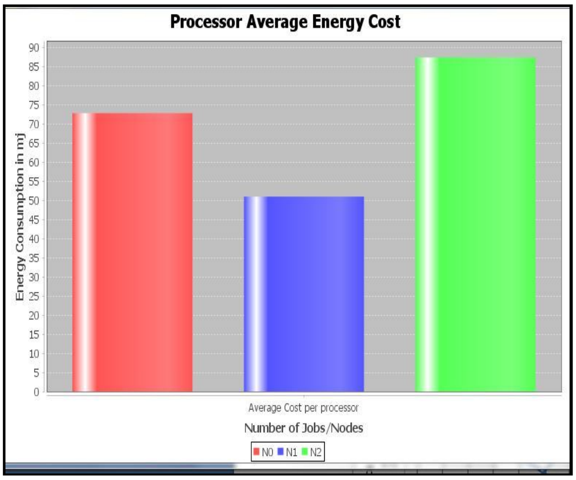

Figure 6 Processor Average Energy Cost in milli joules

a) CCR (Computational Cost Ratio) = Total computed energy consumption by proposed algorithm/ Total energy consumed if all the jobs are scheduled at one processor only.

b) Make span= Total time consumed for the evaluation of all jobs.

c) Total consumed energy for the execution of all jobs.

5. CONCLUSION

DAG was designed to solve multiprocessor complex problems and is quite an efficient algorithm. Even though, it is very significant, it requires some modifications into it and that what has been done in the proposed solution. The DAG overlooked the concept of dependency and in the proposed solution; the dependency diagram has been drawn. The entire framework has been designed to reduce the consumption of energy in the entire set. 10 different set of Job series has been taken to perform the proposed algorithm and it has been observed that the results are significant and almost identical. This proves that even if the situation changes, the proposed algorithm works well. Usage of Bee Colony Optimization (BCO) has improved the processor selection procedure and as a reflection of the improvement the energy consumption has looked consistent in all set of jobs. The entire framework has been evaluated over 3 set of parameters namely CCR, MAKESPAN and Total Energy Consumed.

The current framework provides great opportunity for the future research workers. The researchers may opt. Genetic Algorithm instead of BCO and see the effect. Usage of other swarm intelligence techniques like Cuckoo Search, Particle Swarm would be also interesting to observe.

REFERENCES

[1] Chauhan N. and Saxena A., “A Green Software Development Lifecycle for Cloud Computing”, IEEE’s IT Pro, pp. 28-34,2013.
[2] Elkington J., “Cannibals with forks: The triple Bottom Line of 21st century business”, New Society Publishers, UK,1997.
[3] Ellison D. , “Addressing Adaptation in the EU Policy Framework”, Springer, E.C.H. (ed.),2010.
[4] Epstein M. and Roy M., “Sustainability in Action: Identifying and measuring the key performance drivers”, Long Range Planning Journal, Vol.34, pp. 585-604,2001.
[5] Hedman J. and Henningsson S., “Three Strategies for Green IT”, IEEE’s IT Pro, pp. 54-57,2011.
[6] Krikke J., “Recycling e-waste” The sky is the limit”, IEEE’s IT Pro, pp.50-55,2008.
[7] Z. Li; S. Tesfatsion; S. Bastani; A. A. e. Hassan; E. Elmroth; M. Kihl; R. Ranjan, “A Survey on Modeling Energy Consumption of Cloud Applications: Deconstruction, State of the Art, and Trade-off Debates,” in IEEE Transactions on Sustainable Computing , no.99, pp.1-1,2017.
[8] K. Dhal, P. K. Pattnaik and S. C. Rai, “Critique of Some Fine-Grained Access Control Models in Cloud Computing Environment,” International Conference on Information Technology (ICIT), Bhubaneswar, India, pp. 206-211,2016.
[9] Q. Lei, Y. Jiang and M. Yang, “Evaluating Open IaaS Cloud Platforms Based upon NIST Cloud Computing Reference Model,” IEEE 17th International Conference on Computational Science and Engineering, Chengdu, pp. 1909-1914,2014.
[10] R. B. Bohn, J. Messina, F. Liu, J. Tong and J. Mao, “NIST Cloud Computing Reference Architecture,” IEEE World Congress on Services, Washington, DC, 2011, pp. 594-596,2011.
[11] P. Jain, A. Datt, A. Goel and S. C. Gupta, “Cloud service orchestration based architecture of OpenStack Nova and Swift,” International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, pp. 2453-2459,2016.
[12] W. Tan, Y. Sun, L. X. Li, G. Lu and T. Wang, “A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing,” IEEE Systems Journal, Vol. 8, No. 3, pp. 868-878, Sept. 2014.
[13] S. Ahmad, S. Paul and A. P. Singh, “Tokenization based service model for cloud computing environment,” International Conference on Inventive Computation Technologies (ICICT), Coimbatore, pp. 1-7,2016.
[14] M. U. Bokhari, Q. M. Shallal and Y. K. Tamandani, “Cloud computing service models: A comparative study,” 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 890-895,2016.
[15] K. Xue and P. Hong, “A Dynamic Secure Group Sharing Framework in Public Cloud Computing,” IEEE Transactions on Cloud Computing, vol. 2, no. 4, pp. 459-470, Oct.-Dec. 1 2014.
[16] R. SoaresBoaventura, K. Yamanaka, G. Prado Oliveira, E. Rodrigues Pinto and M. Braz de Andrade Maciel, “Methodology for statistical analysis comparing the algorithms performance: case of study in virtual environments in private cloud computing,”

IEEE Latin America Transactions, Vol. 15, No. 2, pp. 333-340, Feb. 2017.

[17] Z. Bouzidi, L. Terrissa, A. Lahmadi, N. Zerhouni and R. Gouriveau, “Neuro-fuzzy model for Prognostic as a Service in private cloud computing,” 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), Marrakech, pp. 360-367,2016.

[18] X. Mao, C. Li, W. Yan and S. Du, “Optimal Scheduling Algorithm of MapReduce Tasks Based on QoS in the Hybrid Cloud,” 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Guangzhou, pp. 119-124,2016.

[19] GaoXincheng and Wang Yan, “Research of digital community service platform based on cloud computing,” 8th International Conference on Computer Science & Education, Colombo, 2013, pp. 824-828,2013.

[20] U. Idachaba and F. Wang, “A Community-Based Cloud Computing Caching Service,” IEEE International Congress on Big Data, New York, NY, 2015, pp. 559-566,2015.