Survey of Load Balancing Algorithms in Clouds

Kiranveer Kaur [1],
[1]M.Tech Research Fellow,
Department of Computer Science Engineering,
Sri Guru Granth Sahib World University,
Fatehgarh Sahib, Punjab

Amritpal Kaur[2]
[2]Assistant Professor,
Department of Computer Science Engineering,
Sri Guru Granth Sahib World University,
Fatehgarh Sahib, Punjab

Abstract: **Cloud computing is the way of computing, via the internet that shares computer resources instead of using software or storage on a local PC. It stores the data and resources in the open environment. So now a day's amount of data storage increase quickly. Load Balancing is the main issues in Cloud which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. Load Balancing provides proper utilization of resources and enhancing the performance of the system. The existing algorithms that can provide load balancing and also provide better strategies through efficient job scheduling and resource scheduling techniques. In order to gain maximize the profit and balancing algorithms, it is necessary to utilize resources efficiently. This paper discusses some of the existing load balancing algorithms in cloud computing.**

I. INTRODUCTION

Cloud computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients requirement at exact time. Cloud is a metaphor term used for Internet. The whole Internet can be view as a cloud. A user is need to pay only for the usage resources per time.

There is no standard definition of Cloud computing. Generally it consists of a group of distributed servers known as masters, providing demand services and resources to different clients known as clients in a network with scalability and reliability of data center. The distributed computers provide on-demand services. Services may be of software resources (e.g. Software 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 an example of cloud computing services. In the research work different algorithms are used to maintain the load that is discussed in this work.

➢ *Cloud Components*

A Cloud system consists of 3 major components such as clients, data centre and distributed servers. Each element has
a definite purpose and plays a specific role.

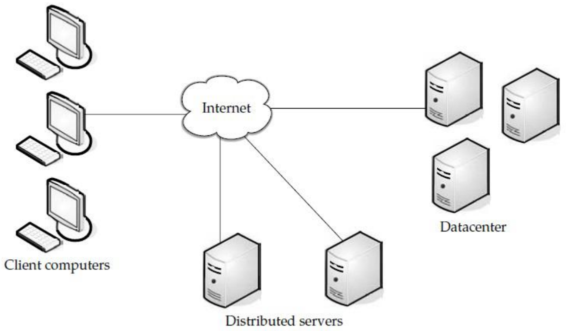

Figure 1: Three components make up a cloud computing solution

(a) Clients

End users interrelate with the clients to manage information related to the cloud. Clients generally drop into three categories as given in [1]:

- Mobile: Windows Smartphone, smart phones like a Blackberry or I phone.
- Thin: They don't do any computation work. They only present the information. Servers do all the mechanism for them. Thin clients don't have any internal memory.
- Thick: These apply different browsers like IE or Mozilla Firefox or Google Chrome to connect to the Internet cloud.

Now-a-days thin clients are more popular as compare to other clients because of their low price, security, low consumption of power, less noise, easily replaceable and repairable etc.

(b) Data Center

Data center is nothing but a collection of servers hosting different applications. End users connect to the data center to subscribe different applications. A data center might exist at a large distance from the clients.

(c)Distributed Servers

Distributed servers are the parts of a cloud which are present throughout the Internet hosting different applications. But using the application from the cloud, the user will be aware of that he is using this application from its own machine.

➢ **Type of Clouds**

Based on the sphere of influence or environment in which clouds are used, clouds can be divided into 3 types:
_ Public Clouds
_ Private Clouds
_ Hybrid Clouds

_Community Clouds

➢ *Services provided by Cloud computing*

Service means different types of applications provided by different servers across the cloud. It is generally given as “as a service”. Services in a cloud are of 4 types as given in [1] :

_ Anything as a Service (XaaS)
_ Software as a Service (SaaS)
_ Platform as a Service (PaaS)
_ Hardware as a Service (HaaS) or Infrastructure as a Service (IaaS)

- *Anything as a Service(XaaS)*

It is a collective term said to stand for a number of things including "X as a service", and also called everything as a service. The short form refers to an increasing number of services that are delivered over the Internet rather than provided locally or on-site. The most common examples are Software as a Service, Infrastructure as a service and Platform as a service. The combine use of these three is sometimes referred to as the SPI model.

- *Software as a Service (SaaS)*

In software as a service, the user use different software applications from different servers through the Internet. The user uses the software as it is without any change and do not need to make lots of changes or don’t require integration to other systems.

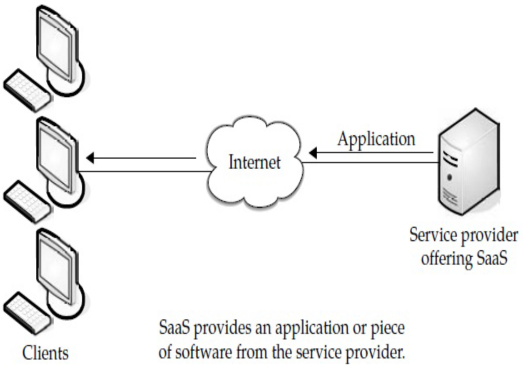
Internet
Application
Service provider offering SaaS
Clients
SaaS provides an application or piece of software from the service provider.

Figure 2: Software as a service

The client will have to pay for the time he use the software. The software that does a clear-cut task without any need to interact with other systems makes it an ideal candidate for Software as a Service.

Some of these applications include:

_ Customer resource management (CRM)
_ Video conferencing
_ IT service management
_ Accounting
_ Web analytics
_ Web content management

- *Platform as a Service (PaaS)*

Platform as a service provides all the resources that are required for building applications and services completely from the Internet, without download or install software. It’s services are software plan, development, testing, consumption, and hosting. Other services can be team cooperation, database integration, web service integration, data security, storage space and versioning etc.

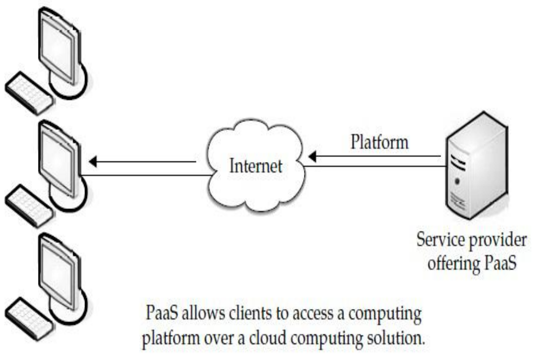
Internet
Platform
Service provider offering PaaS
PaaS allows clients to access a computing platform over a cloud computing solution.

Figure 3: Platform as a service

- *Hardware as a Service (HaaS)*

It is also known as Infrastructure as a Service. It offers the hardware as a service to an organisation so that it can put anything into the hardware according to its will [1].

It allows the user to “rent” resources (taken from [1]) as

_ Server space
_ Network equipment
_ Memory
_ CPU cycles
_ Storage space

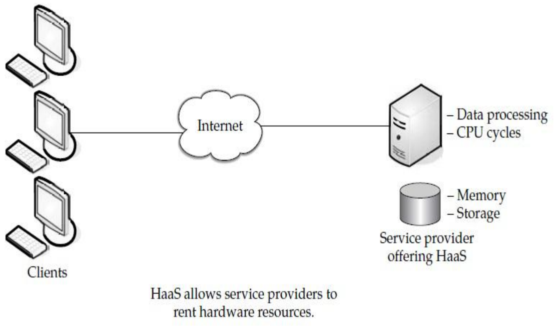
Internet
– Data processing
– CPU cycles
– Memory
– Storage
Service provider offering HaaS
Clients
HaaS allows service providers to rent hardware resources.

Figure 4: Hardware as a service

Cloud computing provides a Service Oriented Architecture (SOA) and Internet of Services (IoS) type applications, excluding fault tolerance, high scalability, accessibility flexibility, reduced information technology overhead for the user concentrated cost of ownership, on demand services etc. Middle to these issues lies the establishment of an efficient load balancing algorithm.

LOAD BALANCING IN CLOUD

Load Balancing is a computer networking method to distribute workload across multiple computers or a computer cluster, net links, central processing units, disk drives and other resources, to get optimal resource utilization, maximize throughput, reduce response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase consistency through laying-off. The load balancing service is generally provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server. Load balancing is one of the central issues in cloud computing [2].

It is a mechanism that distributes the dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to complete a high user fulfilment and resource utilization ratio, hence recovering the overall performance and resource utility of the system. It also ensure that every computing resource is distributed efficiently and fairly [3]. It further prevents bottlenecks of the system which may occur due to load inequality. When one or more components of any service fail, load balancing helps in maintenance of the service by implement fair-over, i.e. in provisioning and de-provisioning of instances of applications without fail. The purpose of load balancing is improving the performance by balancing the load among these various resources (network links, central processing units, disk drives) to achieve optimal resource consumption, maximum throughput, maximum response time, and avoid overload. To distribute load on different systems, different types of load balancing algorithms are used.

In general, load balancing algorithms follow two major classifications:

- Depending on how the charge is distributed and how processes are allocated to nodes (the system load).
- Depending on the information position of the nodes (System Topology).

In the first case it designed as a centralized approach, distributed approach or hybrid approach and in the second case as static approach, dynamic or adaptive approach.

a) Classification According to the System Load

- Centralized approach: In this only one node is responsible for managing the distribution within the whole system.
- Distributed approach: In this each node independently builds its own load vector by collecting the load information of other nodes. Decisions are made using local load vectors. This approach is more suitable for generally distributed systems such as cloud computing.
- Mixed approach: A combination of above two approaches to take advantage of each approach.

b) Classification According to the System Topology

- Static approach: This approach is generally defined in the design or implementation of the system.
- Dynamic approach: This approach takes into account the current state of the system during load balancing decisions. This approach is more suitable for widely distributed systems such as cloud computing.
- Adaptive approach: This approach adapts the load distribution to system status changes, by changing their parameters dynamically and even their algorithms.

  There are different types of load balancing algorithms that discussed in this paper.

II. LITERATURE SURVEY

Saeed javanmardi et al. [2014] In this paper with the aid of genetic algorithm and fuzzy theory, present a hybrid job scheduling approach, which consider the load balancing of the system and reduces total execution time and execution cost. The main goal of this research is to assign the jobs to the resources with considering the VM MIPS and time-span of jobs. The new algorithm assigns the jobs to the resources with considering the job length and resources capacities. Evaluate the performance of the approach with some famous cloud scheduling models. The result of the experiments shows the efficiency of the proposed approach in term of execution time, execution cost and average degree of imbalance [4].

Hitesh A. Ravani et al. [2013] this paper discusses that Resource Scheduling is the process of mapping tasks to available resources on the basis of tasks characteristics and requirements. The received tasks are group on the basis of data and resources. Resource selection is done on the basis of its cost and turnaround times both using greedy approach and task selection on the basis of a priority. This way of resource selection and task selection gives better results over sequential scheduling. The available resources should be utilized efficiently without affection the service parameters of cloud. Main aim of this paper is to analyze the various scheduling algorithm and manage the resources which are precisely available at certain fixed times and for fifed intervals of time. Find the optimizes scheduling algorithm for resource so the cloud provider get benefits in term of efficient resource management which provide more resources to allocate without postponing or declining any user requests. Cloud users also get benefits in term of their monetary gains at each front [5].

Florin Pop et al. [2013] in this paper evolutionary computing offers different methods to solve NP-hard problems, finding a near-optimal solution. Task scheduling is a composite problem for large environments like Clouds. Genetic algorithms are a superior method to find a solution for this problem considering multi-criteria constrains. This is also a method used for optimization. In these types of environments service provider want to increase the profit and the customers (end-users) want to minimize the costs. So, it's all about money and minimum two optimization constrains. On the other hand, a good performance to ensure the QoS is to use the reputation of resources offered. This aspect is very important for service providers because represents a ranking method for them. In this paper a reputation guided genetic scheduling algorithm for independent tasks in inter-Clouds environments. The characters is considered in the selection phase of genetic algorithm as evolutionary criteria for the algorithm and evaluate the proposed solution considering load-balancing as a way to measure the optimization impact for providers and maxspan as a metric for user performance [6].

Jianfeng Zhao et al. [2011] it's a basic requirement in cloud computing that scheduling virtual resources to physical resources with balance load. The simple scheduling methods can not meet this requirement. This paper proposed a virtual resources scheduling model and solved it by advanced Non-dominated Sorting Genetic Algorithm II (NSGA II). This model was evaluated by balance load, virtual resources and physical resources were abstracted a lot of nodes with attributes based on analyzing the flow of virtual resources scheduling. NSGA II was engaged to address this model and a new tree sorting algorithms was adopted to improve the efficiency of NSGA II. In experiment, verified the correctness of this model. Comparing with Random algorithm, Static algorithm and Rank algorithm by a lot of experiments, at least 1.06 and at most 40.25 speed-up of balance degree can be obtained by NSGA II [7].

Lucio Agostinho [2011] In 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 datacenters. Resource allocation and scheduling also impact in federated clouds where resources can be leased from partner domains. This paper proposes a bio-inspired VM allocation method based on Genetic Algorithms to optimize the VM distribution across federated cloud domains. The main contribution 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 [8].

Andrew J.Younge et al. [2010] This paper represent the notion of Cloud computing has not only reshaped the field of distributed systems but also fundamentally changed how businesses utilize computing today. While cloud computing provides many advanced features, it still has some shortcoming such as the relatively high operating cost for both public and private clouds. The area of Green computing is also becoming increasingly important in a world with limited energy resources and an ever-rising demand for more computational power. In this paper a new framework is accessible that provides efficient green enhancements within a scalable cloud computing architecture. Using power sensitive scheduling techniques, variable resource management, live migration, and a minimal virtual machine design, overall system efficiency will be vastly improved in a data center based cloud with minimal performance overhead [9].

III. LOAD BALANCING ALGORITHMS CLASSIFICATION

Load balancing algorithms are divided into two categories:

i) **Traditional algorithms:** Like First come first serve, Round robin, Priority based scheduling etc.

ii) **Bio inspired algorithms:** These algorithms are further divide into two parts:
   a) Swam based: Honey Bee, Ant colony optimization, Bat etc
   b) Evolutionary: Genetic, Honey Bee Mating, PSO etc

i) Traditional Algorithms:

First come first serve (FCFS): First come first serve basis means that task that come first will be execute first.

Algorithm for FCFS:

1. FCFS VM load balancer maintains an index table of virtual machines & number of requests currently allocated to the VM. At start all have zero allocation.
2. a) The VM load balancer allocates the cloudlets/user requests to the available VMs on the basis of requests sent by the data center controller.
b) The data center controller stores the user requests in a queue on the basis of their arrival time.
c) The first request according to the arrival time is allocated to the VM which is under loaded or free by FCFS VM load Balancer.
3. The FCFS VM load balancer will execute the cloudlets and calculate the turnaround time, avg. waiting time and response time. After that it will display the result.
4. The data center controller receives the response to the request sent and then allocates the waiting requests from the job pool/queue to the available VM & so on.
5. Continue from step-2.

Round robin algorithm (RRA): In this Scheduling algorithm time is to be given to resources in a time slice manner. Processes are kept in a circular queue as shown in Figure. The CPU scheduler goes around this queue and allocates the CPU to each process for a time period of time slices without priority. It provides a complete fairness among the processes. Round-robin algorithm has been widely used in many scheduling approaches [10].

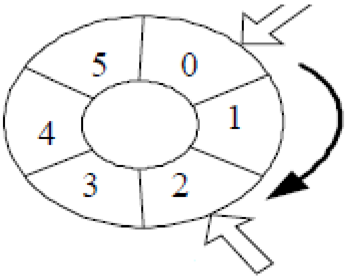

Figure 5: Round robin scheduling

Algorithm for RRA:
1. Round Robin VM load balancer maintains an index of VMs and state of the VMs (busy/available). At start all VM's have zero allocation.
2. a) The data center controller receives the user requests or cloudlets.
b) It stores the arrival time & burst time of the user requests.
c) The requests are allocated to VMs on the basis of their states known from the VM queue.
d) The round robin VM load balancer will allocate the time quantum for user request execution.

3. a) The round robin VM load balancer will calculate the turn- around time of each process.
b) It also calculates the response time and average waiting time of user requests.
c) It decides the scheduling order.
4. After the execution of cloudlets, the VMs are de-allocated by the round robin VM load balancer.
5. The data center controller checks for new pending or waiting requests in queue.
6. Continue from step-2.

Priority based scheduling (PBS): In this scheduling each job should be given a priority and according to the priority jobs should be execute.

ii) Bio inspired algorithms:
a) Swam based:
Honey Bee: A colony of honey bees can extend itself over Long distances (more than 10 km) and in multiple directions simultaneously to exploit a large number of food sources. A colony prospers by organize its foragers to good fields. In principle, flower patch with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar should receive fewer bees.

The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another. During the harvesting season, a colony continues its examination, keeping a percentage of the population as scout bees.

While harvesting from the patch, the bee monitor its food level. This is necessary to decide upon the waggle dance when they return to the hive. If the patch is still good as a food source, then it will be advertise in the waggle dance and more bees will be recruited to that source [11].

Basic Bee Algorithm:

```
1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) //Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.
```

Ant Colony Optimization (ACO): The basic idea for Ant colony optimization is to simulate the foraging behaviour of ant colonies. When a group of ants tries to search a food, they use a pheromone (chemical) to communicate with each other [12].

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
```

b) Evolutionary:
Genetic Algorithm: The Genetic Algorithm is based on natural selection and genetic recombination. The algorithm works by choosing solutions from the current population and then applying genetic operators – such as mutation and intersect to create a new population. The algorithm efficiently exploits historical information to speculate on new search areas with improved performance.

Algorithm for Genetic algorithm:

```
Begin Main
[Initialize] Produce random population of n chromosome.
[Fitness] In the given population, calculate the fitness value f(x) of every chromosome.
[Selection] According to the fitness value, select two parent individuals from the population.
[Crossover] Generation of the new offspring by performing the parents by using the crossover probability.
[Mutation] Mutate the new child at some position with the probability of mutation.
[Accepting] Now the new offspring the part of the next generation of population.
[Replace] Use the new generation as the current generation.
End Main
```

Particle Swam Optimization (PSO): Particle swarm optimization (PSO) is a computational intelligence oriented, population-based global optimization technique proposed by Kennedy and Eberhart in 1995[13]. It is stimulated by the social behavior of bird flocking searching for food. PSO has been widely applied to many engineering optimization areas due to its unique searching method, effortless concept, computational ability, and easy implementation. In PSO, the term ―particles refers to population members which are mass-less and volume-less (or with an arbitrarily small mass or volume) and are subject to velocities and accelerations towards a better mode of behavior. Each particle in the swarm represents a solution in a high-dimensional space with four vectors, its current position, best position found so far, the best position found by its neighborhood so far and its velocity and adjusts its position in the search space based on the best position reached by itself (pbest) and on the best position reached by its neighborhood (gbest) during the search process [13].

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).

Honey Bee Mating Algorithm: Honey-bees mating may also be considered as a typical swarm-based approach to optimization, in which the search algorithm is encouraged by the process of mating in real honey-bees. The behavior of honey-bees is the interaction of their (1) Genetic potentiality. (2) Ecological and physiological environments, and (3) The social conditions of the colony, as well as various earlier and ongoing interactions between these three parameters. Each bee undertakes sequences of actions which unfold according to genetic, ecological, and social conditions of the colony [14].

Honey-Bees Mating Optimization (HBMO) algorithm may be constructed with the following five main stages:
1. The algorithm starts with the mating–flight, where a queen (best solution) selects drones probabilistically to form the spermatheca (list of drones). A drone is then selected from the list at random for the creation of broods.
2. Creation of new broods (trial solutions) by crossoverring the drones’ genotypes with the queen’s.
3. Use of workers (heuristics) to conduct local search on broods (trial solutions).
4. Adaptation of workers’ fitness based on the amount of improvement achieved on broods.
5. Replacement of weaker queens by fitter broods.

IV. CONCLUSION

In Cloud computing there are many existing issues like Load Balancing, virtual machine migration, Energy management etc, which have not been fully addressed. Central of these issues the main issue is load balancing, that is required to distribute the dynamic local workload to all the nodes in the whole cloud to achieve a higher satisfaction and resource utilization ratio. This paper presents a concept of load balancing and its algorithms.

REFERENCES:

1. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, “ Cloud Computing A Practical Approach”, TATA McGRAW-HILL Edition 2010, pp.334
2. B. P. Rima, E. Choi, and I. Lumb, “A Taxonomy and Survey of Cloud Computing Systems”, Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pp. 44-51
3. A. M. Alakeel, “A Guide to dynamic Load balancing in Distributed Computer Systems”, International Journal of Computer Science and Network Security (IJCSNS), Vol. 10, No. 6, June 2010, pp. 153-160.
4. Saeed Javanmardi, MohammadShojafar, Danilo Amendola, Nicola Cordeschi, “ Hybrid Job scheduling Algorithm for Cloud computing Environment”, adfa, 2014, pp.2
5. Hitesh A. Ravani, Hitesh A. Bheda, “Genetic Algorithm Based Resource Scheduling Technique in Cloud Computing”, International Journal of Advance Research of Computer Science and Management Studies, Volume 1,Issue 7,Fecember 2013,ISSN:2321-7782
6. Florin Pop, Valentin Cristea, Nik Bbbessis, Stelios Sotiriadis, “Reputation guided Genetic Scheduling Algorithm for Independent Tasks in Inter-Clouds Environments”, International Conference on Advanced Information Networking and Applications Workshops, 2013, pp. 772-776
7. Jianfeng Zhao, Wenhua Zena, Min Liu, Guangming Li, “Multi-Objective Optimization Model of Virtual Resources Scheduling under Cloud Computing and It’s Solution”, International Conference on Cloud and Service Computing, Grant no 60903129, 2011
8. 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
9. Andrew J. Younge, Gregor von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, warren Carithers, “Efficient Resource Management for Cloud Computing Environments”, IEEE, 2010, pp.
10. Yu-lung Lo, Min-Shan Lai , “The Load Balancing of Data base Allocation in the Cloud”, International MultiConference of Engineers and Computer Scientists, Volume I, March 13-15,2013, ISSN: 2078-0958
11. D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, “The Bees Algorithm-A Novel Tool for Complex Optimisation Problem”, Cardiff CF243AA UK, 2012
12. Er. Shimpy, Mr. Jagandeep Sidhu, “Different Scheduling Algorithms In Different Cloud Environment”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 9, September 2014, ISSN: 2278-1021
13. Binitha S, S Siva Sathya, “A Survey of Bio inspired Optimization Algorithms”, International Journal of Soft Computing and Engineering, Vol. 2, Issue 2, May 2012, ISSN: 2231-2307
14. Omid Bozorg Haddad, Abbas Afshar, Miguel A. Marino, “Honey Bees Mating Optimization (HBMO) Algorithm”, Water Resource Management, DOI: 10.1007/s11269-005-9001-3, 2006, pp. 661-680