Hybrid Approach of Black Hole Algorithm and ACO for improve multi objective based dynamic workflow scheduling in Cloud Computing

[1]Archana N. Jethava, [2]Megha R. Desai
[1]Department of Computer, Silver Oak College of Engineering and Technology, Ahmedabad Gujarat India
jethavaarchana@gmail.com
[2]Department of Computer, Silver Oak College of Engineering and Technology, Ahmedabad Gujarat India
megha.rdesai1@gmail.com

Abstract : In cloud computing environment many scheduling algorithm are available to provide task scheduling easy, scalable and make task execution fast. In this paper we are implemented the hybrid structure of BHO and ACO Algorithm. For implementing dynamic workflow scheduling BHO algorithm. In scheduling Makespan and Cost optimization is main issue now days. These task scheduling algorithms are used by many cloud service providers and various researchers for static and dynamic task scheduling of many applications. Task scheduling mainly focuses on efficient utilization of resource and reduce the task Compilation time. Many different techniques are used to solve this problem of task scheduling. Here we provide propose flow and implementation work of proposed flow to reduce the QoS parameters like Makespan, Cost and Throughput. And evaluate the result in montage scientific workflow with size small, medium and large also make the comparisons of BHO with proposed algorithm.

Index Terms - **Black Hole Algorithm, Ant colony Optimization, Workflow Scheduling**

I. INTRODUCTION

Cloud Computing can be simply defined as computing services delivered to the user over the internet. Cloud computing is shared pool of resource in distributed environment. Cloud computing have four deployment models Public, private hybrid and community. Three services models IaaS, PaaS and SaaS and have five main characteristic.it provide resource sharing using distributed network by common internet protocols and network standards. Cloud computing provide pay - per – use service and this provider cloud service is called as service provider of cloud. For example Amazon, Microsoft, Google are cloud service provider.[1]

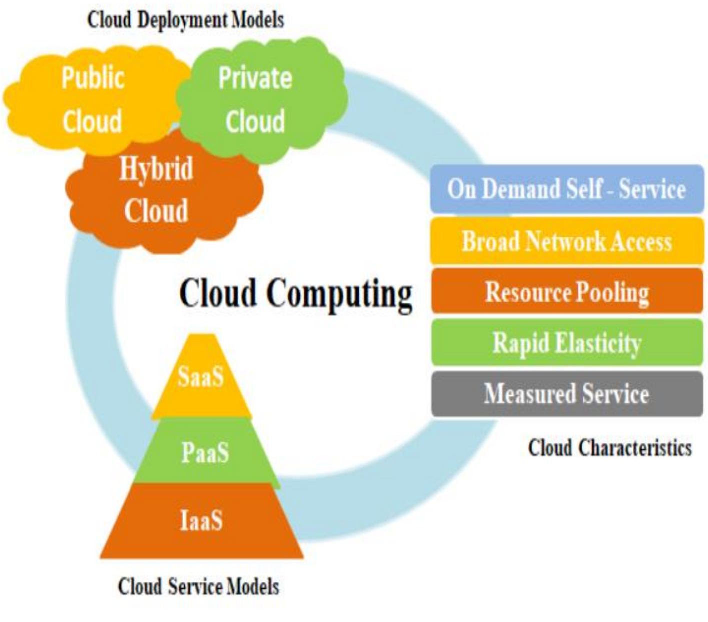

Figure 1 Overview of Cloud Computing

1.1 Workflow Scheduling in Cloud Computing

Workflow scheduling is the development of appropriate resource to the particular job in particular time. A workflow represents a process as consisting of a series of steps that simplifies the complexity of execution and management of applications. Workflow technology constitutes a common model for de-scribing a wide range of scientific applications in distributed systems [2]. It has an necessary part in the Processes of many elementary science fields, such as physics, chemistry, biology, and computer science. The interest for a workflow comes mainly from the use to build upon tradition codes that would be too costly to rewrite. It is collectively denoted by a Directed Acyclic Graph (DAG) in which nodes denote compute tasks and edges represent precedence and flow bind between tasks.

Scientific workflows utilize appropriate resources in order to access, manage, and process a large amount of data from a higher level. Processing and managing such large amounts of data require a distributed collection of computation and repository facilities. These resources are often limited and common among many competing users. Allocating suitable resource to the workflow task and each task is arranged as workflow.

1.2 Classification of Workflow Scheduling Algorithm

The scheduling can be distinguished as Static Scheduling and Dynamic Scheduling.[3]

1) Static Scheduling Algorithm:
In Static Scheduling all information are known to scheduler about tasks and resources before starting execution. It has less runtime overhead. The resource allotment and scheduling workflow problems in cloud environments. This strategy is based on three complementary bi-criteria access. It takes into account the global execution time and the cost incurred by using a set of systems and resource. The first approach enables to decrease the execution cost. The second attempts to minimize the total completion time. Finally, the third approaches associate the objectives of the above techniques by selecting only the non-dominated results.

2) Dynamic Scheduling:
In Dynamic Scheduling task is not known before the stating of execution it is known at run time. And the task execution time may not be known before execution starts. It has more runtime overhead. Dynamic algorithms consist in measuring resources and scheduling tasks at runtime to take into reach the dynamic aspect of the cloud. They recollecting regularly. the systematic solutions in order to minimize the execution cost based on the present network conditions and resources. We describe in the first part several approaches that are based on dynamic algorithms for scheduling a single workflow. Then, we address those destined for a set of workflows.

II. PROPOSED WORK

Here we proposed the flow of the proposed work by modifying some steps in Black Hole Algorithm to provide dynamic workflow management using the ant colony Optimization algorithm.

A. Black Hole Algorithm

Black hole algorithm has been first presented in 2013 by Hemmatloo [4]. Black hole algorithm is a population-based algorithm and has some common features in other population based algorithms. Compared with other population-based algorithms, In This Black hole algorithm proposed that, the evolution of the population is done by finding the fitness of candidate and the best fittest candidate is called as Black Hole and other candidates are the stars which moving around the search space. The absorption of the candidates towards the black hole is formulated as follows:

$$xi(t + 1) = xi(t) + rand \ast (xBH - xi(t)) \quad i = 1, 2,...N \quad (1)$$

Where xi(t) and xi(t + 1) are the locations of the ith star at iterations t and t + 1, respectively. xBH is the location of the black hole in the search space. rand is a random number in the interval [0, 1]. N is the total number of stars (candidate solutions).

B. Ant Colony Optimization

The ant colony optimization algorithm is a distributed Algorithm that is used to solve combinatorial optimization problems [5,6]. ACO algorithm is meta heuristic technique for finding shortest path to reach at destination.in this algorithm ant are finding path for reaching at the food. Ants are travel to the different path the pheromone is used to find the optimal path local updates and global updates of path is done based on the pheromone value. The shortest path has higher pheromone value and is updated as global path.

2.1 Flow of proposed algorithm

Proposed Flow represents the steps or working of proposed algorithm. which is the hybrid combination of black hole algorithm and ant colony optimization.in proposed flow first the initialization of task and resource is done after initialization it check for the fitness of task on particular virtual machine or in resource. Task is assign on best fittest virtual machines which gives the optimized task execution time and reduce the makespan of particular job. Task location is updated when all task is executed one by one and pheromone is update by the task run on which virtual machine at same time. last step of the proposed flow gives the final result in total makespan and total execution time of given task.

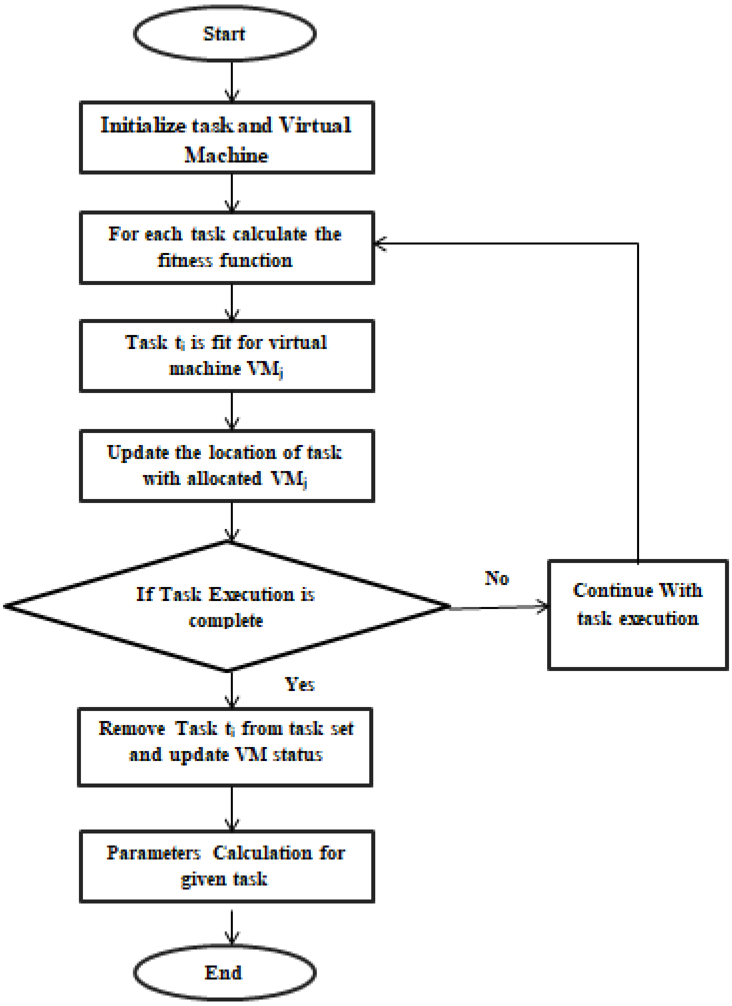

Figure 2 Flow Chart for Proposed Approach

2.2 Steps for proposed algorithm

Steps of proposed Algorithm represent the flow of algorithm implementation

Step 1: Start
Step 2: Initialize the task and virtual machine
Step 3: For each task calculate the fitness function
F = *min* ⬚*ni*=1 Execution time of task ti
Step 4: Check the task ti is fit for virtual machine VMj
Step 5: Update location of the task with allocated VMj
Step 6: If task execution is complete
Step 7: Remove task ti from task set and update the Virtual machine Status
Step 8: Else continue with task execution go to the step 3
Step 9: Parameter Calculations for the given number of task
Makespan = total task waiting time + total execution time
Total Cost = processing cost of task
Throughput = task size / task finish time
Average Utilization = task actual time /task processing cost)*100
Step 10: End

III. IMPLEMENTATION WORK

Implementation of proposed algorithm is done in WorkflowSim tool.WorkflowSim, is extends the existing CloudSim simulator by providing a higher layer of workflow management. We also indicate that to ignore system overheads and failures in simulating scientific workflows could cause significant inaccuracies in the predicted workflow runtime.[11]

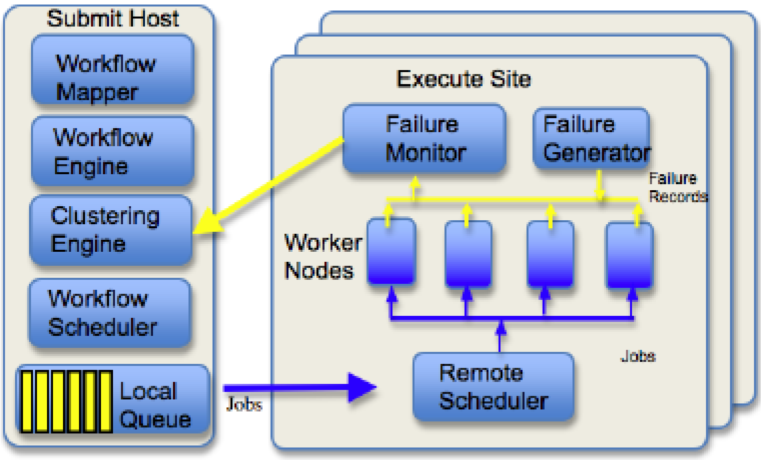

Figure.3 Overall Structure of Workflow Simulator [11]

Workflow management system which contains: a Workflow Mapper to map abstract workflows to concrete workflows that are dependent on execution sites; a Workflow Engine to handle the data dependencies; and a Workflow Scheduler to match jobs to resources. Other components include a Clustering Engine that merges small tasks into a large job, a Provenance collector that tracks the history of task/job execution and a Workflow Partitioned that divides the user workflow into multiple sub-workflows

Here we calculate the makespan and cost for the different type of scientific workflow.

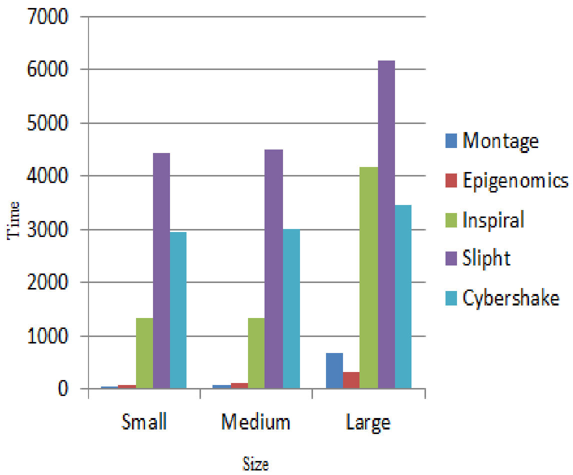

Figure.4 Makespan Calculation in Different type of workflow

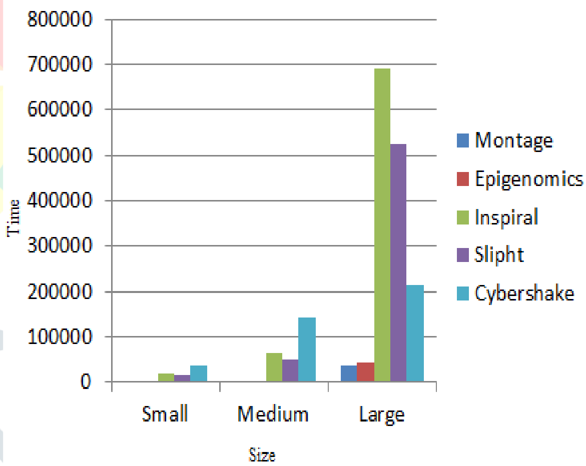

Figure.5 Total Cost Calculation in Different type of workflow

Figure.4 and 5 shows the comparison of makespan and cost in different scientific workflow such as montage, epigenomic, inspiral, slipht and cybershake. In montage and Epigenomic workflow it gives less makespan and less total cost for small, medium and large size in compare with other workflow.

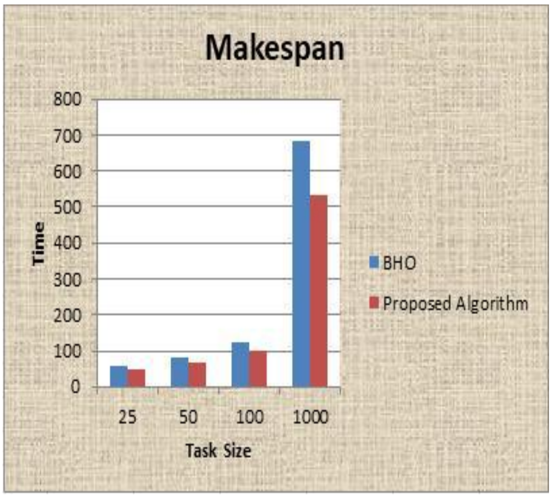

Figure 6: Makespan Results Comparison

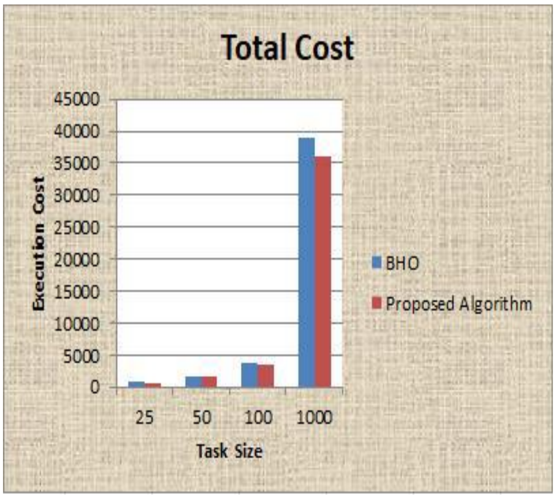

Figure 7: Total Cost Results Comparison

Figure 6 shows the result of makespan comparison of the proposed algorithm and base paper algorithm in different size of workflow. And Figure 7 shows the result of total cost comparison of the proposed algorithm with base paper algorithm in different size of workflow.

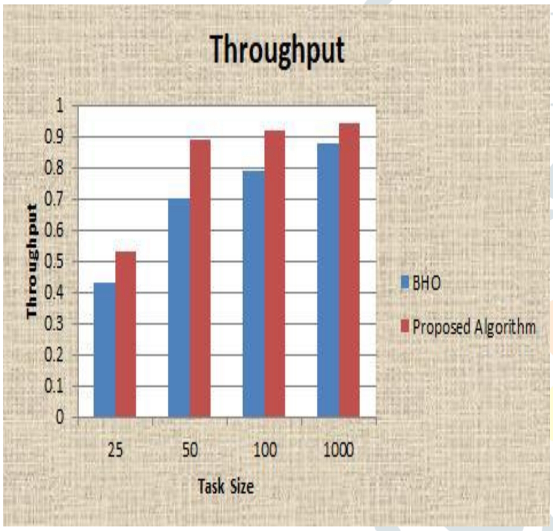

Figure 8: Throughput Results Comparison

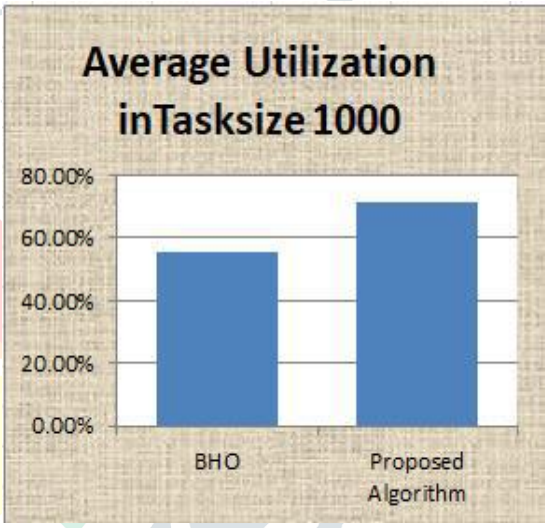

Figure 9: Average utilization in task size 1000

Figure 8 shows the result of throughput comparison of the proposed algorithm with base paper algorithm in different size of workflow. And Figure 9 shows the result of average utilization comparison of the proposed algorithm with base paper algorithm of the workflow size 1000. Result comparisons are calculated in percentage.

IV. CONCLUSION

In this paper we have analysed and compared Existing approaches for scheduling workflows in the cloud.. The elastic nature of cloud environment enables such dynamic workflow to be enacted more efficiently since it facilitates the changing of resource quantities at runtime. Depending on the requirements of the workflow, resources can be added or released at runtime on demand. Thus, the cloud may be used as an effective accelerator to treat changing computational requirements as well as that of QoS constraints (e.g., deadlines) for a dynamic workflow. It can improve the application time-to-completion and handle unexpected situations. In contrast, using an environment with fixed resources leads to a poor performance but since the task scheduling is heuristic problem the more research can be done in this field and more optimized solutions can be achieved.by implementing proposed algorithm it gives less makespan and less total cost in some scientific workflow like montage and Epigenomic.

REFERENCES

1. Barrie Sosinsky, "Defining CloudComputing" in Cloud ComputingBible, Indianapolis, Indiana: Wiley Publishing, Inc. 2011.
2. E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Work flows and e-Science: An overview of workflow system features and capabilities," Future Generation Computer Systems, vol. 25, no. 5, pp. 528–540, 2009.
3. Henzinger, Thomas A., et al. "Static scheduling in clouds." memory 200.o1 (2011): i1.
4. Fatemeh Ebadifard and Seyed Morteza Babamir , "Optimizing Multi Objective Based Workflow Schedul ing in Cloud Computing Using Black Hole Algorithm", 3th International Conference ON Web Research ,2017.

5. Priyadarsini RJ, Arockiam L. "PBCOPSO: A Parallel Optimization Algorithm for Task Scheduling in cloudenvironment." Indian Journal of Science and Technology2015; 8(16):1–5.
6. Liyun Zuo,Lei Shu, chunsheng Zhu,Takahiro Hara,"multi – objective optimization scheduling method based on Ant Colony Algorithm in cloud computing",IEEE December 23,2015.
7. Lavanya M, Janani TN, Sushmita S, Sunandha M, Vaithiyanathan V, Saravanan S. "A constraint-based decentralized task routing approach to large-scale scheduling in cloud environment". Indian Journal of Science and Technology. 2015; 8(35):1–4.
8. Zhang F, Cao J, Tan W, Khan SU, Li K, Zomaya AY. "Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud". IEEE Transaction on Emerging Topics in Computing. 2014; 2(3):338–51.
9. Rodriguez MA, Buyya R. "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds". IEEE Transactions on Cloud Computing. 2014; 2(2):222–35.
10. Anita chaudhari,M.C.Govil "Workflow Scheduling Algorithms in Cloud Environment: a Review, Taxonomy,and Challenges",IEEE conference on parallel,distributed and grid computing ,2016.
11. Weiwei Chen and Ewa Deelman ," WorkflowSim: A Toolkit for Simulating Scientific Workflow in Distributed Environment", IEEE 978-1-4673-4466-1/12,2012.