An Efficient Multi-Objective Genetic Algorithm for Optimization of Task Scheduling in Cloud Computing

Mohd Rahul
Research Scholar OPJS University,
Churu, Rajasthan, India
Email: mohd.rahul@gmail.com

Dr. Vaibhav Bansal
Associate Professor - CSE
OPJS University,
Churu, Rajasthan, India

Abstract-**Cloud computing is a type of computing based on internet that provides shared processing resources and data to computers and other devices on request. Now a day's availability of high capacity networks low cost computers and storage devices as well as the global adoption of hardware virtualization service-oriented architecture and autonomic and utility computing have guide to a growth in cloud computing. cloud computing has become a highly demanded service or utility due to the advantages of fast computing power, low cost of services, high performance, scalability, accessibility as well as availability. But in cloud computing task scheduling and provision of resources are main problem areas in cloud computing. However there are so many algorithms are given by various researchers for clouds, but none of the existing algorithms have considered the multi-objective approach to minimize the cost and completion time of a set of tasks for better scheduling of jobs to the resources. Therefore the research work is motivated to derive the task scheduling based on multi-objective genetic algorithm employs an improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in cloud computing environment.**

General Terms: Fault tolerance, network bandwidth, task scheduling, virtualised environment, hybrid clouds, data security

Keywords: Cloud components, clients, distributor server, types of clouds, services provided by cloud computing, saas, paas, haas, task scheduling in cloud computing.

1. INTRODUCTION

1.1 Cloud Computing

Cloud computing is an on demand service in which shared resources, information, software and other devices are given according to the clients requirement at specific time. It's a term which is generally used in link of Internet. The entire Internet can be viewed as a cloud. Capital and operational costs can be cut using cloud computing. In contention of Cloud computing services can be used from diverse and widespread resources, instead of remote servers or local machines. There is no specific definition of Cloud computing. Generally it consists of a group of distributed servers called as masters, providing necessitate services and resources to different clients in a network with datacenter reliability and communion. The distributed computers provide on-demand services. Services may be of software resources i.e SaaS (e.g. Software as a Service) or physical resources (e.g. Platform as a Service, (PaaS), hardware/infrastructure Hardware as a Service (HaaS) or Infrastructure as a Service (IaaS). Amazon Elastic Compute Cloud (Amazon EC2) is an example of cloud computing services [2].

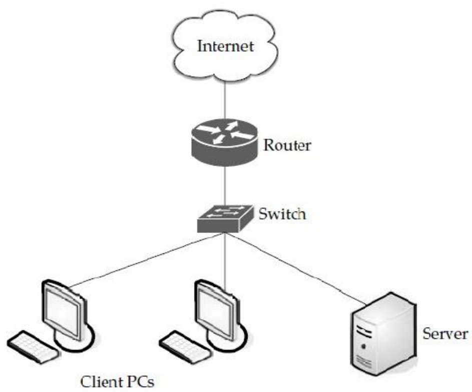

Fig. 1.1: A cloud is used in the network diagrams to depict the internet [1]

1.2 Cloud Components

A Cloud system consists of three major components such as clients, distributed and datacenter servers. Each element has a definite motive and plays a specific role.

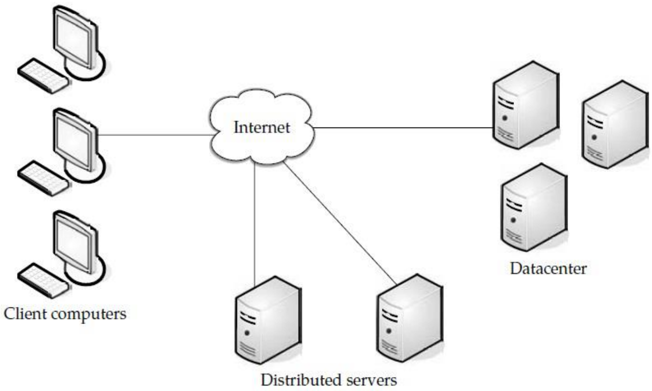

Fig. 1.2: Three components make up a cloud computing solution [1]

1.2.1 Clients

End users interact with the clients to manage information linked to the cloud. Clients generally came into three categories as given in [3]:

•Mobile: Windows Mobile Smartphone, Smart phones example Blackberry, or iPhone etc.

•Thin: These clients don't do any computation work. They only display the information. For them Servers do all the works. These clients don't have any internal memory.

•Thick: These use different browsers such as IE, Mozilla Firefox or Google Chrome to link to the Internet cloud.

Now-a-days thin clients are more popular as contrast to other clients because of the low price, security, low consumption of power, less noise, easily repairable.

1.2.2 Datacenter

Datacenter is a assembly of servers organizing different applications. An end user linked to the datacenter to take different applications. A datacenter may situated at a great distance to the clients. Now-a-days a method called virtualization is used to install software that permits multiple instances of virtual server applications.

1.2.3 Distributed Servers

Distributed servers are the parts of a cloud which are available all through the Internet, hosting various applications. But during using the application from the cloud, the user will think that he is using this application from its own machine.

1.3 Type of Clouds

Rely on the domain or environment in which clouds are used, clouds can be divided into three categories:
1. Public Clouds
2. Private Clouds
3. Hybrid Clouds (combination of both private and public clouds)

1.4 Virtualization

It is a very useful concept in terms of cloud systems. Virtualization is "something which is not real", but support all the facilities of a real. It is the software of a computer which will execute various programs like a real machine. Virtualization is linked to cloud, because taking virtualization an end user can use various services of a cloud. The remote datacenter will provide various facilities in a full or partial virtualized fashion.

Two types of virtualization are endowed in case of clouds as given in:

•Full virtualization
•Para virtualization [4].

1.4.1 Full Virtualization

In term of full virtualization a entire installation of one machine perform on another machine. It will derived in a virtual machine which will have all the software's that are also in the actual server [1].

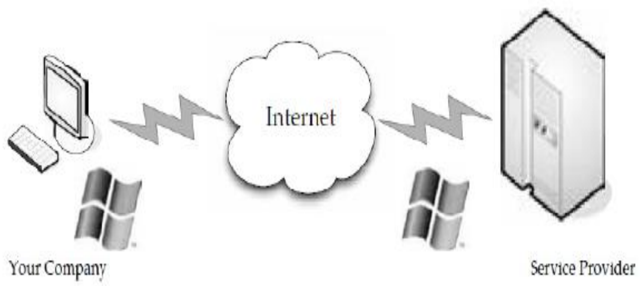

Fig.1.3: Full Virtualization [1]

1.4.2 Para virtualization

The hardware grants multiple operating systems to run on single machine by effective use of system devices namely memory and processor In Para virtualization. Example VMware software. Here all the services are not completely available, instead the services are provided partially [5].

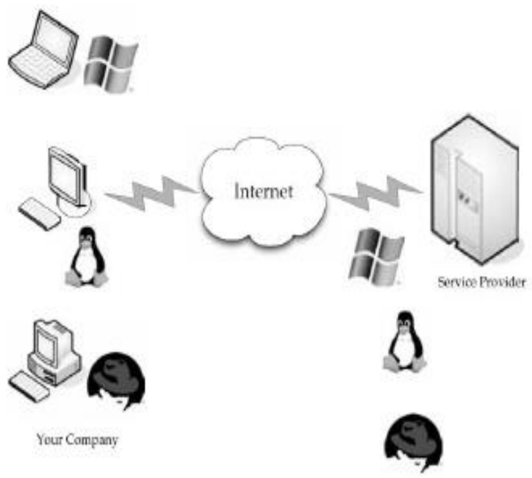

Fig. 1.4: Para virtualization [1]

1.5 Services provided by Cloud computing

Service means various types of applications given by different servers over the cloud. It is generally given as a service. Services in a cloud are of three types as given in [2]:

- SaaS: Software as a Service
- PaaS: Platform as a Service
- HaaS: Hardware as a Service or
- IaaS: Infrastructure as a Service

1.5.1 Software as a Service (SaaS)

In SaaS, the user uses various software applications from different servers through the Internet. The user utilizes the software as it is not requiring any change and do not need to make lots of changes or doesn't require combination to other systems. The provider does all the changes and patching while keeping the infrastructure running [17].The client have to pay for software for the time he uses. The software that does a simple task without any need to interact with other systems makes it an ideal candidate for Software as a Service. Customer who isn't prefer to perform software development but requires high-powered applications can also be benefitted from SaaS. Some of these applications include [6]:

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

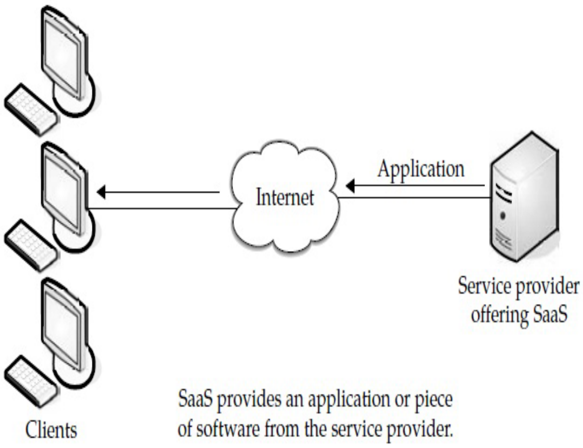

Fig. 1.5: Software as a service (SaaS) [1]

1.5.2 Platform as a Service (PaaS)

PaaS provides all the resources that are needed for building applications and services completely from the Internet, without downloading or installing software [10].
PaaS services include software design, development, testing, deployment, and hosting. Some Other services can be team cooperation, database integration, web service integration, data security, storage and versioning etc [1].

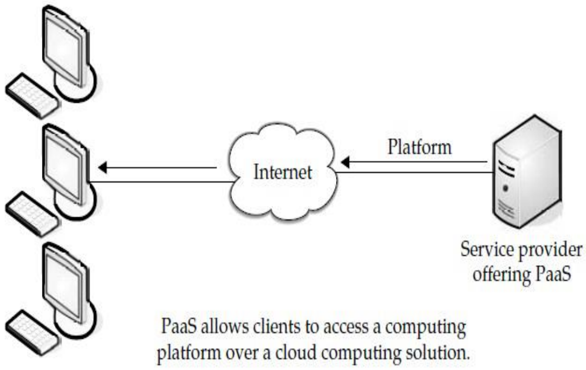

Fig. 1.6: Platform as a service (PaaS) [1]

1.5.3 Hardware as a Service (HaaS)

It is also known as IaaS (Infrastructure as a Service). It gives the hardware as a service to a organization so that it can put anything into the hardware according to its will [1].

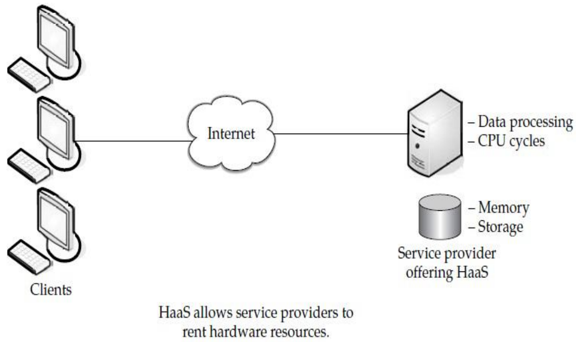

Fig. 1.7: Hardware as a service (HaaS) [9]

1.6 Task Scheduling in Cloud Computing

Scheduling refers to the set of strategies to control the order of work to be performed by a computer system. Job scheduling is very demanding task in cloud computing because it has parallel and distributed design. The job completion time calculation is difficult in cloud because the job may be divided between more than one Virtual machine. Task scheduling and allocation of resources are main problem areas in both Grid and in cloud computing. Cloud computing is transpiring technology in IT sector. The scheduling of the cloud services to the consumers by service providers affects the cost benefit of these computing standards. Though, there are so many algorithms are presented by various researchers for task scheduling. In 2008, a heuristic approach to schedule bag-of-tasks (tasks with less execution time and no dependencies) in a cloud is presented so that the number of virtual machines to execute all the tasks within the budget is lower and the even time speedup. In 2009, Marios D. Dikaiakos and George Pallis suggested the method of organization of Distributed Internet Computing as Public benefit and addressed the several significant problems and undeveloped opportunities concerning the deployment, effective operations and use of cloud computing infrastructures [7].

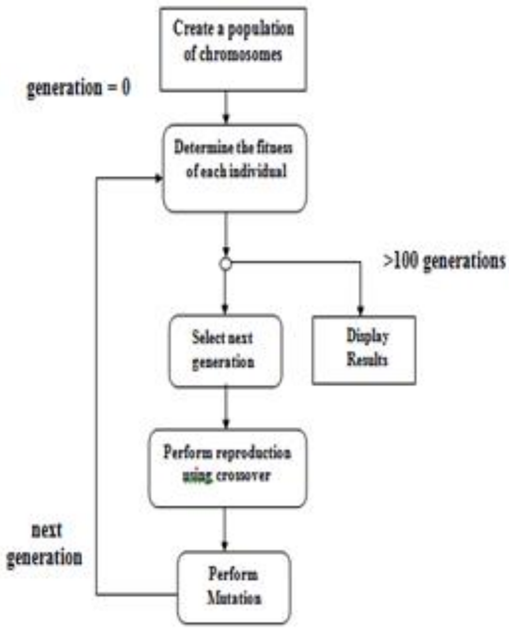

Fig.1.8: Standard Genetic Algorithm (SGA) [6]

A GA (Genetic algorithm) is a search heuristic that imitate the process of natural evolution. This heuristic is routinely used to produce useful solutions to optimization and search problems [4].Genetic algorithms belong to the huge class of evolutionary algorithms (EA), which produce solutions to optimization problems using procedures inspired by natural extension, such as inheritance, mutation, selection, and crossover. Although, the appropriate representation of potential solutions is critical to ensure that the variation of any set of individual (i.e. chromosome) will result in new proper and meaningful individual for the problem. An output scheme of tasks is an array of population called chromosomes or the genotype of the genome, which put into code candidate solutions to an optimization problem, evolves toward better results. Time minimization will give gain to service provider and fewer maintenance cost to the resources [8].

2. RELATED WORK

The research work performed in this field by different researchers is presented as follows:

Verma A. et al. [11] suggested a modified genetic algorithm for single user jobs in which the fitness was developed to encourage the formation of solutions to attain the time minimization and compared it with existing heuristics. Experimental results showed that, under the heavy loads, the suggested algorithm gives a good performance.

Shuai G. et al [12] introduced operating costs, the demanding of cloud computing environment was not high to the network, so a good resource scheduling algorithm was an important key. The genetic simulated annealing algorithm considered the QOS requirements of various type tasks, corresponding to the properties of user tasks in cloud computing environment. As the measurements of parameters are different and orders of magnitude were very different, the parameters were dealt with dimensionless. The experimental results showed that the algorithm efficiently completed resource search and allotment in the cloud computing environment computing.

Sigalas P. et al. [13] introduced improved independent task scheduling heuristics assuming a heterogeneous distributed processing environment. Among them, the GA-based heuristic (IGA) offered the most promising results since apart from providing the smallest makespan (in most cases), it also resulted in minimum node completion time, without affecting much the average task completion time. In an extended version a plan made to extensively evaluate all heuristic approaches and possibly incorporate other.

Ramachandram S. et al. [14] presented genetic algorithm for grid scheduling using best rank power, to schedule the independent tasks onto Grid computing resources. Roulette Wheel Selection Genetic Algorithm using Best Rank Power (PRRWSGA) speeds up convergence and shortens the search time. The heuristic initialization of current population using MCT (minimum completion time) algorithm allow the PRRWSGA to find out a high quality feasible scheduling solution. According to the simulation results, PRRWSGA algorithm had better search performance than both IRRWSGA and standard genetic algorithms in sense of time. Also PRRWSGA algorithm had better search performance than standard genetic algorithm (SGA) in sense of quality of the schedule. Although, SGA had problem when scaled up, while PRRWSGA owns robustness and scalability features. All that, the limited iteration with a feasible result made PRRWSGA algorithm suitable for realistic scheduling in Grid environment.

Zhu X. et al. [15] introduced genetic simulated annealing a hybrid random searching technique fusing GA (genetic algorithm) and SA (simulated annealing), inheriting the convergence property of SA and parallelization capability of GA. So, the performance of the hybrid algorithm was more excellent than SA and GA respectively.

3. PROPOSED WORK

3.1 Problem Formulation

Important part of cloud computing is task scheduling. Task scheduling is a mapping technique from users tasks to the appropriate selection of resources and its execution. By using the technology of virtualization, all physical devices are virtualized and translucent for users. All users have their own virtual device, these devises do not connect with each other and they are created based on users' requirements. Moreover, one or more virtual machines can run on a single host computer so that the usage rate of resources has been effectively improved. Because of these new features, grid computing, the original task scheduling procedure can't work effectively in cloud computing environments. Task scheduling and planning of resources are main problem parts in both Grid as well as in cloud computing. Concern to related work study, we summarized that the existing scheduling strategies in clouds are based on the techniques developed in related areas like distributed systems and Grids. Scheduling in these parts is basically tailored toward ensuring single application Service Level Agreement (SLA) objectives.

Task scheduling depend on Multi-Objective Genetic algorithm in cloud computing environment has come up as a promising solution strategy for all these stated concerns of Cloud computing.

3. 2 Proposed Work

In the paper, task scheduling approach based on Multi-Objective Genetic algorithm (MOGA) in cloud computing environment implies an improved cost-based scheduling algorithm for making effective mapping of tasks to available resources in cloud computing environment. This scheduling algorithm measures to achieve the minimization of both resource cost and computation performance.

4. RESULTS AND ANALYSIS

the objective to analyze the performance of genetic algorithm in minimizing the makespan and cost of the processor and to find the best scheduling of jobs on the resources, MOGA (Multi-Objective genetic algorithm) is implemented on Intel core i3 machine with 500 GB HDD, 4 GB RAM on Windows 8 OS, Java language version 1.6 and simulation work is performed using CloudSim3.0 [54] toolkit on Netbeans IDE 7.3. To formulate the problem we considered Jobs (J1, J2, J3…..Jn) run on Resources (R1, R2, R3…..Rn). Our objective is to minimize the Make-span and Cost. The speed of processors is expressed in Million instructions per second (MIPS) and length of job can be defined as number of instructions to be executed. Each processor is assigned different processing power and respective cost in Indian rupees. Our objective is to compute the Make-span (completion time of tasks) and the corresponding Cost of output schedules from the proposed algorithm and compare the result with existing Improved ABC algorithm.

4.1 Executing the algorithms

Run the algorithm manually. Right click on Main.java file and select run as Java Application.
Input: Running the algorithm manually.

For the execution of the coding, first the source file of the coding must be imported and then it is explored to get the main.java file. Once the main .java is right clicked, then select run as option. The coding is being executed. Following parameters are used during simulation of task scheduling algorithm:

Table 4.1: GA Parameters

Parameter	Value
No. of Resources	6
No. of Jobs	25-100
Population size	5
Number of Iterations	1000
Crossover Type	Two-point Crossover
Crossover Probability	0.5
Mutation Type	Simple swap
Mutation Probability	0.015
Selection method	Tournament
Termination Condition	Number of Iterations

Table 4.2: List of Resources

Resource Id	Processor Capacity(Mips)
R1	120
R2	131
R3	153
R4	296
R5	126
R6	210

Output:

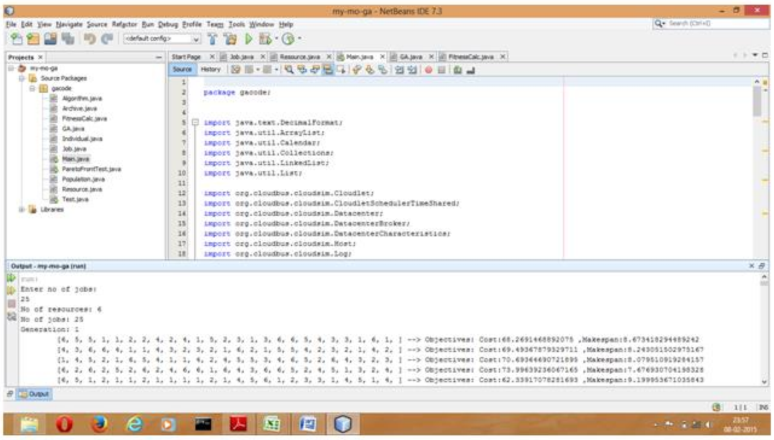

Fig.4.1 Executing the algorithm manually

Table 4.3: Cost and Makespan when generation=100 and Population size=5

Generation: 100	
Pop. Size 1	[3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 ,Makespan:10.505124055674118
Pop. Size 2	[3, 5, 2, 5, 2, 2, 5, 5, 4, 2, 2, 5, 6, 1, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:60.138163493855004 ,Makespan:10.410979911529974
Pop. Size 3	[3, 5, 2, 5, 2, 2, 1, 5, 4, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:61.86191423478132 ,Makespan:10.015865194850374
Pop. Size 4	[3, 5, 2, 3, 2, 6, 1, 5, 4, 2, 2, 5, 6, 1, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:63.63047377674678 ,Makespan:10.090207607697502
Pop. Size 5	[3, 5, 2, 5, 2, 6, 5, 5, 4, 6, 2, 5, 6, 4, 1, 5, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:66.5546774723919 ,Makespan:9.611560699515344
Archive:	
[3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 ,Makespan:10.505124055674118	
[3, 5, 2, 5, 2, 2, 5, 5, 4, 2, 2, 5, 6, 1, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:60.138163493855004 ,Makespan:10.410979911529974	
[3, 5, 2, 5, 2, 2, 1, 5, 4, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:61.86191423478132 ,Makespan:10.015865194850374	
[3, 5, 2, 5, 2, 6, 5, 5, 4, 6, 2, 5, 6, 4, 1, 5, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:66.5546774723919 ,Makespan:9.611560699515344	
[3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 ,Makespan:10.505124055674118	
Best: [3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,]Cost:59.646046376737885 ,Makespan:10.505124055674118	

Table 4.4: Cost and Makespan when generation=200 and Population size=5

Generation: 200	
Pop. Size 1	[3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 ,Makespan:10.505124055674118
Pop. Size 2	[3, 5, 2, 5, 2, 2, 5, 5, 4, 2, 2, 5, 6, 1, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:60.138163493855004,Makespan:10.410979911529974
Pop. Size 3	[3, 5, 2, 5, 2, 2, 1, 5, 4, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:61.86191423478132 ,Makespan:10.015865194850374
Pop. Size 4	[3, 5, 2, 3, 2, 6, 1, 5, 4, 2, 2, 5, 6, 1, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:63.63047377674678 ,Makespan:10.090207607697502
Pop. Size 5	[3, 5, 2, 5, 2, 6, 5, 5, 4, 6, 2, 5, 6, 4, 1, 5, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:66.5546774723919 , Makespan:9.611560699515344
Archive:	
[3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 , Makespan:10.505124055674118	
[3, 5, 2, 5, 2, 2, 5, 5, 4, 2, 2, 5, 6, 1, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:60.138163493855004 , Makespan:10.410979911529974	
[3, 5, 2, 5, 2, 2, 1, 5, 4, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:61.86191423478132 , Makespan:10.015865194850374	
[3, 5, 2, 5, 2, 6, 5, 5, 4, 6, 2, 5, 6, 4, 1, 5, 1, 1, 2, 2, 2, 2, 5, 2, 2,] Cost:66.5546774723919 , Makespan:9.611560699515344	
[3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 , Makespan:10.505124055674118	
Best: [3, 5, 2, 5, 2, 2, 5, 5, 1, 2, 2, 5, 6, 4, 1, 2, 1, 1, 2, 2, 1, 2, 5, 2, 2,] Cost:59.646046376737885 , Makespan:10.505124055674118	

Table 4.5: Cost and Makespan when generation=300 and Population size=5

Generation: 300	
Pop. Size 1	[2, 2, 2, 5, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:53.74716769659518 ,Makespan:10.749433539319032
Pop. Size 2	[2, 2, 2, 5, 2, 2, 5, 5, 2, 2, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:53.81078092814734 ,Makespan:10.762156185629465
Pop. Size 3	[2, 2, 2, 5, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:53.74716769659518 ,Makespan:10.749433539319032
Pop. Size 4	[2, 2, 2, 5, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:53.74716769659518 ,Makespan:10.749433539319032
Pop. Size 5	[2, 2, 2, 5, 2, 2, 5, 2, 2, 6, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:55.67628438143705 ,Makespan:10.582875923906455
Archive:	
[2, 2, 2, 5, 2, 2, 5, 2, 2, 6, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:55.67628438143705 , Makespan:10.582875923906455	
[2, 2, 2, 5, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:53.74716769659518 , Makespan:10.749433539319032	
Best: [2, 2, 2, 5, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 5, 2, 2, 5, 2, 2, 2, 2,] Cost:53.74716769659518 , Makespan:10.749433539319032	

Table 4.7: Cost and Makespan when generation=500 and Population size=5

Generation: 500	
Pop. Size 1	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2,] Cost:53.373621713316375 ,Makespan:10.674724342663271
Pop. Size 2	[2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 4, 2, 2, 5, 2, 5, 5, 2,] Cost:54.8793870206084 ,Makespan:10.519796323040595
Pop. Size 3	[2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2,] Cost:54.13498478271716 ,Makespan:10.643990420595717
Pop. Size 4	[2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 4, 2, 2, 5, 2, 5, 5, 2,] Cost:54.963356486257254 ,Makespan:10.536590216170366
Pop. Size 5	[2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 5, 5, 2,] Cost:53.55640373197625 ,Makespan:10.711280746395248
Archive:	
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2,] Cost:53.373621713316375 , Makespan:10.674724342663271	
[2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 4, 2, 2, 5, 2, 5, 5, 2,] Cost:54.8793870206084 , Makespan:10.519796323040595	
[2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2,] Cost:54.13498478271716 , Makespan:10.643990420595717	
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2,] Cost:53.373621713316375 , Makespan:10.674724342663271	
Best: [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2,] Cost:53.373621713316375 , Makespan:10.674724342663271	

Table 4.10: Cost and Makespan when generation=800 and Population size=5

Generation: 800	
Pop. Size 1	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 ,Makespan:10.62595419847328
Pop. Size 2	[2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:54.65855168145245 ,Makespan:10.404683309263461
Pop. Size 3	[2, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2,] Cost:54.37245547073792 ,Makespan:10.607824427480915
Pop. Size 4	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 ,Makespan:10.62595419847328
Pop. Size 5	[2, 2, 2, 2, 4, 2, 2, 4, 2, 2, 2, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 6, 2, 2,] Cost:55.93026286919417 ,Makespan:10.29564073342699
Archive:	
[2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:54.65855168145245 , Makespan:10.404683309263461	
[2, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2,] Cost:54.37245547073792 , Makespan:10.607824427480915	
[2, 2, 2, 2, 4, 2, 2, 4, 2, 2, 2, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 6, 2, 2,] Cost:55.93026286919417 , Makespan:10.29564073342699	
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 , Makespan:10.62595419847328	
Best: [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 , Makespan:10.62595419847328	

Table 4.12: Cost and Makespan when generation=1000 and Population size=5

Generation: 1000	
Pop. Size 1	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 ,Makespan:10.62595419847328
Pop. Size 2	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 ,Makespan:10.62595419847328
Pop. Size 3	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 ,Makespan:10.62595419847328
Pop. Size 4	[2, 2, 5, 2, 1, 2, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:55.66799951532776 ,Makespan:10.457409426875074
Pop. Size 5	[2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.22773536895674 ,Makespan:10.645547073791347
Archive:	
[2, 2, 5, 2, 1, 2, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:55.66799951532776 , Makespan:10.457409426875074	
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 , Makespan:10.62595419847328	
Best: [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,] Cost:53.12977099236642 , Makespan:10.62595419847328	

After getting the values of Cost and Makespan at 1000th generation then we have best schedule.

Output of Best schedule:

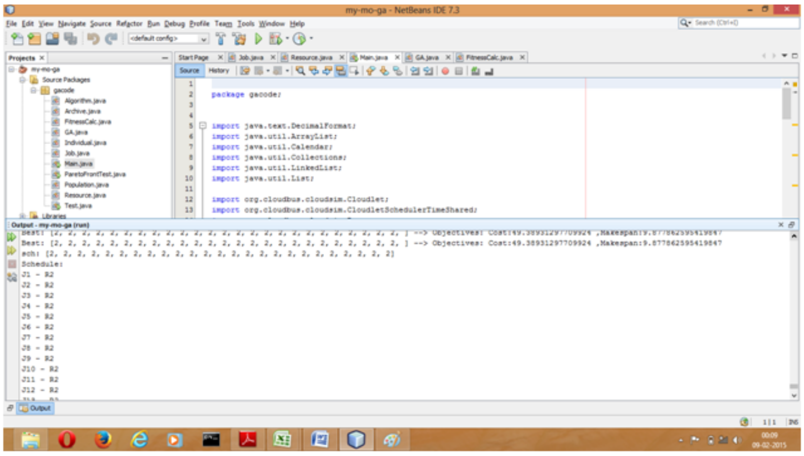

Fig.4.5 Best scheduling at the Population Generation=1000

Table 4.4 Best scheduling of Jobs on Resources at Pop. Generation=1000

Jobs	Resources
J1	R2
J2	R2
J3	R2
J4	R2
J5	R2
J6	R2
J7	R2
J8	R2
J9	R2
J10	R2
J11	R2
J12	R2
J13	R2
J14	R2
J15	R2
J16	R2
J17	R2
J18	R2
J19	R2
J20	R2
J21	R2
J22	R2
J23	R2
J24	R2
J25	R2

4.2 Cloud simulation

There are some steps of Cloud simulation. The steps are following:

- Datacentre entity registers with the CIS Registry.
- CIS then provides information registry-type practicality, like as match-making services for mapping user/brokers, requests to suitable Cloud providers.
- Datacentre brokers acting on part of users consult the CIS service to obtain the record of cloud providers who can offer infrastructure services that relate application’s QoS, hardware, and software requirements.

•In the matter of a match, the DataCenter broker place the application with the CIS suggested cloud.

The transmission flow defined so far linked to the basic flow in a simulated experiment. Some differences in this flow are possible rely on policies. For example, messages from Brokers to Datacenters may require verification from other sections of the Datacenter, about the execution of an action, or regarding the maximum number of VMs that a user can create.

Output: Cloudlets executed on virtual machines.

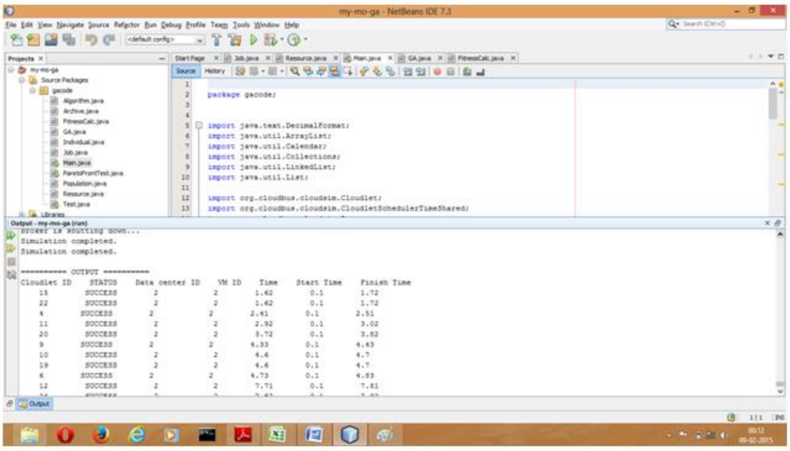

Figure 4.3 Cloudlets executed on Virtual machines

Table 4.5 Cloudlets executed on Virtual machines

Cloudlet ID	STATUS	VM ID	Time	Start Time	Finish Time
11	SUCCESS	2	3.1	0.1	3.2
13	SUCCESS	2	3.47	0.1	3.57
23	SUCCESS	2	3.47	0.1	3.57
2	SUCCESS	2	4.97	0.1	5.07
5	SUCCESS	2	4.97	0.1	5.07
22	SUCCESS	2	5.43	0.1	5.53
4	SUCCESS	2	5.98	0.1	6.08
12	SUCCESS	2	6.12	0.1	6.22
24	SUCCESS	2	6.25	0.1	6.35
8	SUCCESS	2	6.9	0.1	7
18	SUCCESS	2	7.24	0.1	7.34
3	SUCCESS	2	7.99	0.1	8.09
25	SUCCESS	2	8.49	0.1	8.59
10	SUCCESS	2	8.6	0.1	8.7
7	SUCCESS	2	9.69	0.1	9.79
15	SUCCESS	2	9.8	0.1	9.9
14	SUCCESS	2	9.91	0.1	10.01
17	SUCCESS	2	9.91	0.1	10.01
19	SUCCESS	2	10.02	0.1	10.12
1	SUCCESS	2	10.13	0.1	10.23
6	SUCCESS	2	10.28	0.1	10.38
21	SUCCESS	2	10.44	0.1	10.54
9	SUCCESS	2	10.55	0.1	10.65
20	SUCCESS	2	10.55	0.1	10.65
16	SUCCESS	2	10.66	0.1	10.76

4.3 **Comparison of Improved ABC Algorithm with MOGA**

The results are compared for processing time and processing cost for varying numbers of Cloudlets namely 25, 50, 75 and 100. Table 4.16. From the below Figure it can be seen that for MOGA scheduling the processing cost spent to complete tasks is very less when compared with the processing cost spent to complete the tasks with IABC algorithm. In Table 4.17. From the below Figure it can be seen that for MOGA scheduling the time taken to complete tasks is very less when compared with time taken to complete the tasks with Improved ABC Algorithm. From the above graphs, which show the comparison of completion time taken and processing cost spent for MOGA scheduling algorithm and Improved ABC scheduling algorithm, we can conclude that the MOGA scheduling algorithm is better than Improved ABC scheduling algorithm.

Table 4.16: Simulation of Processing Cost for Improved ABC Algorithms and MOGA (Multi-objective Genetic Algorithm)

No. of Cloudlets	Improved ABC Algorithm (Processing Cost in Rs.)	MOGA (Processing Cost in Rs.)
25	72.34	53.12
50	374.01	105.13
75	402.61	141.53
100	543.32	203.17

5. CONCLUSION AND FUTURE SCOPE

This paper have concluded that the proposed approach leads to the better results in a multi-objective genetic algorithm on cloud computing environment for scheduling of jobs on the processors. The capability is developed to encourage the formation of solutions to achieve both the cost and makespan minimization. The performance of MOGA (on minimization of cost and makespan of the resources) with the variation of its control parameters is evaluated. Increasing the Population generation and making population size constant enables the genetic algorithm to obtain minimum Cost and Makespan which results in a better scheduling. From the comparison of completion time taken and processing cost spent for MOGA scheduling algorithm and Improved ABC scheduling algorithm, concluded that the MOGA scheduling algorithm is better than Improved ABC scheduling algorithm. In future, study has to be done to select the correct parameter. It is designed to extend the suggested work for dynamic scheduling of both dependent and independent tasks considering different objectives. It is designed to compare the simulated results with other scheduling algorithms also. It can also be extended by considering the motive like maximizing the processor utilization. This simulation will be tested on the real world scenario.The simulation is tested for reliability of fitness.

REFERENCES

[1]Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGraw-HILL Edition 2010, pp. 3-11.

[2]Randles M., Lamb D., Taleb-Bendiab A.,"A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 2010, IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, pp. 551-556.

[3]Dikaiakos, M., katsaros, D., Mehra, P., Vakali, A., "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Transactions on Internet Computing, Volume-13, Issue-5, 2009, pp. 10-13.

[4]Jhawar R., Piuri V., Santambrogio M. "Fault Tolerance Management in Cloud Computing: A System level Perspective", Systems Journal, IEEE, Volume-7, Issue-2, June, 2013, pp. 288-297.

[5]Khethavath, P., Thomas, J., Chan-Tin, E., Hong Liu,"Introducing a distributed cloud architecture with efficient resource discovery and optimal resource allocation", Services (SERVICES) IEEE Ninth World Congress, July, 2013, pp. 386-392.

[6]Kaur S.,Verma A.," An Efficient approach to genetic algorithm for task scheduling in cloud computing environment", I.J. Information Technology and Computer Science, 2012, 10, pp.74-79.

[7]Tayal S., "Tasks Scheduling Optimization for the Cloud Computing Systems", International Journal of Advanced Engineering Sciences and Technologies, Volume-5, Issue No.-2, 2011, pp. 111-115.

[8]Zhao L., Ren Y., Sakurai K., "A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems", International Conference on Advance Information Networking and Applications, IEEE, 2011, pp. 275-282.

[9]Han Zhao, Xiaolin Li, "AuctionNet: Market oriented task scheduling in heterogeneous distributed environments", IEEE, Atlanta, GA, 2010, pp.1-4.

[10]Emeakaroha V.C., Brandic I., Maurer M., Breskovic I., "SLA-Aware Application Deployment and Resource Allocation in Clouds", IEEE, 2011, pp. 298-303.

[11]Sureshbabu G.N.K, Srivatsa S.K.," A Review of Load Balancing Algorithms for Cloud Computing", International Journal of Engineering and Computer Science, Volume-3, Issue-9 September, 2014, pp. 8297-8302.

[12] Guo G., Ting-Iei H., Shuai G., " Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 60-63.

[13] Loukopoulos T., Lampsas P., Sigalas P., "Improved Genetic Algorithms and List Scheduling Techniques for Independent Task Scheduling in Distributed Systems", IEEE, Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies, Adelaide, SA, 2007, pp. 67-74.

[14] Abdulal W., Jadaan O.A., Jabas A., Ramachandram S., "Genetic Algorithm for Grid Scheduling using Best Rank Power", IEEE World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009, Coimbatore, 2009, pp. 181-186.

[15] Wang J., Duan Q., Jiang Y., Zhu X., " A New Algorithm for Grid Independent Task Schedule： Genetic Simulated Annealing", IEEE World Automat ion Congress (WAC), 2010,Kobe, pp. 165-171.

[16] Mohd Rahul, Impact of Cloud Computing on IT Industry: A Review & Analysis, International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 01– Issue 02, November 2012

[17] Selvarani, S., Sadhasivam, G.S., "Improved cost-based algorithm for task scheduling in cloud computing, IEEE, 2011, pp. 1-5.