Optimal Scheduling and Load Balancing in Cloud using Enhanced Genetic Algorithm

Kiranveer Kaur
M.Tech Research Fellow,
Department of Computer
Science Engineering,
Sri Guru Granth Sahib World
University, Fatehgarh Sahib,
Punjab

Amritpal Kaur
Assistant Professor,
Department of Computer
Science Engineering,
Sri Guru Granth Sahib World
University, Fatehgarh Sahib,
Punjab

ABSTRACT

The cloud computing is the enlargement of distributed computing, equivalent computing and gridiron computing, or defined as the commercial achievement of these computer science concepts. One of the elementary issues in these circumstances is interrelated to task scheduling and Load Balancing. Cloud task arrangement is an NP-hard optimization dilemma, and numerous meta-heuristic algorithms have been anticipated to crack it. A superior task scheduler should acclimatize its arrangement stratagem to the varying situation and the types of tasks. This manuscript proposes a cloud task arrangement course of action based on Load Balancing Enhanced Genetic (EGA) algorithm. The major involvement of our exertion is to balance the whole system load although trying to minimizing the Makespan of a prearranged tasks set. The innovative scheduling strategy was simulated using the Net Beans toolkit package. Experiments results showed the proposed Enhanced Genetic (EGA) algorithm and Compare the EGA, ACO.

General Terms

Cloud Computing, Genetic algorithm (GA), Ant Colony algorithm (ACO), Load Balancing,

Keywords

Load Balancing; Fitness, maximum iteration, Population Scale, Virtual machine.

1. INTRODUCTION

Cloud computing is associate on demand service during which shared resources, data, package and alternative devices are provided per the clients demand at precise time. Cloud could be a figure term used for Internet. The whole Internet is often seen as a cloud. A user has to be compelled to pay just for the usage resources per time.

Distributed computing is the utilization of the pooled registering assets open over Internet. Registering assets can be equipment or programming. Cloud gets its name from the cloud molded image speaking to Internet, as it is utilized as a deliberation for its unpredictable base. It gives benefits according to necessity. It permits client to redo, arrange, and convey cloud administrations. It offers benefits as per instalment. Cloud gives assets over Internet utilizing virtualization modernization, multi-tenancy, web administration, and so on. Virtualization gives reflection of autonomous equipment access to each virtual machine. Multi-tenure permits the same programming stage to be shared by different applications. Multi-occupancy is essential for creating programming as an administration application. Applications impart over the Internet utilizing web administrations [1]. Burden adjusting is one of the focal issues in distributed computing. It is a component that conveys the dynamic nearby workload equally over all the hubs in the entire cloud to dodge a circumstance where a few hubs are intensely stacked while others are sit still or doing little work. It serves to accomplish a high client fulfilment and asset usage proportion, subsequently enhancing the general execution and asset utility of the framework. [1].

As cloud computing is in its evolving stage, therefore there are several issues current in cloud computing [2]. Such as:

- Ensuring correct access management (authentication, authorization, and auditing)
- Network level migration, in order that it needs minimum value and time to maneuver employment.
- To offer correct security to the info in transit and to the info at rest.
- Knowledge availableness problems in cloud.
- Legal quagmire and transitive trust problems
- Data lineage, knowledge origin and unintended speech act of sensitive data is feasible.

And also the most current issue in Cloud computing is that the problem of load balancing.

There are four arrangement models of cloud.

Open Cloud: Public cloud makes administrations, (for example, registering, stockpiling, application, and so on.) accessible to general open. These administrations may be free or offered as installment as per utilization. Real open cloud suppliers are Amazon, Google, Microsoft, and so forth.

Private Cloud: Private cloud is a cloud foundation worked just for a solitary association. It is not accessible to overall population.

Community Cloud: Community cloud shared infrastructure between several organization with common concerns such as compliance, jurisdiction, etc.

Hybrid Cloud: Hybrid cloud is a combination of two or more clouds (public, private, or community).

2. LOAD BALANCING

Cloud computing is inexpensive and scalable however maintaining the immovability of processing such that a huge quantity of jobs surrounded by the cloud computing environment could be a terribly advanced drawback with load balancing receiving a lot of attention for researchers. Load balancing within the cloud computing setting has a very important impact on the performance. Sensible load balancing makes cloud computing a lot of economical and improves user satisfaction. Load balancing in clouds could be a mechanism that distributes the dynamic native work equally across all the nodes, ensuring that no solitary node is weak, thus raising the general performance of the system.

Load balancing may be a comparatively latest modus operandi that facilitates networks and assets by provided that a utmost throughput with lowest reaction time. Correct load balancing will facilitate in utilizing the accessible resources optimally, thereby minimizing the resource consumption. Dividing the traffic between servers, information is sent and received while not major delay. Absolutely diverse types of algorithms are accessible that helps traffic overloaded between accessible servers. A basic example of load balancing in our lifestyle is associated with websites. While not load balancing, users might expertise delays, timeouts and doable long system responses. Dividing the traffic between servers, information is sent and received while not major delay. There are various load balancing Bio-inspired algorithms like Genetic Algorithm, Ant Colony Optimization, Particle Swam Optimization etc that are used to balance the load [3].

In general, load balancing algorithms follow 2 major classifications [2]:

- Depending on however the charge is distributed and how processes allotted to nodes (the system load).
- Depending on the data status of the nodes (System Topology).

In the 1st case it designed as designed as centralized approach, distributed approach or hybrid approach within the second case as static approach, dynamic or reconciling approach.

2.1 Classification consistent with the System Load

- **Centralized approach:** during this approach, one node is to blame for managing the distribution at intervals the whole system.
- **Distributed approach:** during this approach, every node severally builds its own load vector by collection the load information of alternative nodes. decision are made locally using local load vectors . This approach is a lot of appropriate for widely distributed systems like cloud **computing.**
- **Mixed approach:** a mixture between the 2 approaches to require advantage of every approach.

2.2 Classification consistent with the System Topology

- **Dynamic approach:** This approach takes under consideration the present state of the system throughout load balancing decisions. This approach is a lot of appropriate for distributed systems like cloud computing.
- **Adaptive approach:** This approach adapts the load distribution to system standing changes, by ever-changing their parameters dynamically and evens their algorithms. This approach adapts the load distribution to system status changes, by changing their parameters dynamically and even their algorithms. This approach is a lot of appropriate for distributed system like Cloud Computing.

3. RELATED WORK

In complicated and huge systems, there's an tremendous need for load balancing. For simplifying load equalization globally (e.g. in an cloud), one factor which may be done is, using techniques would act at the elements of the clouds in such how that the load of the whole cloud is balanced. Following analysis papers that mentioned concerning load equalization algorithms and varied different techniques in cloud:

3.1 Tingting Wang et al. [2014]

This paper mentioned regarding the scheduling and load balancing. To solve the material, consider the new distinctiveness of cloud computing and unique adaptive genetic algorithmic program (AGA) a brand new scheduling algorithm supported dual-fitness adaptive algorithm-job straddling time and load balancing genetic algorithm (JLGA) is conventional. Then judge against the performance of JLGA with AGA through simulations [4].

3.2 Saeed javanmardi et al. [2014]

In this paper with the assistance of genetic formula and fuzzy theory, describe a hybrid job planning approach, that take under consideration the load feat of the coordination and reduces total completing time and completing value. The main goal of the analysis is to assign the roles to the resources with considering the VM unit of measuring and time-span of jobs. The results of the experiments show the potency of the planned approach in term of finishing time, execution value and average degree of inequity [5].

3.3 Tarun Goyal et al. [2013]

This paper represents that cloud computing may be a paradigm within which IT (information technology) application offer as a service. Cloud computing permits users to utilize the computation, storage, knowledge and services from round the world in commercialize manner. In cloud atmosphere, scheduling is that the major issue. Scheduling is responsible economical utilization of the resources. during this manuscript, a development model based on smallest amount network impediment using Suffrages Heuristic coupled Genetic algorithm for scheduling sets of freelance jobs algorithm is projected, the target is to reduce the make span[6].

3.4 Lucio Agostinho [2011]

This paper discussed about the cloud computing the distribution and arrangement of numerous virtual possessions, such as VM's are still a dispute. The optimization of these processes brings the improvement of recovering the energy savings and load balancing in bulky datacenters. Resource allotment and arrangement also impact in federated clouds anywhere assets can be leased from correlate domains. This paper projected a bio-inspired VM distribution method based on inherent Algorithms to optimize the VM transmission across federated cloud domains. The most important involvement of this exertion is an inter-domain allotment algorithm that takes into depiction the capability of the links concerning the domains in order to evade superiority of service deprivation for VMs allocated on collaborator domains. Architecture to replicate federated clouds is also involvement of this manuscript [7].

A lot of research has been done on the scheduling and load balancing in cloud. There are a many algorithms in cloud computing that used to balance the load between the nodes.

So, Load Balancing deserves more research. In this paper, an approach for Load Balancing in cloud using Enhanced Genetic Algorithm is presented.

4. LOAB BALANCING IN CLOUD USING GENETIC ALGORITHM

Genetic algorithm is predicated on biological thought of generation of the population, a speedy growing area of Artificial intelligence. GA's area unit impressed by Darwin's theory concerning Evolution. According to the Darwin "Survival of the fittest". It also a used as the method of scheduling in which the tasks are assigned resources according schedules in situation of scheduling, which tell about which resource is to be assigned to which task. Genetic algorithmic program is predicated on the biological idea of population generation [9].

Genetic algorithms are part of evolutionary computing, that could be a rapidly growing space of Artificial intelligence. Algorithm is started with a collection of solutions (represented by chromosomes) known as population. Solutions from one population are taken and used to produce new population. This is often motivated by a hope, that the new populations are higher than the previous one. Solutions that are selected to make new solutions (offspring) based on their fitness - the a lot of appropriate they're the lot of possibilities they need to reproduce. To use a genetic algorithm, you want to represent reply to your crisis as a genome (or chromosome). The inherent algorithm then creates a inhabitants of solutions and applies inherent operators like mutation and crossover to evolve the solutions so as to search out the simplest one(s). The rule repeatedly modifies a population of individual solutions. At every step, the genetic rule randomly selects individuals from this population and uses them as parents to produce the kids for future generation. Over serial generations, the population "evolves" toward a best resolution.

Some Terms Used in Genetic Algorithms

- **Initial Population**

  Initial population is that situate of all the individuals that are employed in the Genetic algorithm to search out the best resolution.

- **Fitness Function**

  It is the measure of the superiority of an individual in the population.

- **Selection**

  Selection mechanism is employed to select an intermediate resolution for consecutive generation supported the survival of the Darwin law.

- **Crossover**

  Crossover/hybridizing operation may be achieved by choosing 2 parents and so making a replacement individual tree by alternating and reforming the elements of these oldsters.

- **Mutation**

  Mutation takes place whenever the population tends to become consistent because of continual use of copy and crossover operator.

4.1 Working of the Proposed Technique

Start describes that system can being for running. Once the system start user will be going to login on the system or cloud atmosphere. If the user all ready register then he or she enter into cloud environment otherwise initial register then login. Check Authentication won't to certify user that enters is the member of this atmosphere or not. Then decide the algorithm and process the nodes to maintain the load. One's processing the node then evaluate the resources like CPU time, Memory usage, Flow time, Makespan time, Bandwidth etc. and at the end compare the algorithms like proposed algorithm (Enhanced Genetic Algorithm), Ant Colony Optimization (ACO) and then stop the whole process.

Flow Chart of Proposed Work

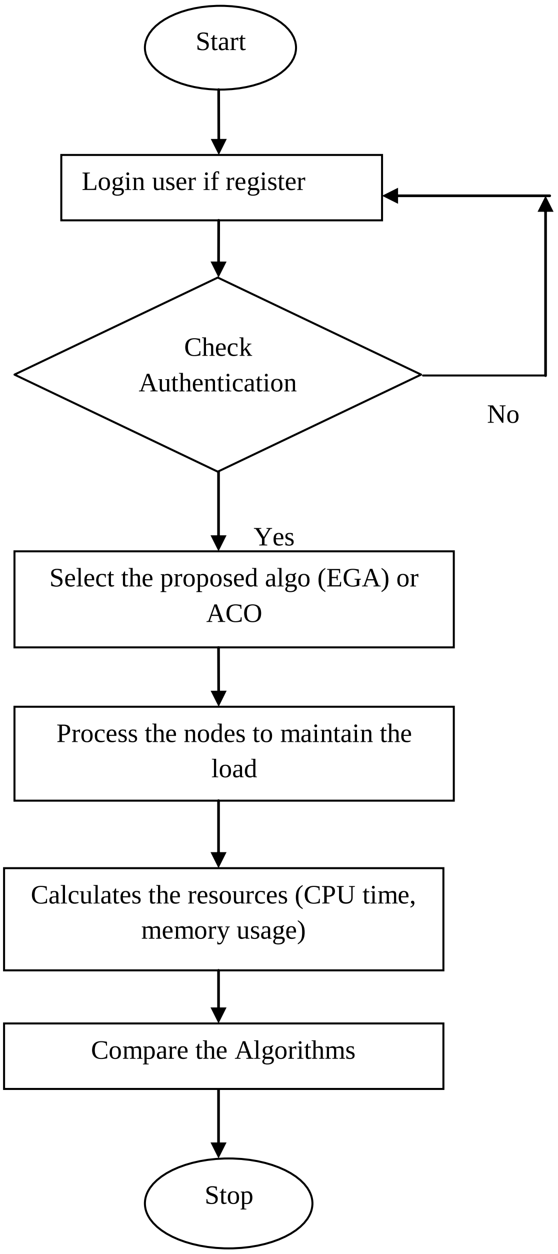

Start
Login user if register
Check Authentication
No
Yes
Select the proposed algo (EGA) or ACO
Process the nodes to maintain the load
Calculates the resources (CPU time, memory usage)
Compare the Algorithms
Stop

Fig 1: Flow chart

4.2 Proposed Algorithm for Load Balancing (Enhanced Genetic Algorithm)

```
1.  Login the user for authentication to begin  the load
    balancing system.

2.  Check the authorization

3.  choose the load balancing algorithm program to
    keep the load

4.  provide the amount of nodes to process the
    iterations

            1, N: Best answer
            Iterator, $\lambda \leftarrow$ zero; fitness$\leftarrow$ 0

5.  Initialize a population with every generated
    individual (chromosome).
6.  Calculate fitness for every and each individual.
      whereas iterator < MaxIter do

      $\lambda$ = random (0, 1)

       if $\lambda < \lambda 2$ then

             Fitness = Fitness2

             else

             Fitness = Fitness1

             end if

          for i = one to S do

          Fitness i $\leftarrow$ fitness (i)

    end for

7.  Choose 2 chromosomes, as parents that have  best
    fitness worth.

           N$\leftarrow$ fitness (i)

8.  Apply crossover between the parents with chance
    and crossover rate.

            if random(0, 1) < pc then

           temp1, N $\leftarrow$ crossover (p1, p2)

                 end if

9.  Apply mutation with possibility and mutation  rate.

           if random(0, 1) <  pm then

          P1,N $\leftarrow$ mutation(temp)

           end if

          iterator $\leftarrow$ iterator + one

          end while

10. Repeat Step 4- Step nine, till enough members are
    generated.

11. Repeat from step three, till stopping criteria to be
    met.
```

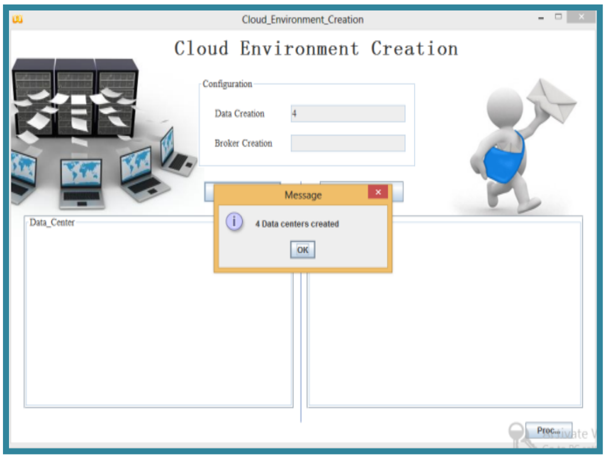
Where S: Population Scale, N: Worker node number, $\lambda 1$, $\lambda 2$: Probability of fitness 1 and fitness 2, Maxiter: Maximum iterations

5. IMPLEMENTATION

5.1 Evaluation

In order to assess the effectiveness of our proposal for execution on Clouds, we've got processed a true case study for finding a really well-known benchmark drawback projected within the literature, see as an example. Broadly, the experimental methodology concerned 2 steps. First, we execute the problem on single machine by variable a private problem parameter by employing a finite part package that allowed U.S. to collect real job information, i.e., process times and input/output file information sizes by suggests that of the generated job information, we tend to instantiated the Net beans simulation. Lastly, the obtained results relating to the performance of our proposal compared with some Cloud scheduling alternatives are reported.

5.1.1 Software Development of the Proposed Technique



Cloud_Environment_Creation

Cloud Environment Creation

Configuration

Data Creation: 4

Broker Creation:

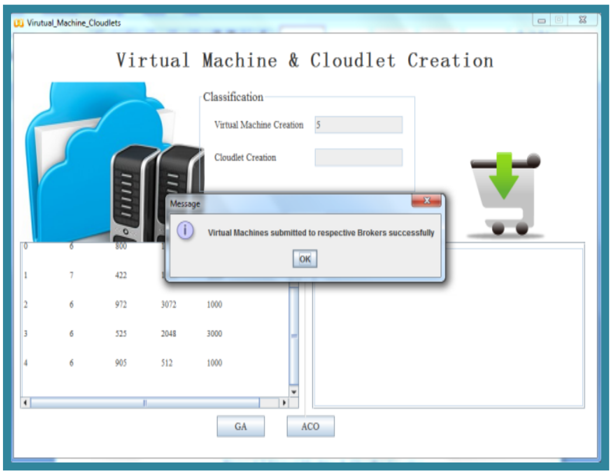
Data_Center

Message

4 Data centers created

Fig 2: Cloud Environment Creation

The fig 2 is used for the cloud environment creation. There are data creations and broker creations that are further used by virtual machine.



Virutual_Machine_Cloudlets

Virtual Machine & Cloudlet Creation

Classification

Virtual Machine Creation: 5

Cloudlet Creation:

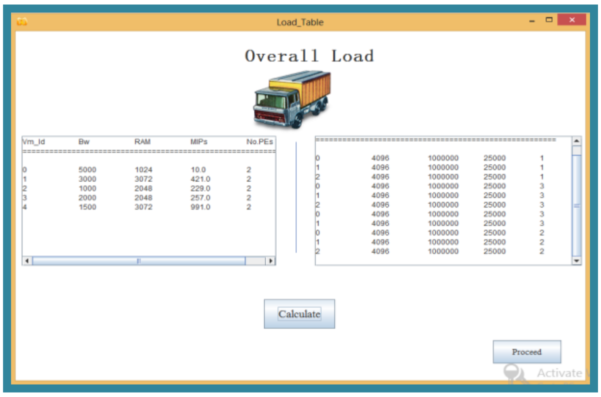
0	6	800	[illegible]	[illegible]
1	7	422	[illegible]	[illegible]
2	6	972	3072	1000
3	6	525	2048	3000
4	6	905	512	1000

Message

Virtual Machines submitted to respective Brokers successfully

Fig 3: Virtual Machine and Cloudlet Creation

This fig 3 is used for virtual machine and cloudlet creation according to the data creation. After this virtual machine submitted to respective brokers successfully.



Overall Load

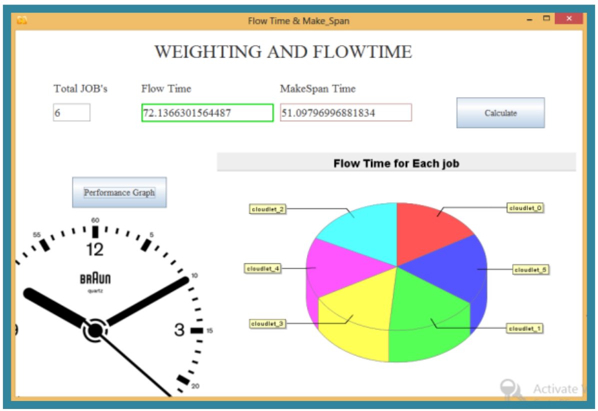
```
Vm_Id      Bw      RAM      MIPs      No.PEs
=====================================================

0          5000    1024     10.0      2
1          3000    3072     421.0     2
2          1000    2048     229.0     2
3          2000    2048     257.0     2
4          1500    3072     991.0     2
```

```
=====================================================

0      4096     1000000     25000     1
1      4096     1000000     25000     1
2      4096     1000000     25000     1
0      4096     1000000     25000     3
1      4096     1000000     25000     3
2      4096     1000000     25000     3
0      4096     1000000     25000     3
1      4096     1000000     25000     3
0      4096     1000000     25000     2
1      4096     1000000     25000     2
2      4096     1000000     25000     2
```

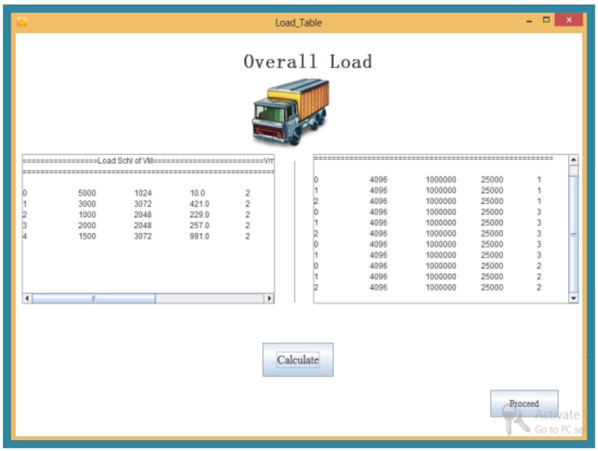
Fig 4: Overall load according to cloudlets for ACO



WEIGHTING AND FLOWTIME

Total JOB's: 6
Flow Time: 72.1366301564487
MakeSpan Time: 51.09796996881834

Fig 5: Weighting and Time Flow for Ant Colony Optimization (ACO)



Overall Load

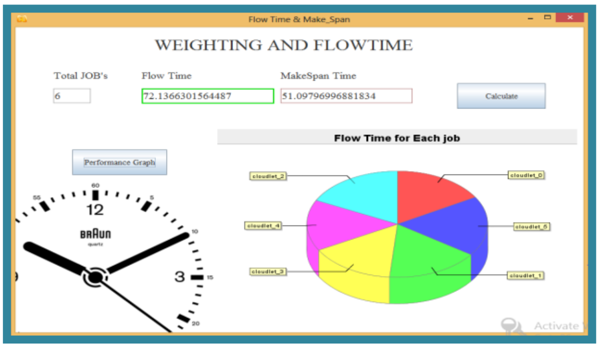
```
================Load Schl of VM========================Vm
=====================================================

0          5000    1024     10.0      2
1          3000    3072     421.0     2
2          1000    2048     229.0     2
3          2000    2048     257.0     2
4          1500    3072     991.0     2
```

```
=====================================================

0      4096     1000000     25000     1
1      4096     1000000     25000     1
2      4096     1000000     25000     1
0      4096     1000000     25000     3
1      4096     1000000     25000     3
2      4096     1000000     25000     3
0      4096     1000000     25000     3
1      4096     1000000     25000     3
0      4096     1000000     25000     2
1      4096     1000000     25000     2
2      4096     1000000     25000     2
```

Fig 6: Overall load for EGA



WEIGHTING AND FLOWTIME

Total JOB's: 6
Flow Time: 72.1366301564487
MakeSpan Time: 51.09796996881834

Fig 7: Weighting and Time Flow for Enhanced Genetic Algorithm (EGA)

The fig 5 and fig 7 is used to show the final weighting and time flow after the processing of cloudlet according to host id and virtual machine id. There is performance graph that shows the time spam.

From the above graph of a EGA shows that the flow time and make span time is less as compare to ACO so EGA is more efficient from ACO. It reduces the VM migrations and handles the load in better way.

Table 1. Comparison between ACO and EGA Algorithms

Data Creation	Broker Creation	Virtual Machine(VM) Creation	Cloudlet	ACO Flow and Makespan Time	EGA Flow and Makespan Time
2	3	4	5	FT -201.8 MT -15.34	FT -4.3 MT -3.8
5	6	7	8	FT –1079.7 MT -54.2	FT –19.3 MT -7.77
8	9	10	11	FT – 18955.3 MT -153.0	FT –1267.4 MT -42.8
11	12	13	14	FT –1564.5 MT -54.7	FT –10.78 MT -3.4

The table 1 displays the results of ACO flow time and EGA flow time with different data creation, broker creation, virtual machine and cloudlets. In this table the EGA flow time and makespan time is less than the ACO flow time and makespan time that is our research work to proof that EGA is best than ACO.

6. CONCLUSION

The anticipated GA schedules VMs such that it achieves load complementary and there is fewer need of VM migrations because it allocate VMs to physical machines in elegant way using fitness function. GA evaluates the load of the knot once VM is deployed on knot before essentially deploy on that and find a clarification which gives the superlative load balancing. The development algorithms are as NP complete downside and another is the execution of algorithm with the virtual machine's actual migration will save 30 – 40 % of the total physical machine's. During this work ACO and GA algorithm is implemented to keep up the load and solved the NP problem and then association is done with scheduling techniques i.e. ACO (Ant colony optimization). The EGA is best calculated during this work.

Future Scope

- To contain VM consolidation to save power and also save electricity prices.
- To include thermal components similarly since cooling prices for information centers also are large to save lots of electricity prices.
- Thus, VM management software system will be developed to incorporate of these necessities that conflicting in nature with each other which might be set depending upon current demand of the cloud supplier.

7. REFERENCES

[1] Kiranveer kaur and Amritpal kaur, “Survey of Load Balancing Algorithm in Cloud”, International Journal of Engineering Research and Technology (IJERT), ISSN: 2278-0181, Vol. 4, Issue 3, March 2015.

[2] Amandeep kaur sidhu and Supriya kinger, “Analysis of Load Balancing Techniques in Cloud Computing”, International Journal of Computer and Technology, ISSN: 2277-3061, Vol. 4, No. 2, March-April 2013.

[3] Shilpa V Pius and Shilpa T S, “Survey on Load Balancing in Cloud Computing”, International Conference on Computing, Communication and Energy Systems (ICCCES), 2014.

[4] Tingting Wang, Zhaobin Liu, Yi Chen, Yujie Xu, “Load Balancing Task Scheduling based on Genetic Algoritm in Cloud Computing”, IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, 2014.

[5] Saeed Javanmardi, MohammadShojafar, Danilo Amendola, Nicola Cordeschi, “Hybrid Job Scheduling Algorithm for Cloud Computing Environment”, adfa, 2014.

[6] Tarun Goyal, Aakankaha Agrawal, “Host Scheduling Algorithm using Genetic Algorithm in Cloud Computing Environment”, International Journal of Research in Engineering and Technology (IJERT), Vol. 1, June 2013.

[7] Lucio Agostinho, Guilherme Feliciano, Leonardo Olivi, Eleri Cardozo, “A Bio-Inspired Approach to Provisioning of Virtual Resources in Federated Clouds”, IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, 2011,pp.548-604.

[8] Andrew J. Younge, Gregor von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, Warren Carithers, “Efficient Resource Management for Cloud Computing Environments”, IEEE, 2010.

[9] Rajkumar Buyya, Rajiv Ranjan, Rodrigo N.Calheriros, “Modeling and Simulation of Scalable Cloud Computing Environment and the CloudSim Toolkit: Challanges and Opportunities”, International Conference on High Performance Computing and Simulation, HPCS 2009.