Hybrid Genetic Algorithm for Solving the Knapsack Problem in the Hybrid Cloud

Asish Srivastava, Tinesh Kumar Goyal, Amitesh Kumar Gupta, Deeapk Gaur

***Abstract*— Cloud computing is a concept which allows its users to work with a pay-as-you-go model. This request-response coupling needs to be managed, using resource allocation strategies, as there can be fluctuating request volume as well as unpredictable request arrival patterns. Therefore, it is very important to dynamically allocate resources, as it is not predictable when a cloud user logs in or logs out of the cloud environment. Knapsack problem is a NP-complete problem, so we take it for solving cloud based solution. We used several benchmarks to validate our solutions against the optimal solution computed by dynamic programming. Results are very liberal of promises and exemplify reasonable scalability as well as speed up factor for our implementation using Eucalyptus..**

***Index Terms*— Cloud Computing, Knapsack Problem, IaaS, PaaS, SaaS, genetic Algorithm, Virtualization, Eucalyptus.**

I. INTRODUCTION

By Cloud computing has been emerging as a pliable and sledgehammer computational architecture to attempt universal services to users. It adapt inter connected hardware and software resources in an integrated way, which is different to traditional computational environments. A diversity of hardware And software resources are integrated together as a pool of resource, the software is no longer inhabit in a single hardware atmosphere, it is featured upon the schedule of the resource pool for optimized resource utilization. Large data centres permit resource sharing across hosted applications and lead to economies of scale at both the hardware and software level. Software services can get probably endless scalability and incremental growth to meet customers' resilient request. The pay-as-you-go model and agile provisioning can result in more efficient resource utilization and reduced costs.[1]Cloud computing provides access to IT resources as services ranging from direct access to hardware equipment to more sophisticated applications. According to this definition, one can distinguish between three levels of Cloud services, namely infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). [4]

Cloud is a centralized pool of resources. In cloud Computing client access the resources form the centralized server. It may be Hardware of software. It follow pay as you go concept. So every time no of client request fluctuated, because of this fluctuation we need to find optimal resource distribution strategy. So we take this problem as Knapsack problem. Knapsack problem is known NP Complete problem. It is very hard to find optimal solution. Knapsack problem includes solutions to find an optimal set of objects from a finite set of available objects. In cloud computing we find an optimal resource distribution from pool of resource. Knapsack solution have been applied to a wide range of areas involving operational research, computer science, cryptography, Spanning tree, eight queens' puzzle and Knapsack have been recognized among well-known problems of combinatorial optimization. For instance, we apply knapsack problem for resource allocation algorithms in Cloud environment with presence of several Quality of Service and other constraints. [2] The knapsack problem difficulties is directly proportional to size of knapsack, This statement give the clear idea if knapsack size increased then complexity of finding optimal solution is also increased.

we have many well known algorithm for solving knapsack problem i.e. dynamic programming, Greedy algorithm, Backtracking, Meet-in-the-middle. Knapsack problem solving algorithm have two major drawbacks:(1) Algorithm are usually very time consuming (2)they do not give always optimal solution. In consequence, it motivates researchers to discover other algorithms to solve this NPC problem, hopefully they will better built up and legitimate also.

A. The Architecture of the Cloud Service Models

1. Software as a Service: In SaaS Vendor provides a software service to user when user demands the service. (I.e. Google drive)[1][4][5]

2. Platform as a Service: In PaaS vendor Provide platform to the user to develop application. Vendor Provide some supporting software also like Database, API's etc. [1] [4]

3. Infrastructure as s Service: In IaaS Vendor provide to user only infrastructure. [1][4]

Manuscript received April 24, 2014

Asish Srivastava, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-9555531144.

Tinesh Kumar Goyal, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-9417170800.

Amitesh Kumar Gupta, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-7532836750.

Deepak Gaur, Presently he is working as an Assistant Professor in CSE Department, Amity University, Noida, Uttar Pradesh, India. His Research area includes Natural Language Processing, Data mining. +91-9873905675,

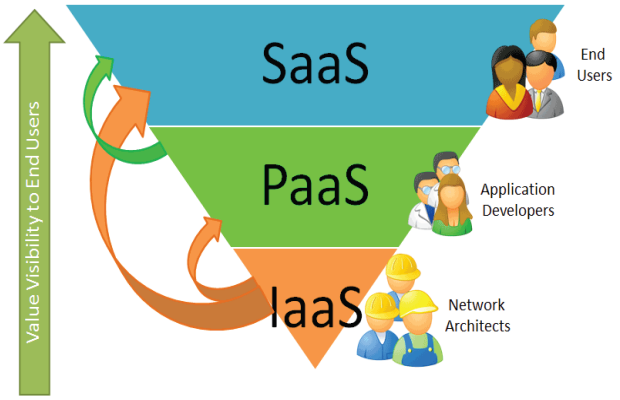

Figure 1: Cloud Service Models

Genetic Algorithm (GA) is imitator the process of raw selection. This heuristic search is applied to produce relevant solutions to optimization and search problems. Genetic algorithms pertained to the evolutionary algorithms (EA).Genetic Algorithm (GA) is a durable algorithm. It was presented as an alternative algorithm to illustrate Knapsack problem. [8]

In genuine real time puzzle numerous interruptions exist in knapsack. I proposed a hybrid algorithm. Hybrid Genetic Algorithms (HGA) has also been considered as an appropriate alternative. It has been shown in my recent studies that HGAs have better performance, although might be very challenging to implement using current cloud infrastructure. It easily implemented by large enterprise or organizations private or hybrid cloud infrastructure. [2][7][8][9].

In this paper, we attempt to solve the Knapsack problem with our HG Algorithm in the Private/hybrid Cloud, a technology that moves computing and data away from our desktop and portable PCs into large data canters. We implemented genetic algorithm in cloud for finding optimal solution. In a simple Hybrid genetic algorithm (HGA) in the Cloud is presented which called SMRHGA and implemented based on Hadoop. [2][7][9] However, none of these works attempt to solve any combinatorial optimization solutions with the HGA in the Cloud environment. Hence, our main donation in this paper is to present a scalable framework that deploys HGA to solve the difficult Knapsack problem in the Cloud.

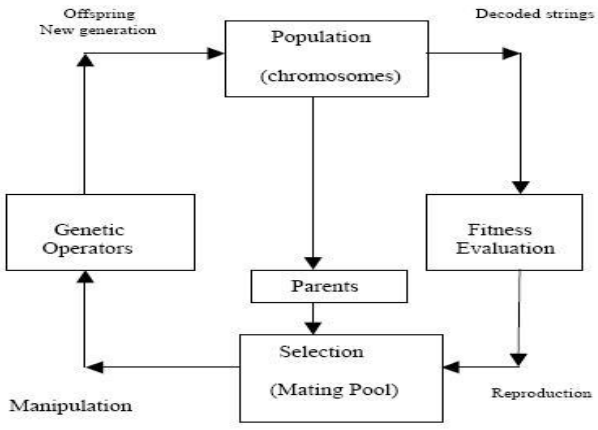

Figure 2: Flow Chart of Genetic Algorithm

This paper is organized as follows. Problem statement discuss in Section II, basics of a generic GA, and basics of Eucalyptus as our Cloud environment. We will present our system Design in Section III. Section IV includes the outcome of our study; Section presents discussion followed by conclusion in Section V.

II. Problem Statement

A. The Knapsack Problem

Knapsack problem, as a famous NP problem and can be defined as follow. Assume a Knapsack with the maximum capacity of C, also assume a set of N items/objects (I = 1... N), each with the profit value of Pi > 0 and weight Ri > 0. The objective of the problem is to select a subset of N items such that the total profit value of the selected items is maximized, while their accumulative weight does not exceed C. knapsack problem can be mathematically. Formulated as follows where Qi ∈ {0, 1} [5]:

. $$\text{MAX} \sum_{i=1}^{k} \text{Pi} \times \text{Qi} \;\; \text{s.t.} \;\; \sum_{i=1}^{k} \text{Ri} \times \text{Qi} \leq \text{C}$$

B. Genetic Algorithms

Genetic algorithm is a mirror of heuristic search. It is a copy of ordinary selection. Heuristic search is usually applied to construct significant result to improve search problems. Many algorithms are autonomously subsisting to determine knapsack problems, based on dynamic programming approach, branch and bound approach or mixture of both approaches. [7] [8]

In a genetic algorithm, we take an example of populace, we have various solutions. Every solution has some properties (chromosomes or genotype) which can be altered and mutated; generally, solutions are articulated in binary (strings of 0s and 1s,) but other encodings are also approachable. [8][9].

C. Eucalyptus

Eucalyptus software is open source by the nature. Eucalyptus software allows you to build AWS- consistent private clouds, from single machines to many of cores. Eucalyptus combines your compute, storage, and other resources into an on-request, volunteering private cloud that is highly compatible with AWS. The Eucalyptus cloud is highly scalable due to its distributed nature. Split flair. The private cloud tectonics is included three separate layers. [3][6]:

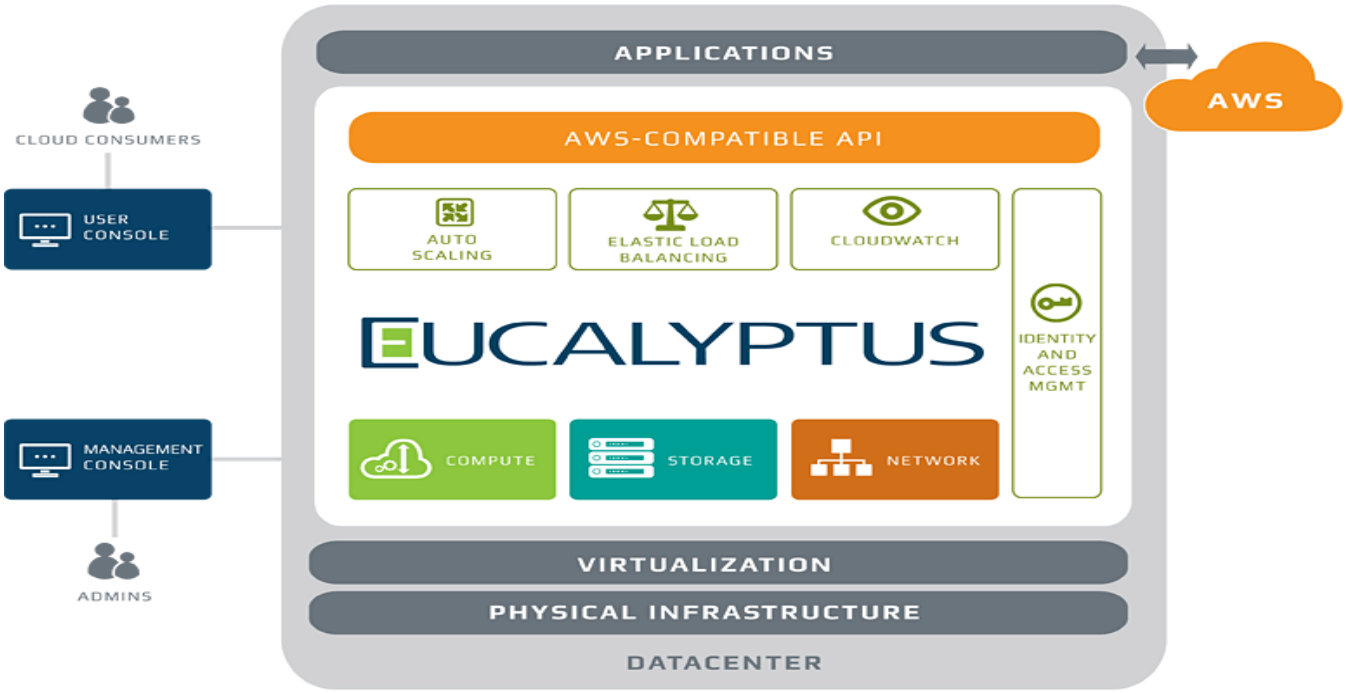

Figure 3: Eucalyptus Architecture

III. SYSTEM DESIGN

As mentioned, we used eucalyptus to solve the knapsack problem using our cloud. We use Hybrid Genetic algorithm for solve knapsack problem.

Hybrid Genetic Algorithm structure: Hybrid GA-KS uses one cloud Controller, walrus, storage controller, two queues and four tables to perform its computational tasks.

The *Cloud controller* is written in Java program that provides us EC2-compatible SOAP and interpellation interfaces, and Web interface to the outside people. Therewith serving inward requests, the Cloud Controller work like the prefectural interface in order to cloud supervision and carrying out advance-degree of system accounting and resource scheduling. Cloud Controller recognize client from monitoring phalanx interfaces (command line) like GUI-based or euca2ools tools similar to User Console of eucalyptus and run the main storage, network and compute resources. In Eucalyptus cloud sole CLC allowed. Cloud Controller is responsible for quote management, authentication, reporting and accounting.

Walrus, inscriptive in Java programming language, is the Eucalyptus compeer to Amazon Web S3.S3 stands for simple storage service. The Walrus grant tenacious stockpile to each of the VM's inside the cloud of eucalyptus. It can be consumed as an easy HTTP get/put operation on cloud storage. Walrus does not have any inhibition for data type, and it can store operating system images (i.e. the images used to launch VM's, storage segment snapshots, and data of application. In Eucalyptus cloud sole walrus allowed.

Cluster Controller inscriptive in C Language and CC acts as the frontage for a cluster Eucalyptus cloud and inform to the SC (Storage Controller) and NC (Node Controller). It manages the execution of virtual machine and SLAs on every cluster.

Storage Controller inscriptive in Java Language. The Eucalyptus is identical to AWS EBS. It spells out with the CC and NC. Storage Controller manages Eucalyptus storage and snapshots of the instances inside its distinct cluster. If an instance goes for inscriptive tenacious data to reminiscence outward of the cluster, then it would require inscribing to Walrus, it can be reside in anyone instance in anyone cluster.

Queues: When the Web Role uploaded a file, it sends a message to the Job queue to alert Worker Roles that a task needs to be accomplished. When one of the Worker Roles receives such message, it sends other messages using also the Job queue to inform other Worker Roles as well. It then handles the task and assigns subtasks to other workers through the Chromosomes queue; other Worker Roles always listen to this queue to perform tasks.

Tables: Several tables are used in our framework. Item Table is used to store items of the uploaded file. Thus, Worker Roles can read items from this table. Chromosomes Table is used to store chromosomes created by Worker Roles. The information of a chromosome contains the iterations number, chromosome ID, total weight of the chromosome, and the total value of the chromosomes. Worker Roles work in parallel whilst listening to queue messages to generate chromosomes for this table. File Table is used to store URL link of the submitted files as well as their solution file along with the duration time to complete the solution. History

Solution Table is used to record history of solutions observed from multiple generation phases of the HGA-KS. History solution is used to analyze the performance of HGA -KS.C. Hybrid Genetic Algorithm Based on our algorithm, a submitted file is received by the cloud controller and then stored in the blob. The Web Role sends a message to the Job queue to inform Worker Roles of the arrival of a new task. All Worker Roles listen to the Job queue until one of them receives the message and becomes the job handler for the given task. In order to process the task in parallel, the job handler will send messages to other Workers to inform them that they are the normal workers and their job is to process subtasks only. Other Worker Roles start listening to another queue, called Chromosomes queue, and start processing the assigned subtasks. The job handler sends messages to Chromosomes queue to inform the workers to create the chromosome. Each message carried the information of the chromosomes from the last population to be used to generate chromosomes for the new population.

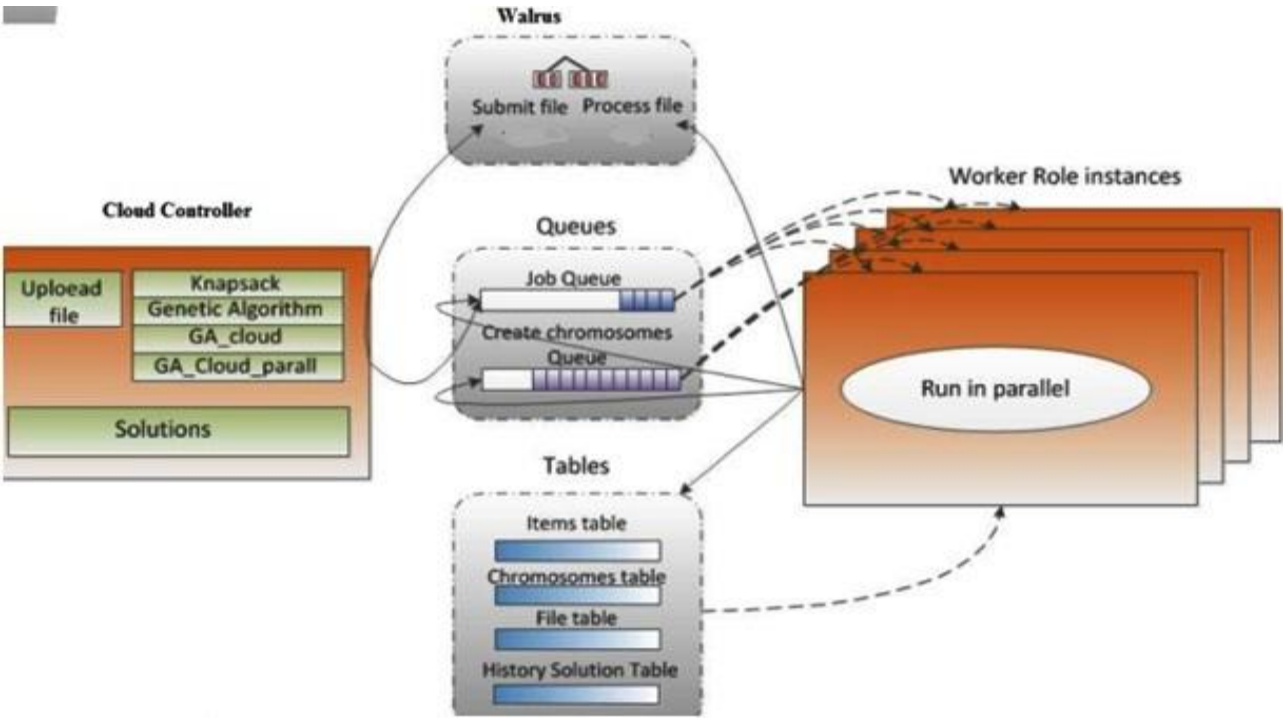

Figure 4: Hybrid Algorithm Structure

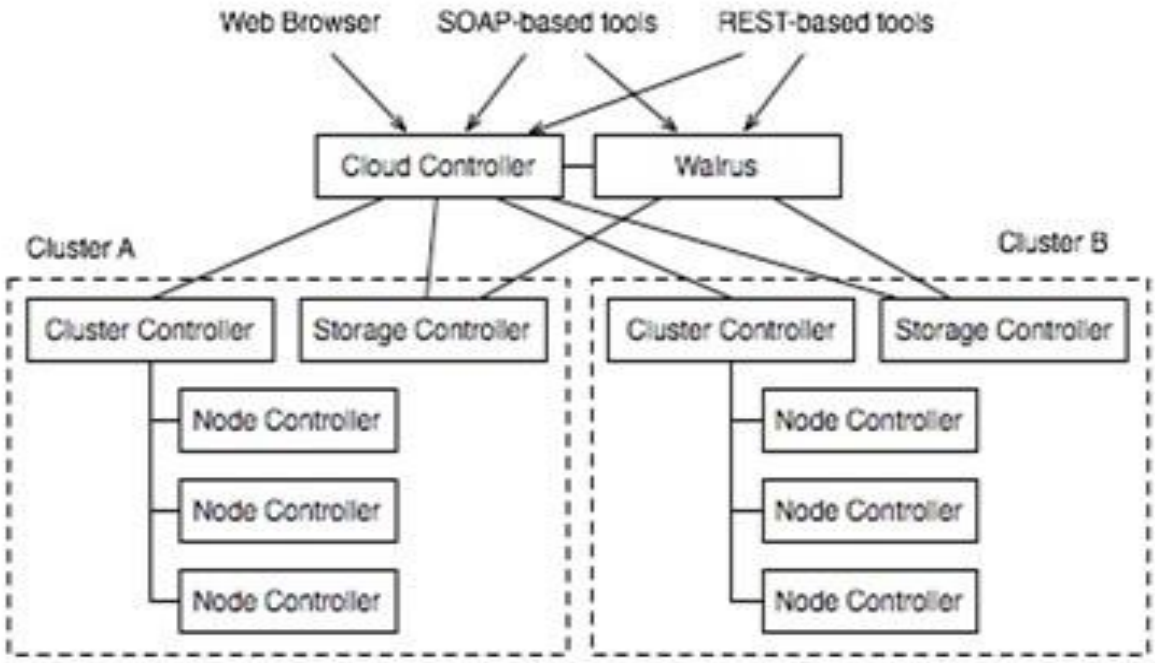

Figure 5: Eucalyptus Component

Hybrid Algorithm for Knapsack Problem

Input: Submitted File (in-File)
Output: Solution File (out-File)
Begin

- CC stores in-File to a walrus;
- CC sends a message to Job Queue;
- One of the CC picks the message from job Queue and becomes Job Handler;
- Job Handler sends other message into Job Queue to inform other Worker Roles;
- Job Handler initializes the History Table;
- Job handler reads In-File and produce sub task and sends to chrQueue to generate the initial population;
- Let Itr ← 0;

While (Itr < Max Number of Iterations) **do**

- Job Handler produces sub tasks from the CurPop and to chrQueue. These sub tasks involve elitism, crossover and mutation operators to combine chromosomes of the current population and produce the next one;

While (ChrQueue is not empty**) do**

- Worker Roles perform Sub task in ChrQueue;

End

- Job Handler Collects results of all sub tasks and update solutions in Curpop;
- Job Handler update the History Table;
- Itr ←Itr +1;

End

- Job Handler Stores results into out-File;

End

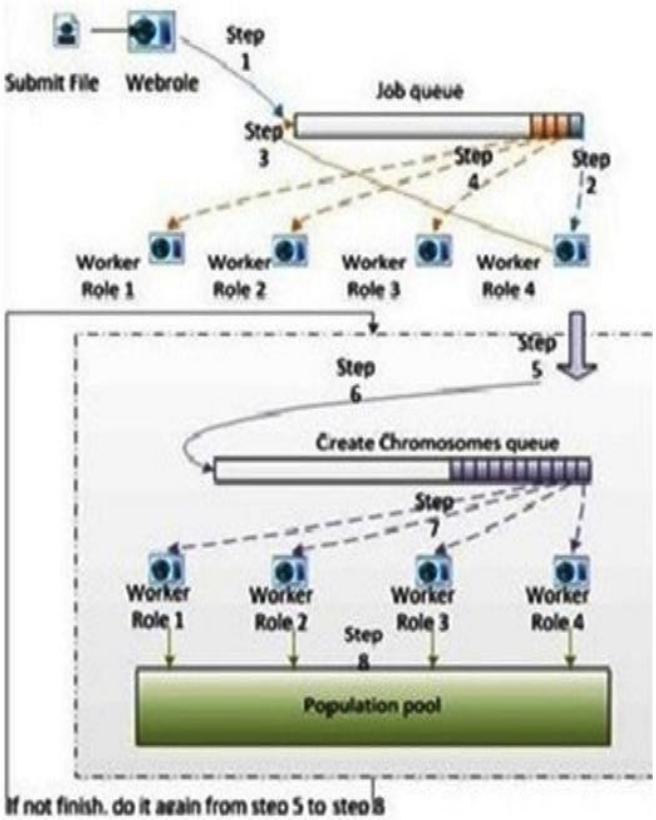

Figure 6: Hybrid GA Working

IV. DISCUSSION AND ANALYSIS

Its performance can be greatly influenced by number of Factors. We highlighted our observation through the following subtopics.

A. Accuracy of Hybrid Genetic Algorithm

Hybrid genetic algorithm gives optimal solution in all cases. The performance of HGA is satisfactory, when the knapsack size is increased.

B. Execution Time of Hybrid Genetic Algorithm

Hybrid Genetic algorithm takes more time when the knapsack size increased, but knapsack size decreased Execution time also decreased. HGA find the optimal solution in all cases so the performance of Hybrid Genetic Algorithm is satisfactory.

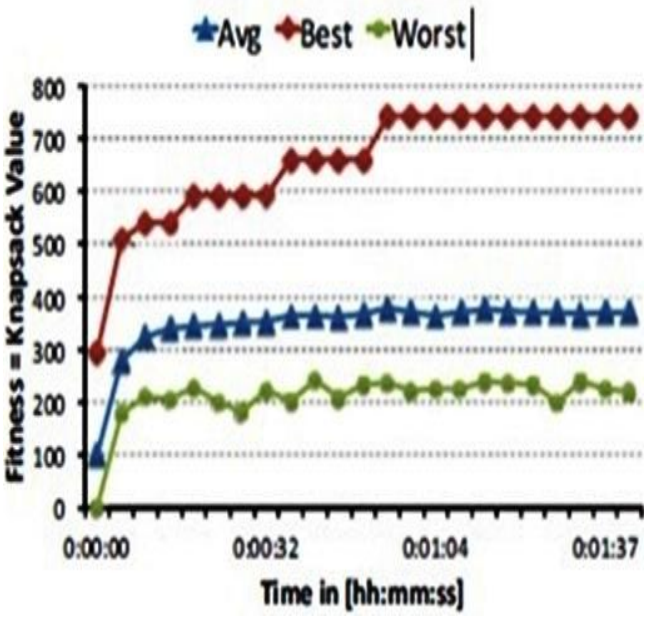

Figure 7: A Simple Hybrid GA Solving History

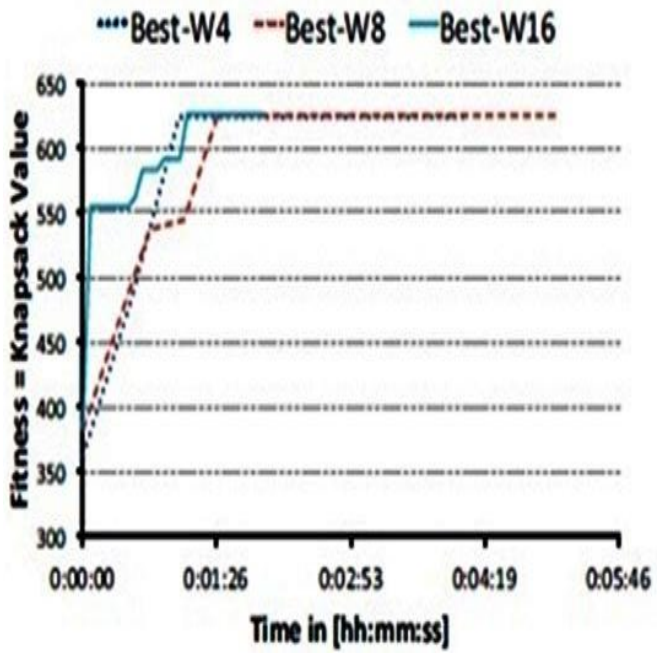

Figure 8: Execution time of HGA

V. CONCLUSIONS AND FUTURE WORK

Knapsack problem is a combinatorial optimization. It is used to find the optimal solution from real world Problem. In this paper, we use Eucalyptus software which manage the resource pool and try to find optimal solution. Knapsack is a NP Complete problem .In this paper we introduce an approach to solve knapsack problem in the cloud environment. Our solution is a scalable Hybrid Genetic Algorithm (HGA-KS) designed and implemented in Eucalyptus IaaS, PaaS environment.

Our experimental result shows that HGAKS Processed job from job pool and achieved High speed when executed. HGA-KS was also Give the optimal solution when resource pool is very big. HGA execution time is directly proportional to knapsack size. If knapsack size increase increased then the time to find optimal solution also increased. In this paper we simulate the hybrid algorithm and check the performance of the algorithm. The performance of Hybrid Genetic Algorithm is satisfactory.

Another flank for our subsequent work is to use Open Stack and solve the knapsack problem using modern Database in IaaS, PaaS, and SaaS environment..

REFERENCES

[1] Barrie Sosinsky, "Cloud Computing Bible," 1th ed, January 11, 2011.
[2] M. Hristakeva and D. Shrestha. Solving the 0/1 knapsack problem with genetic algorithms. In Science & Math Undergraduate Research Symposium, 2004.
[3] http://en.wikipedia.org/wiki/Eucalyptus_%28software%29.
[4] Rajnish Choubey et al., International Journal on Computer Science and Engineering (IJCSE), 2011.
[5] R. Buyya, C. S. Yeo, and S. Venugopa, "Marketoriented Cloud Computing: Vision, hype, and reality for delivering it services as computing utilities", in Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08, IEEE CS Press, Los Alamitos, CA, USA) 2008.
[6] https://www.eucalyptus.com/resources/cloud-overview.
[7] P. Chu and J. Beasley. A genetic algorithm for the multidimensional knapsack problem. Journal of heuristics, 4(1):63–86, 1998.
[8] S. Khuri, T. B¨ack, and J. Heitk¨otter. The zero/one multiple knapsack problem and genetic algorithms. In Proceedings of the 1994 ACM symposium on Applied computing, SAC '94, pages 188–193, New York, NY, USA, 1994. ACM.
[9] K. Li, G. Dai, and Q. Li. A genetic algorithm for the unbounded knapsack problem. In Machine Learning and Cybernetics, 2003 International Conference on, volume 3,pages 1586–1590. IEEE, 2003.

Asish Srivastava Pursuing Master of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India. Area of Interest: Cloud Computing, Virtualization, Computer Networks and Network Security.

Tinesh Kumar Goyal Pursuing Master of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India, Area of Interest: Cloud Computing, Computer Networks and Software Engineering.

Amitesh Kumar Gupta Pursuing Masters of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India, Area of Interest: Computer Networks, Network Security, Cloud Computing, Data Mining and Warehousing.

Deepak Gaur is working as an Assistant Professor In CSE Department, Amity University, Noida, Uttar Pradesh, India. His Research area includes Natural Language Processing, Data mining.