Academic Science

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 6, Issue 1
January 2017

Load Balancing in Cloud Using Enhanced Genetic Algorithm

Harshdeep Sharma
Dept. Computer Engineering
Yadwindra College of Engineering
Punjabi University, Patiala

Gianetan Singh Sekhon
Assistant Professor
Dept. Computer Engineering
Yadwindra College of Engineering
Punjabi University, Patiala

Abstract: The cloud computing is the development of distributed computing, parallel computing and grid computing, or defined as the commercial implementation of these computer science concepts. One of the fundamental issues in this environment is related to task scheduling and Load Balancing. Cloud task scheduling is an NP-hard optimization problem, and many meta-heuristic algorithms have been proposed to solve it. A good task scheduler should adapt its scheduling strategy to the changing environment and the types of tasks. This paper proposes a cloud task scheduling policy based on Load Balancing Enhanced Genetic (EGA) algorithm. The main contribution of our work is to maintain the balance in the entire system load while trying to minimizing the Make span of a given tasks set. The new scheduling strategy was simulated using the Net Beans toolkit package. Experiments results showed the proposed Enhanced Genetic (EGA) algorithm outperformed FCFS (First Come First Serve) and Compare the EGA, ACO.

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

I. 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.

There's no commonplace definition of Cloud computing. Usually it consists of a bunch of distributed servers called masters, providing demand services and resources to completely different clients called clients in a network with quantifiability and reliableness of knowledge centre. The distributed computers give on-demand services. Services could also be of package resources (e.g. package as a Service) or physical resources or hardware/infrastructure (e.g. Hardware as a Service or Infrastructure as a Service). Amazon EC2 (Amazon Elastic compute Cloud) is associate example of cloud computing services. Within the analysis work completely different algorithms are accustomed to maintain the load [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.

II. LOAD BALANCING

Cloud computing is economical and scalable however maintaining the stability of processing such that a large amount of jobs within the cloud computing environment could be lead to a terrible consequences 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 single node is weak, thus raising the general performance and effectiveness of the system.

Load balancing may be a comparatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time. Correct load balancing will facilitate in utilizing the accessible resources optimally, thereby minimizing the resource consumption. Dividing the traffic between servers, information is transferred to both ends sent and received without major delay. Different types of algorithms are available that helps traffic load between accessible servers. A basic example of load balancing in our lifestyle is associated with websites. Without load balancing, users might expertise delays, timeouts and doable long system responses. 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 how the charge is distributed and how processes are allotted to nodes (the system load).
- Depending on the data status of the nodes (System Topology).

In the first case in designed as designed as centralized approach, distributed approach or hybrid approach within the second case as static approach, dynamic or reconciling approach.

a) Classification consistent with the System Load

• **Centralized approach:** during this approach, one node is responsible for managing the distribution at intervals of the whole system.

• **Distributed approach:** during this approach, every node severally builds its own load vector by collection the load information of alternative nodes. Decisions made locally using local load vectors. This approach is a lot of advantageous for widely distributed systems like cloud computing.

• **Mixed approach:** a mixture between of the 2 approaches to required advantage of every approach.

b) 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 advantageous 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.

III. RELATED WORK

In complicated and huge systems, there's a tremendous need for load balancing. For simplifying load equalization globally (e.g. in a 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:

• **Tingting Wang et al. [2014]** this paper mentioned regarding the scheduling and load balancing. To solve the matter, considering the new characteristics of cloud computing and original adaptive genetic algorithmic program (AGA) a brand new scheduling algorithm supported double-fitness adaptive algorithm-job spanning time and load balancing genetic algorithm (JLGA) is established. Then compare the performance of JLGA with AGA through simulations [4].

• **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 system and reduces total execution time and execution 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].

• **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 paper, a scheduling model based on minimum network delay using Suffrages Heuristic coupled Genetic algorithm for scheduling sets of freelance jobs algorithm is projected, the target is to reduce the make span[6].

• **Lucio Agostinho [2011]** this paper discussed about the cloud computing the allocation and scheduling of multiple virtual resources, such as virtual machines (VMs), are still a challenge. The optimization of these processes brings the advantage of improving the energy savings and load balancing in large data enters. Resource allocation and scheduling also impact in federated clouds where resources can be leased from associate domains. This paper proposed a bio-inspired VM allocation method based on Genetic Algorithms to optimize the VM distribution across federated cloud domains. The main involvement of this work is an inter-domain allocation algorithm that takes into account the capacity of the links connecting the domains in order to avoid quality of service degradation for VMs allocated on partner domains. Architecture to replicate federated clouds is also a contribution of this paper [7].

- **Andrew J.Younge et al. [2010]** this paper represents the notion of Cloud computing has not only reshaped the field of distributed systems however conjointly basically modified however businesses utilize computing these days. Whereas cloud computing provides several advanced options, it still has some disadvantage like the comparatively high operating cost for each public and personal clouds. The world of Green computing is also becoming increasingly important in a world necessary with restricted energy resources and an ever-rising demand for more computational power. During this paper a brand new framework is accessible that has economical Green enhancements inside a scalable cloud computing design. Using power sensitive scheduling techniques, variable resource management, live immigration, and a minimal virtual machine design, overall system power are immensely improved in a very data centre based mostly cloudy with minimal performance overhead [8].

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.

IV. 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 is also a used as the method of scheduling in which the tasks are assigned resources in accordance to the 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 recent population is 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 problem as a genome (or chromosome). The genetic algorithm then creates a population of solutions and applies genetic 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 the set 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 produce 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 their ancestors.

➢ Mutation

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

A. Working of The Proposed Technique

Start describes that system is in run able state and ready for running. Once the system starts user will be going to login on the system or cloud atmosphere. If the user had already registered then he or she enters into cloud environment otherwise he has to register before login. Check Authentication is required 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 evaluates the resources like CPU time, Memory usage, Flow time, Make span time, Bandwidth etc. and at the end compare the algorithms like proposed algorithm (Enhanced Genetic Algorithm), Ant Colony Optimization (ACO) and then the process gets terminated.

B. 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, λ ← zero; fitness← 0
5.  Initialize a population with every generated individual (chromosome).
6.  Calculate fitness for every and each individual.
    Whereas iterator < MaxIter do
     λ = random (0, 1)
      If λ < λ2 then
          Fitness = Fitness2
          Else
          Fitness = Fitness1
          End if
             For i = one to S do
             Fitness i ← fitness (i)
          End for
7.  Choose 2 chromosomes, as parents that have best fitness worth.
            N← fitness (i)
8.  Apply crossover between the parents with chance and crossover rate.
        If random (0, 1) < pc then
            temp1, N ← crossover (p1, p2)
            End if
9.  Apply mutation with possibility and mutation rate.
        If random (0, 1) < pm then
       P1, N ← mutation (temp)
        End if
       Iterator ← 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, λ1, λ2: Probability of fitness 1 and fitness 2, Maxiter: Maximum iterations

Algorithm for ACO:

```
Initialize Variables;
Initialize Pheromone on the trail selected by GJAP;
While (Value of Timer < Tl) do
Ants Construct Solutions;
Xnew = min {for j (Pk) |k=1, 2, K};
If Xnew < X then X=Xnew;
Pheromone Update;
End
End
```

Algorithm for PSO:

1) Initialize the swarm by assigning a random position in the problem space to each particle.

2) Evaluate the fitness function for each particle.

3) For each individual particle, compare the particle‘s fitness value with its pbest. If the current value is better than the pbest value, then set this value as the pbest and the current particle‘s position, xi, as pi.

4) Identify the particle that has the best fitness value. The value of its fitness function is identified as guest and its position as pg.

5) update the velocities and positions of all the particles using (1) and (2).

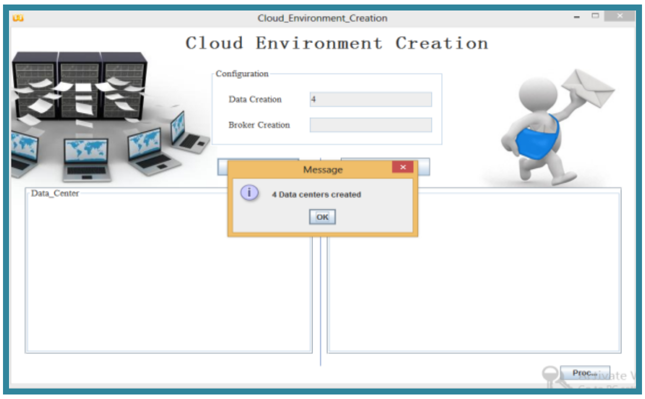
6) Repeat steps 2–5 until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

V. IMPLEMENTATION

A. Evaluation

In order to assess the effectiveness of our proposal for execution on Clouds, we've generated 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 suggesting 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.

- **Software Development of the Proposed Technique**

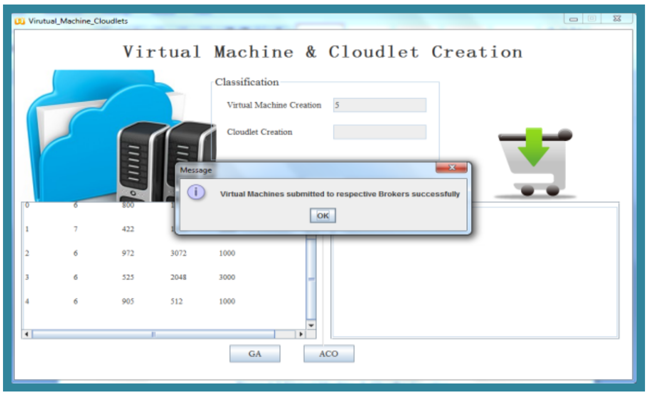


Data Creation: 4
Broker Creation:

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
Message
OK
GA
ACO

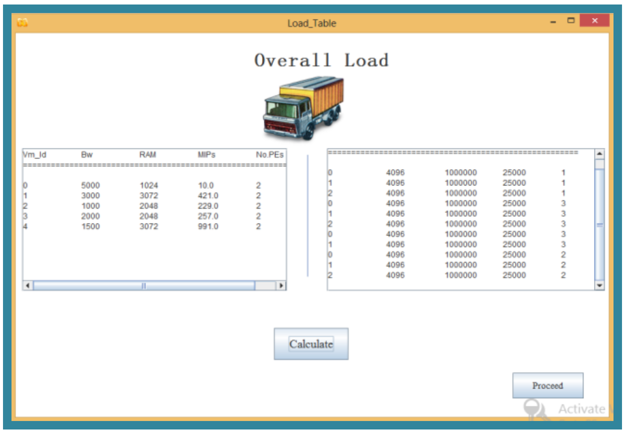
Virtual Machine Creation: 5
Cloudlet Creation:

Virtual Machines submitted to respective Brokers successfully

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

Fig 3 Virtual Machine and Cloudlet Creation

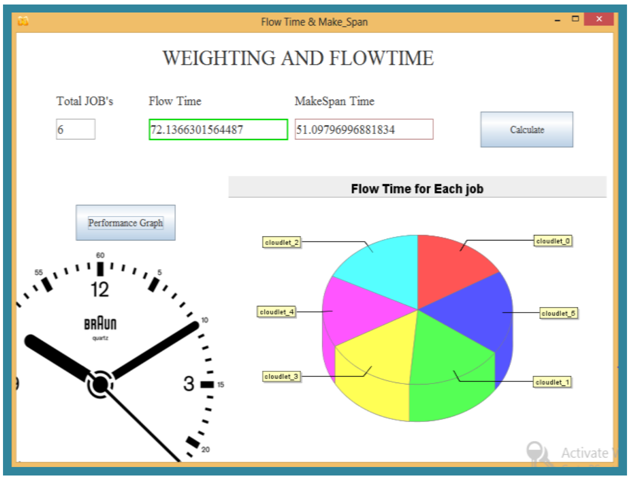
This fig 3 is used for virtual machine and cloudlet creation according to the data creation. After this virtual machine submitted to respective brokers successfully.



```
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
```

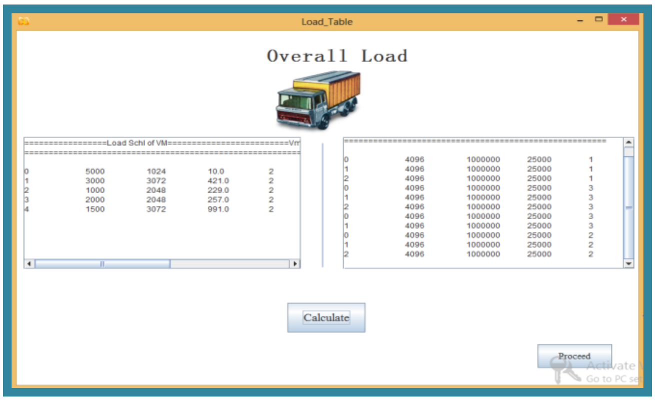
```
==========================================================
0    4096    1000000    25000    1
1    4096    1000000    25000    1
2    4096    1000000    25000    1
0    4096    1000000    25000    3
1    4096    1000000    25000    3
2    4096    1000000    25000    3
0    4096    1000000    25000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



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

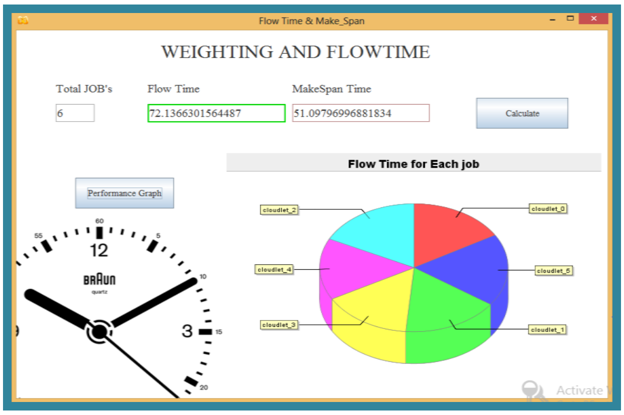
Fig 5 Weighting and Time Flow for Ant Colony Optimization (ACO)



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

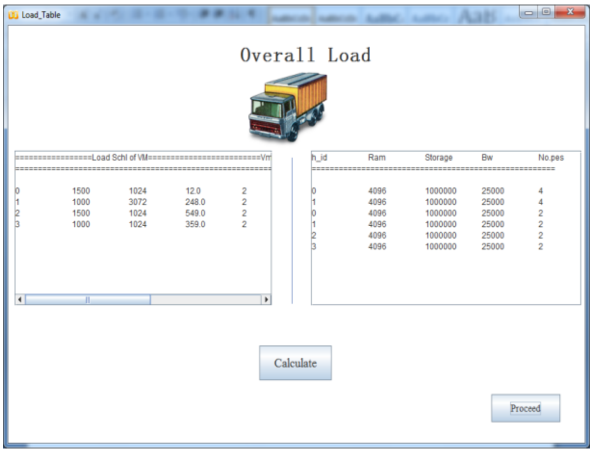
Fig 6 Overall load for EGA


Flow Time & Make_Span

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)


Load_Table

Overall Load

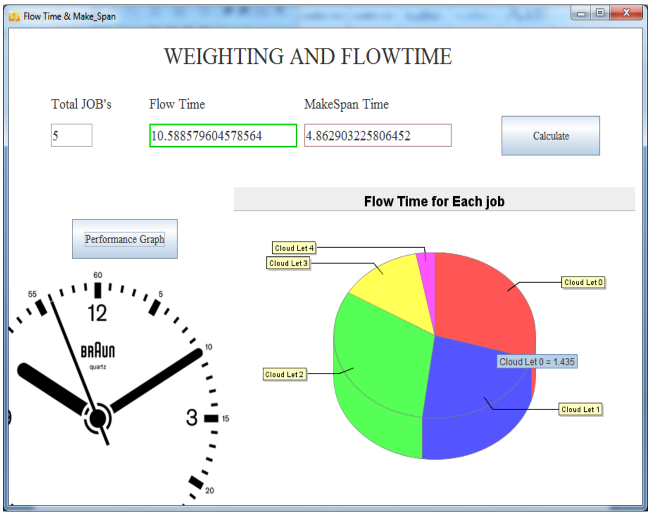
```
================Load Schl of VM=======================Vm
==========================================================

0        1500      1024      12.0      2
1        1000      3072      248.0     2
2        1500      1024      549.0     2
3        1000      1024      359.0     2
```

```
h_id     Ram       Storage    Bw        No.pes
=====================================================

0        4096      1000000    25000     4
1        4096      1000000    25000     4
0        4096      1000000    25000     2
1        4096      1000000    25000     2
2        4096      1000000    25000     2
3        4096      1000000    25000     2
```

Fig 8 Overall load according to cloudlets for PSO


Flow Time & Make_Span

WEIGHTING AND FLOWTIME

Total JOB's: 5
Flow Time: 10.588579604578564
MakeSpan Time: 4.862903225806452

Fig 9 Weighting and Time Flow for Particle Swarm Optimization (PSO)

The fig 5, fig 7,fig 8 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.

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

Table 1 Comparison between ACO, EGA and PSO Algorithms

Data Creation	Broker Creation	Virtual Machine(VM) Creation	Cloudlets	ACO Flow and Make span Time	EGA Flow and Make span Time	PSO Flow and Make span Time
2	3	4	5	FT – 201.8 MT – 15.34	FT- 4.3 MT-3.8	FT- 10.58 MT- 4.86
5	6	7	8	FT – 1079.7 MT – 54.2	FT – 19.3 MT – 7.77	FT - 4.79 MT-2.67
8	9	10	11	FT – 18955.3 MT – 153.0	FT – 1267.4 MT – 42.8	FT – 8.06 MT – 3.48
11	12	13	14	FT – 1564.5 MT- 54.7	FT – 10.78 MT – 3.4	FT – 7.52 MT – 3.13

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

VI CONCLUSION

The proposed GA schedules VMs such that it achieves load balancing and there is less need of VM migrations because it allocates VMs to physical machines in smart way using fitness function. GA evaluates the load of the node once VM is deployed on node before actually deploying on that and finds a solution which gives the best 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, PSO and GA algorithm is implemented to keep up the load solved the NP problem and then association is done with scheduling techniques i.e. ACO (Ant colony optimization). The GA 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 because of large saves a lot of electricity price.
- Thus, VM management software system will be developed to incorporate these necessities that all conflicting in nature with each other which might be set depending upon current demand of the cloud supplier.

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,pp.2.

[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, pp. 1-11.