Volume 8, Issue 3
June 2019

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

# Efficacious Hybrid Algorithm for Scheduling Task in Cloud Computing

**Salil Bharany**

Department of Computer Science and Engineering, GNDU, Amritsar

E-mail:salil.bharany@gmail.com,

## ABSTRACT

Cloud Computing is an expanding technology that provides a wide range of online resources. Sharing resources among cloud users makes task scheduling challenging. The problem of scheduling of tasks in many cases is addressed through a method or approach by meta-heuristic. This paper provides a solution for planning tasks in cloud computing environments based on meta-heuristic, Genetic Algorithm .Suggested solutions, i.e The Efficacious Hybrid Algorithm (EHA) use a hybrid model which is fusion of genetic algorithm and with the predict earliest finish time scheduling on a hybrid-based variant with Directed Acyclic Graph (DAG) A result of the modified genetic algorithm is compared with Genetic Algorithm and with hybrid GA with HEFT (Heterogeneous Finish Time First) scheduling algorithms.In addition, comparative analysis is based on the use on average processor utilization, processing cost metrics. It has been observed that EHA provides better results with respect to the cost of processing and the use of the processor for unlimited number of processors.
.

**Keywords:** Cloud computing, Task scheduling, DAG, Genetic algorithm, HEFT, PEFT

## 1. INTRODUCTION

Cloud is used as an Internet and computer comparisons called cloud computing [2]. The Cloud Computing environment offers three basic services, a Infrastructure as a Service (IaaS), providing a necessary operational environment, Platform as a Service (PaaS) that provides essential tools and services to developers to start applications. Software as a Service (SaaS) that disseminates to the service provider [13]. Clouds have many users from different geographical areas that need the required cloud resources to perform their tasks. For this virtual technology which is known as virtualization creates the virtual source of resources, dividing it into a more computer-centric environment called virtual machines (VMs). Because the user has more than the available resources in the cloud, it needs a schedule.The Task Scheduler [22] splits tasks to Virtual Machine (VM) in a way that reduces total time to complete all tasks. Additional load balancing [2] Spare distributes the load between the available VM in a manner that hosts should not be overweight or overloaded. If so, the task is transferred from VM to VM under-loaded VM. This paper includes GA schedule algorithms with PEFT [1] to manage total time for all tasks, increase the use of the processor and reduce operating costs by using VM. Section 2 in This paper consist to shows a brief study of existing meta-heuristic GA. Section 3 describes the method of the proposed algorithm. In Section 4, the results are evaluated using Indicators of Efficiency of Implementation, and Finally, the conclusions and the scope of the future are compiled in Section 5.

This paper includes GA schedule algorithms with PEFT [1] to manage total time for all tasks, increase the use of the processor and reduce operating costs by using VM. Section 2 in This paper consist to shows a brief study of existing meta-heuristic GA. Section 3 describes the method of the proposed algorithm. In Section 4, the results are evaluated using Indicators of Efficiency of Implementation, and Finally, the conclusions and the scope of the future are compiled in Section 5.

## 2. RELATED WORK

Shaminder Kaur et al. (2012) [12] construct a cloud-based scheduling algorithm. The author uses Shortest Cloudlet to Fastest Processor (SCFP) and Longest Cloudlet to Fastest Processor (LCFA) to begin the population of the GA. Their algorithms use a variable-length processors and variable length tasks that represent real-time scenarios but are considered to work only with one user. Sung Ho Jang et al. [8] Guide an algorithm in 2012, considering the parameters of service quality recorded in the SLA. Their approach uses reboots and stop conditions to find the most appropriate solution that blocks GA from getting the best fall in Local Optimum and finding explore various search spaces. However, the results were evaluated using the normal static number of jobs.

In 2013, GE Junwei et al. [3] Requested an algorithms for the cost and deadline. He sets the time for an intensive task, dividing a great task into the sub task and controlling their reliance on them. But they did not look at a standard dataset and did not assign task priorities. Kousik Dasgupta et al. (2013) [2] has demonstrated a method of processing for the process and reducing productivity but taking on the equal priority task

In 2014 Rajveer Kaur et al. [11] Developed an action algorithm that prioritized the task by following a role-based access management approach. The role is determined by authority, authorization, and responsibility. It reduces the execution time of all tasks but takes into account the number of tasks. Tingting Wang et al. [20] Presented in Job spanning time Load balancing Genetic Algorithm (JLGA) in 2014. He uses a greedy approach to give a start to population . JLGA overcomes local issues of optimism and offers better results than conventional genetic methods. Their algorithms reduce the makespan and balance of burdens, but they do not prioritize the job.

Yuming Xu et al. (2014) [22] Developed a strategy that controlled the computer system of different characteristics. The last technology of the Heterogeneous Earliest Finish Time (HEFT) is used to determine tasks priority . This includes more search space without bringing higher calculation costs and providing higher execution speeds to the subtask. It provides better results than the standard HEFT algorithm, but considered limited number of tasks and number of processors.

In 2016, Xiaodong Sheng et al. [21] Presented a template-based method for an independent task. In their documents, the templates are done by a number of tasks, depending on the maximum size of task. It focuses on QoS requirements and user requirements, except for the cost constraints. Safwat A. Hamad et al. [5] By 2016, algorithms are advised to reduce the totaling time and cost of processing and increase productivity. A special crossover operation used to produce four of the best of two was chosen to produce a new generation. Their algorithms provide better results from Round-Robin and Basic GA, but consider a limited number of independent tasks is the only issue .

## 3. PROPOSED ALGORITHM (PA)

This article provides an algorithm named PA, which combines the Genetic Algorithm (GA) with its existing Predict prediction schedule (PEFT). The main accent of the algorithm is shown in the block diagram of the figure.

```
3. Input the DAG workflow to the cloud simulator.
4. Topological sorting  done on list of  DAG .
5. initial population of dependent tasks is generated . the
   number of chromosome sets is output for this step.
       Fitness of each chromosome set is calculated.
6. While (!termination condition) do
              5.1 Select the fittest parent.
              5.2 Crossover between the selected parent
              offspring using uniform crossover.
              5.3 Mutate the parent offspring by bit flip
              mutation.
     End While
1.   Output the scheduling list.
```

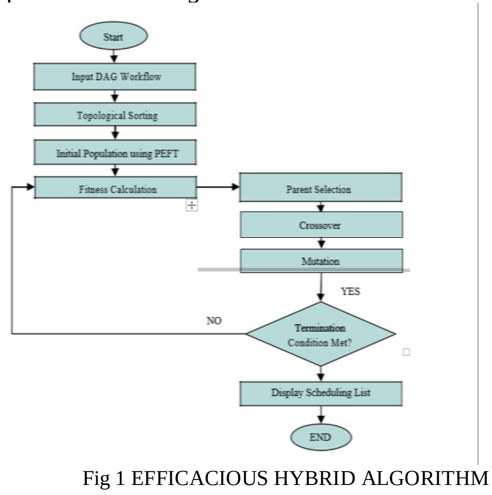


Fig 1 EFFICACIOUS HYBRID ALGORITHM

The steps of algorithm is given as under:-

**3.1** ***Input DAG Workflow*****:** DAG [10] at compile time is input in the EHA algorithm. This DAG file contains the size, instructions, names, and dependencies in XML format. DAG Parser analyzes DAG files in file size, count by the size of the task in seconds, the main contacts **-** the child and the number of tasks to be input and determines the result of the task.

**3.2** ***Topological Sorting:*** It provides the sorted list of all nodes of the DAG by taking care of the precedence dependencies among them.

**3.3** ***Initial Population*****:** It was created with the PEFT algorithm. The result of this step provides the amount of chromosome sets. The proposed chromosomal algorithm determines scheduling list based on minimum time (Ti) and its its successor task (Ti + 1) as shown below, where n is the total number of tasks.

$$\sum_{I=0}^{n}(size(Ti\& Ti+1) < size\ (Ti+1\& Ti+2)) \quad (1)$$

The task of low execution time has a high priority than the task of high execution time.

**3.4** ***Fitness Calculation:*** Each solution (chromosome) as obtained in a previous step is checked against the task execution time and processing cost constraints [15]. The solution that has a lower total execution time and execution cost have a higher fitness value to survive**.** Fitness is calculated using equation 2, where ET is an execution time of a task.

$$F = \min(\max(\sum_{I=0}^{n} \mathrm{ET(Ti)}), \max(\text{number of VMs})) \quad (2)$$

of GA is used as termination condition. While the termination condition is not met the following steps will be executed.

3.5.1 *Tournament Selection (TS):* It initially selects the number of parents randomly. Amongst them the solution which has highest fitness value is selected that acts as a parent offspring for the next generation. TS with tournament size equal to 5 is used to select parent offspring.

**3.6** ***Crossover*****:** It is a convergence operation that generates new solution by merging some of the parent bits for the creation of child offspring. Uniform Crossover with 0.5 probability value is used here to generate new offspring. It selects the two parent offspring and the bits of parent offspring that has a lower value than probability value is selected for child offspring and other one is discarded.

**3.7** ***Mutation:*** It is a divergence operation that is used to preserve the Genetic diversity from one generation to the next generation. Bit Flip Mutation will flip the bit from 0 to 1 and from 1 to 0 for the one that has a lower value than the given probability i.e. 0.015.

**3.8** ***Scheduling List:*** It will display the final scheduling list with their execution time and processing cost.

## 4. PERFORMANCE COMPARISON

The performance of proposed Modified Genetic Algorithm is compared and analyzed with well known scheduling algorithms, namely GA and GA with HEFT using performance calculation metrics.

1. ***Makespan:*** It defines the total time taken by all the incoming tasks from task submission to completion of a last task as given in equation 3.
   **Makespan=LastTask** Finish Time (3)
   where LastTask FinishTime is an exit time of last task.

- ***Processor Cost:*** Processor or VM cost represents the processing cost in term of MIPS (million instructions per second). Cost depends on the size of tasks and number of VMs as calculated in the equation 4.

$$\textbf{Processor cost} = \sum_{i=0}^{m} \left(\frac{\textbf{VmRumtime}}{\textbf{Billing time in seconds}}\right) * \textbf{BillingPriceUnits}$$
(4)

Where m is a number of processors (VMs), VMi_Runtime describes ith VM's runtime,
BillingTime_in Seconds and Billing Price Units define the predefined time on which cost will be applied, i.e. 1.0 Price Unit for 3600 Time Units.

***Number of Virtual Machines:*** The number of VMs directly affects the performance of an algorithm. A lower number of VMs will increase the makespan whereas the larger number will reduce the makespan, but it will increase the processing cost. It is calculated using equation 5.

$$\boldsymbol{Number\ of\ VMs} = \sum_{i=1}^{n} \boldsymbol{i} \qquad (5)$$

IV ***Average Processor Utilization*** Processor utilization defines how efficiently resources are utilized by the number of available tasks. It is calculated using equation 6

**Processor utilizationVMS Avaible time / #Vms*makespan *100**
(6)
where VMsAvailableTime is VM's remaining runtime, #VMs defines the number of virtual machines

V ***Speedup:*** This metric show reduced parallel scheduling time related to sequential scheduling time. Speedup is calculated using equation 7 where #VM is a number of Virtual Machines
**SPEEDUP=Total Execution Time of task on 1VM / Total execution time of task on #VMs** (7)

VI.***Efficiency:*** It is calculated from the speedup. It is a ratio of speedup and number of used resources orprocessors. The efficiency value lies between 0 to 1. Higher value of efficiency represents the algorithm gives higher throughput and productivity. It is calculated using equation 8.

***Efficiency=speedup/VM***

4.1 ***Simulation Results:*** The simulation results are implemented On in 4GB RAM on Windows 7 OS, Eclipse with JAVA with Workflow Sim tool. The performance metrics (makespan, processor cost, number of virtual machines, average processor utilization, speedup and efficiency) of proposed EHA and two other existing GA and GA+HEFT are calculated shown in Table 1 to 6.

- **Airlines:** American, Delta, United, Southwest, JetBlue, Spirit
- **Weather Events:** April 25-28, 2011 Tornado Outbreak, Joplin, Missouri Tornado (May 2011), Hurricane Irene (August 2011), Halloween Nor'easter (October 2011)
- **Weather Services:** The Weather Channel, Accu Weather, NOAA

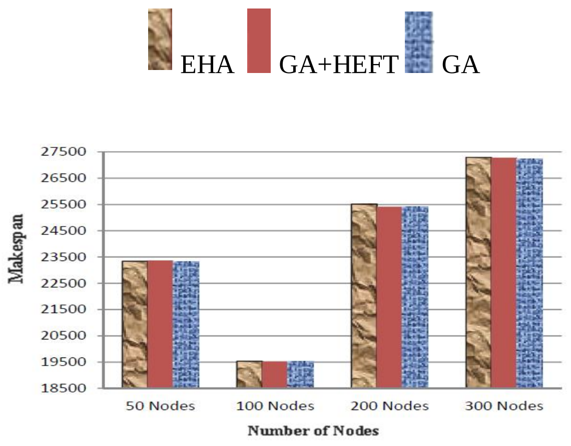
| | No. of Nodes | EHA | GA+HEFT | GA |
|---|---|---|---|---|
| Makespan | 50 Nodes | 23340.83 | 23366.29 | 23332.9 |
| | 100 Nodes | 19533.79 | 19526.28 | 19526.28 |
| | 200 Nodes | 25512.02 | 25414.6 | 25414.6 |
| | 300 Nodes | 27291.3 | 27280.45 | 27240.4 |

Table 1 calculated makespan with respect to varying number of nodes

| | No. of Nodes | EHA | GA+HEFT | GA |
|---|---|---|---|---|
| Processor Cost | 50 Nodes | 38 | 44 | 62 |
| | 100 Nodes | 80 | 126 | 130 |
| | 200 Nodes | 173 | 263 | 272 |
| | 300 Nodes | 272 | 310 | 389 |

Table 2 Calculated processor cost with respect to varying number of nodes

Above table values of makespan and processing cost depicts that the EHA reduces the processing cost than GA and GA+HEFT strategy with slight increase in makespan for unbounded number of VMs.



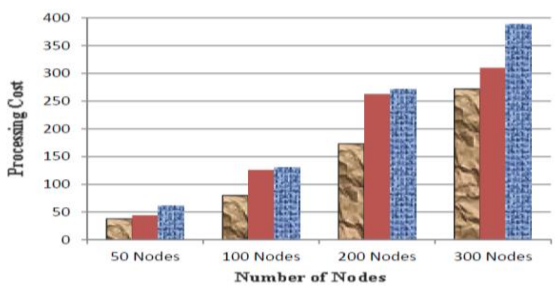


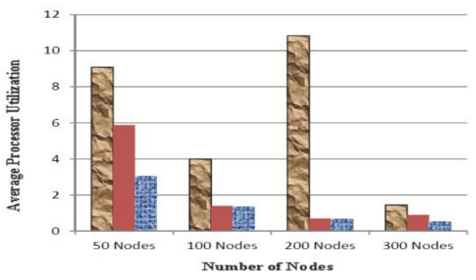
Table 3 Calculated numbers of VMs with respect to varying
number of nodes

Table 3 Calculated numbers of VMs with respect to varying number of nodes

| | No. of Nodes | EHA | GA+HEFT | GA |
|---|---|---|---|---|
| Number of VMs | 50 Nodes | 11 | 17 | 33 |
| | 100 Nodes | 25 | 71 | 74 |
| | 200 Nodes | 49 | 139 | 147 |
| | 300 Nodes | 71 | 110 | 187 |

Above table values shows that the proposed algorithm
Uses the less number of virtual machines with larger average processor utilization rate than GA and GA+HEFT solutions. With 200 nodes, EHA uses only 49 VMs whereas GA uses 147 VMs also GA and GA+HEFT utilizes 0.6% to 0.7% of processor whereas EHA has 10.83% of processor utilization value.

| | No. of Nodes | EHA | GA+HEFT | GA |
|---|---|---|---|---|
| Average Processor Utilization | 50 Nodes | 9.09 | 5.88 | 3.03 |
| | 100 Nodes | 4 | 1.4 | 1.35 |
| | 200 Nodes | 10.83 | 0.7 | 0.68 |
| | 300 Nodes | 1.45 | 0.9 | 0.53 |



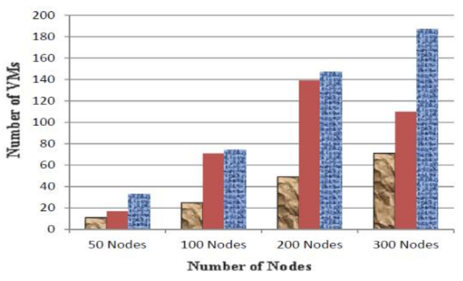


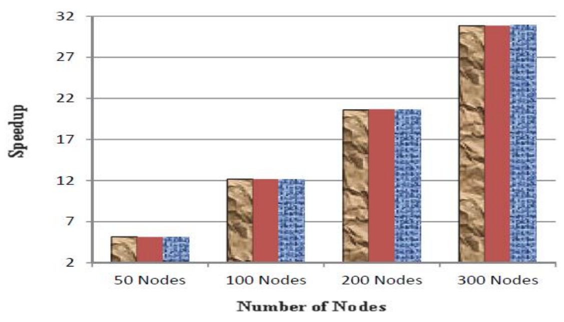
Table 5 Calculated speedup with respect to varying number of nodes

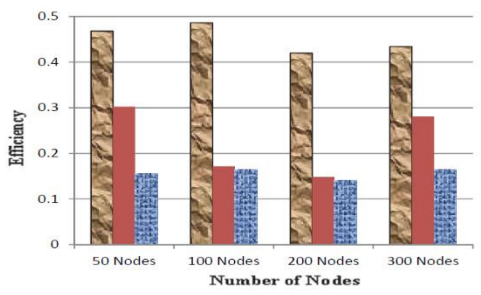
| | No. of Nodes | EHA | GA+HEFT | GA |
|---|---|---|---|---|
| Speedup | 50 Nodes | 5.152 | 5.142 | 5.154 |
| | 100 Nodes | 12.163 | 12.168 | 12.168 |
| | 200 Nodes | 20.589 | 20.668 | 20.668 |
| | 300 Nodes | 30.844 | 30.856 | 30.902 |

| | No. of Nodes | EHA | GA+HEFT | GA |
|---|---|---|---|---|
| Efficiency | 50 Nodes | 0.4684 | 0.3025 | 0.1561 |
| | 100 Nodes | 0.4865 | 0.1713 | 0.1644 |
| | 200 Nodes | 0.4202 | 0.1486 | 0.1406 |
| | 300 Nodes | 0.4344 | 0.2805 | 0.1652 |

Table 6 Calculated efficiency of the algorithms with respect to varying number of nodes

Speedup reflects the total completion time of the algorithm. The minimum total completion time will increase the speedup. As shown in the above table, in case of 50 nodes speedup of GA is 5.154 whereas speedup of EHA is 5.152, which is slightly lower but has higher efficiency.





## 5. Conclusion and Future Scope

This paper proposes a hybrid approach based solution for task scheduling. Proposed solution, Efficacious Hybrid Algorithm(EHA) assigns a priority to the tasks based on Predict Earliest Finish Time and optimizes the scheduling results using the meta – heuristic, Genetic Algorithm. Experimental results have been recorded based on metrics like makespan, processing cost, speedup, efficiency, number of used processors and processor utilization. Further, comparison with basic GA and GA with HEFT clearly shows that proposed EHA gives better results. EHA reduces the processing cost, number of virtual machines and increases the average processor utilization. But this improvement is achieved with a slight increase in makespan. For future work, an optimized algorithm can be implemented that further reduces the total completion time. Dynamic nature of virtual machines can be addressed and accordingly adaptive solutions can be proposed. This algorithm used only computation cost. In future, this algorithm can be extended by integrating computation as well as communication cost for scheduling. To represent the real-time environment, dynamic tasks can be used with varying incoming time.

## 6. REFERENCES

1. Arabnejad, H., & Barbosa, J. G. (2014). List scheduling algorithm for heterogeneous systems by an optimistic cost table. IEEE Transactions on Parallel and Distributed Systems, 25(3), 682-694.
2. Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (ga) based load balancing strategy for cloud computing. Procedia Technology, 10, 340-347.
3. Ge, J. W., & Yuan, Y. S. (2013). Research of cloud computing task scheduling algorithm based on improved genetic algorithm. In Applied Mechanics and Materials (Vol. 347, pp. 2426-2429). Trans Tech Publications.
4. Goyal, P., & Kaur, N. (2015). An optimizing technique based on genetic algorithm for power management in heterogeneous multi-tier web clusters. International Journal of Computer Applications, 115(17).
5. Hamad, S. A., & Omara, F. A. (2016). Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment. International Journal of Advanced computer Science and Applications, 7(4), 550-556.
6. https://download.pegasus.isi.edu/misc/SyntheticWorkflows .tar.gz
7. Hu, J., Gu, J., Sun, G., & Zhao, T. (2010, December). A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on (pp. 89-96). IEEE.
8. Jang, S. H., Kim, T. Y., Kim, J. K., & Lee, J. S. (2012). The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation, 5(4), 157-162.
9. Kashyap, D., & Viradiya, J. (2014). A survey of various load balancing algorithms in cloud computing. International Journal of Scientific and Technology Research, 3(11), 115-19.
10. Kaur, G. (2016). A DAG based Task Scheduling Algorithms for Multiprocessor System-A Survey. International Journal of Grid and Distributed Computing, 9(9), 103-114.
11. Kaur, R., & Kinger, S. (2014). Enhanced genetic algorithm based task scheduling in cloud computing. International Journal of Computer Applications, 101(14).
12. Jang, S. H., Kim, T. Y., Kim, J. K., & Lee, J. S. (2012). The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation, 5(4), 157-162.
13. Kashyap, D., & Viradiya, J. (2014). A survey of various load balancing algorithms in cloud computing. International Journal of Scientific and Technology Research, 3(11), 115-19.
14. Kaur, G. (2016). A DAG based Task Scheduling Algorithms for Multiprocessor System-A Survey. International Journal of Grid and Distributed Computing, 9(9), 103-114.
15. Kaur, R., & Kinger, S. (2014). Enhanced genetic algorithm based task scheduling in cloud computing. International Journal of Computer Applications, 101(14).
16. Maduravoyal, C. LIST BASED SCHEDULING ALGORITHM FOR HETEROGENEOUS SYSYTEM. International Journal of Applied Environmental Sciences (IJAES), 10(1), 2015.
17. Ravichandran, S., & Naganathan, E. R. (2013). Dynamic scheduling of data using genetic algorithm in cloud computing. International Journal of Computing Algorithm, 2(01), 127-133.
18. Sheng, X., & Li, Q. (2016, August). Template-Based Genetic Algorithm for QoS-Aware Task Scheduling in Cloud Computing. In Advanced Cloud and Big Data (CBD), 2016 International Conference on (pp. 25-30). IEEE.
19. Vanitha, M., & Marikkannu, P. (2017). Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines. Computers & Electrical Engineering, 57, 199-208.
20. Wang, T., Liu, Z., Chen, Y., Xu, Y., & Dai, X. (2014, August). Load balancing task scheduling based on genetic algorithm in cloud computing. In Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on (pp. 146-152). IEEE.
21. Xiao, Z., Song, W., & Chen, Q. (2013). Dynamic resource allocation using virtual machines for cloud computing environment. IEEE transactions on parallel and distributed systems, 24(6), 1107-1117.
22. Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. Information Sciences, 270, 255-287.

## AUTHORS PROFILE

**Salil Bharany**, Pursuing PHD, MTech Computer Science Engineering GNDU, Amritsar.
**Email**: salil.bharany@gmail.com