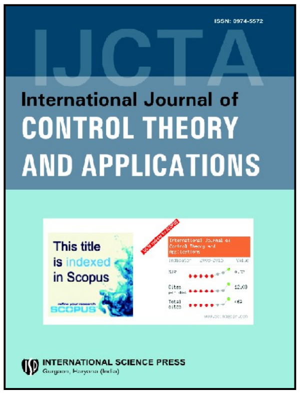


International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 29 • 2017

GAF – Genetic Algorithm based Framework for Cloud Resource Scheduling

R. Muthuram[1] and G. Kousalya[2]

[1] *Assistant Professor, Alagappa Chettiar College of Engineering and Technology, Karaikudi*
[2] *Professor, Coimbatore Institute of Technology, Coimbatore*

Abstract: Cloud computing is emerging and attractive distributed technology that provides various services to the users on pay as use model. Allocating the resources on large interconnected federated cloud to various user applications is one of the challenging issues in cloud computing. The goal of the scheduling is mapping user tasks to the appropriate resources to optimize the objective. In this, we proposed efficient genetic algorithms based scheduling algorithm with optimization of multiple objectives. The proposed algorithm optimizes the execution cost and makes span of user submitted applications. This framework starts with initial population and applies the various genetic operators like selection, crossover, and mutation and produces the optimal solution of job schedule. The performance of the proposed algorithm is compared with existing scheduling algorithms and results shows that proposed genetic algorithm is far better than existing.

Keywords: Cloud, scheduling, evolutionary algorithms, Meta heuristic, Load balancing.

1. INTRODUCTION

Cloud computing [10, 17] has become an emerging technology in the field of high performance distributed computing. Cloud computing has providing the on demand access of resources to the users. Cloud computing is getting high attention due to its features like reliability, scalability, information sharing and low cost. Cloud computing provides everything as a service to the business organizations like Infrastructure as a service (IaaS), Software as a service (SaaS), and Platform as a service (PaaS) with flexible environment and on-demand basis [17]. Cloud service provider consists of large number of data centres with set of resources with the services. The service providers' bundle the resources to the users depends upon the request by the user. Cloud computing provides the benefits for both the users and service providers. With the use of cloud, the user can reduce the cost of design, setting up the hardware and software, maintenance of the data centre setup. Also, the users can dynamically scaling up the resources that required for demand. To provide the scalability for the user, the service provider needs to manage the resources and distribute the resources dynamically with high resource utilization.

Virtualization [10] is the technology that supports the cloud computing for dynamically manage and distribute the resources to the users. For each request of the user, the virtualization instantiates the resource object with appropriate requirements. Execution of the application is done by creating the instance of the resource object and mapping the application to the resources which are available. Virtualization has the ability to carry the remapping of the resources dynamically in cloud to achieve the maximized resource utilization. To achieve maximum utilization, the applications should be scheduled and allocated to the resources properly. One of the important research issue in the cloud is how can schedule and allocate the resources to maximize the utilization.

Scheduling [1] is the process of mapping the tasks to the set of resources in the cloud service provider. Scheduled problem involves that the jobs are scheduled and allocated to the resources subject to the some set of constraints. Scheduling problem should be optimizing problem that maximize or minimize the parameters that related with the performance. Generally, the scheduling problem optimizes any of the following parameters: make span, throughput, flow time, cost, energy consumption and resource utilization. The problem of mapping the jobs to the unlimited computing resources in cloud computing belongs to a intractable problem. Genetic algorithm is an evolutionary algorithm that generates the optimal solution for the scheduling problems with multi dimensional search [2, 15].

In this paper, we proposed genetic algorithm based scheduling [13] for mapping the process. The proposed algorithm takes a make span and execution cost as the optimization parameters for generating the objective function. The algorithm performs the selection, cross over and mutation function for generating the optimal population chromosome. The proposed algorithm performances are compared with the algorithms: P-aware, DAS, and Optimal scheduling. The rest of this paper is organized as follows: Section 2 describes the related work, section 3 describes the detailed proposed scheduling architecture in cloud computing, section 4 gives the system model, section 5 gives the detailed proposed scheduling algorithm, section 6 describes the experimental setup and results and section 7 gives the concludes the paper.

2. RELATED WORK

Scheduling the tasks in the distributed inter connected network like cloud computing is complex issue in which lot of research works are going in this portion. Genetic algorithm is an evolutionary algorithm that used to generate the optimal schedule [13, 15] compared to other algorithms. Wang, Xu et al proposed a task scheduling algorithm [9] that uses the genetic algorithm to optimize the objective Make span and load distribution on virtual machines. The greedy technique is used to generate the initial population. Two types of fitness functions are defined, out of which one is selected randomly and used for each iteration. Their selection strategy is based on fitness ratio and they used self adaptive probabilities for crossover and mutation. Coninck, Verbelen, et al., [19] proposed a dynamic resource allocation algorithm for the workflow in the IaaS clouds. The proposed model contains the two components: Knowledge model which monitors and keeps the information of executed jobs, capacity planner which decides workers needed or the job based the information in the knowledge model. the algorithm calculated the make span needed or the job well in advance and based on that, it performs auto scaling of resources dynamically.

Gu, Hu & Sun had proposed resource scheduling policy [5] on load balancing of VM resources based on the genetic algorithm. Their algorithm finds the solution to mapping the resources to achieve the best load balancing [2, 15] by minimizing the cost divisor function. They used the tree structure encoding representation for the chromosome and each branch of the tree represents the schedule of the tasks. The selection process is performed by the fitness function ratio and uses the self adaptive variation probability. It considers the historical data and current status of VM to compute the schedule in advance and chooses the solution which has less influence of the system. Hang Zhou, Qing Li, et al, [18] proposed preciousness model for multi resource scheduling with proper load balancing and energy conservation. They used the PMDBP algorithm used to task mapping. In this paper to solve the scheduling problem, P-Aware manages to assign heterogeneous application jobs to proper server nodes in cluster in CDC. The proposed scheduler applied PMDBP algorithm to categorize the jobs based optimal proportion and map the job category to proper resource for high resource utilization.

Portaluri, Dorronsoro et al., proposed resource allocation strategy [16] in cloud data centres using genetic algorithm which minimizes the power consumption and cost expenditure. This framework performs the joint allocation of computational and network resources. They applied binary tournament selection strategy one point uniform crossover operator with probability 0.9. This algorithm shows the quadratic complexity dependency with respect to the number of tasks to be allocated. They performed experiments that outputs solution for schedule o two objectives, and power consumption values with variation of servers.

Sellami proposed a new algorithm for scheduling composite service workflows in cloud computing environment [6]. They proposed algorithm to optimize several objective function simultaneously depending upon the requirement of users and providers. In this, associative array with pairs of task and resource is used to represent the chromosome. Fitness function is calculated by summing of all the objectives and applies the normalization that values between 0 and 100. They applied double point crossover [4], random mutation and random selection to obtain the population. After the mutation, they applied the vaccination to correct the improper genes based on the constraint specification.

Wu & Li et al., proposed the new genetic algorithm [7] to place the virtual machines in the server consolidation [3] to optimize the energy consumption in the server and communication network in the data center. As compared to existing approaches, this framework considers not only the energy consumption of physical machines, also considers the energy consumption of communication networks. It applied the biased uniform cross over operator. Li Chunlin, Li La Yuan proposed optimal scheduling [20] on both private and public clouds. The proposed hybrid cloud scheduling policy considers the benefits of private cloud applications and public cloud provider, it can adapt to the changes in the system to find the scheduling optimization. The scheduling optimization is decomposed and conducted across the private cloud and public cloud.

Tao Feng et al., proposed the Pareto-optimal solution based GA [8] for workflow scheduling to optimize the energy conservation and make span. To solve the multi objective optimization, they implemented case library and Pareto solution based hybrid genetic algorithm (CLPS-GA). Major components of the model are multipoint crossover, two stage algorithm structure and case library. A case library consists of task type vectors, task dependency matrix, and corresponding Pareto solution. For generating the initial population, it refers the case library with more similarity that will selected; otherwise generates random population

3. PROPOSED CLOUD SCHEDULING STRUCTURE

Cloud computing made up of with centralized network with set of servers in distributed locations. Each of the servers consists of large number of physical machines. These physical machines perform the execution of applications on it. Generally, these cloud deployment models are divided into three categories: private cloud, public cloud, and hybrid cloud. Private cloud is dedicated to particular organization that provides secure cloud environment. Public cloud provides the services over the network to public usage. Hybrid cloud is integrated more than one cloud models like private and public that provides more benefits to the organizations. The cloud service provider consists of hybrid cloud with heterogeneous resources. The detailed scheduling architecture in cloud is described in Figure 1.

The proposed architecture consists of two entities: the user and cloud service provider. The functionality of the user is sends the request to submit the jobs to the service provider. The user request is gives the details of the job information like arrival time, number of instruction and deadline for completion etc. The user jobs are stored in the queue in the provider storage. The cloud service provider takes the jobs from the queue and performs the schedule based on the available hosts. The service provider consists of cloud service manager which manages the scheduling operation. The CSM consists of Meta scheduler which actually mapping the jobs with the resources. The CSM receives the user request information from the queue and host information from VM Resource Monitor (VRM) and mapping the jobs to the resources. VRM contains the information of available resources like type,

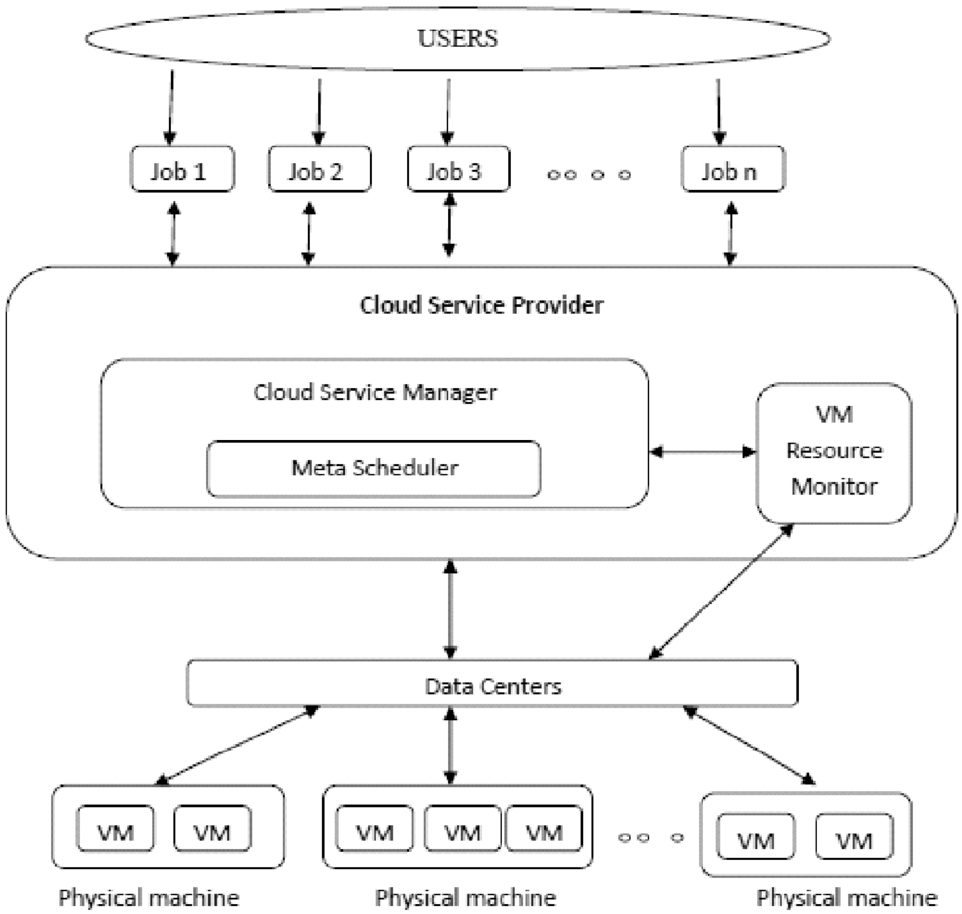


Figure 1: Cloud Scheduler Architecture

MIPS speed, cost of the execution, and status of the resources. VRM updates the availability status of the resources periodically. Virtualization technology instantiates the object of virtual machines on the physical machines and updates the status in the VRM. In this paper, Meta scheduler performs the Multi objective GA based scheduling [12] for the cloud service provider.

4. SYSTEM MODEL

In this proposed model, the user submits K number of the jobs to the service provider that maps into the various virtual machines M available in the host N. The main objectives of the proposed model are minimizing the cost of the user C and make span M. The proposed architecture consists of N number of physical machines that instantiates with various numbers of virtual machines. The VM resource Monitor (VRM) contains the information of virtual machines in the form as follows VM = (NI_{im}, P_{im}, d_{im}) where NI_{im} is number of instructions can be executed by virtual machine i in cloud m, P_{im} is cost of the VM, and d_{im} is delay cost of the VM. The user sends the request to submit the job j which contain in the following form JR = (t_j, IC_j, A_j, W_j) where t_j is type of the service required, IC_j is number of instruction counts that are available in the job, A_j is arrival time of the job, and W_j is worst case completion time of the job. The objective function for the scheduling is defined as follows

$$Cost\ C = \sum_{i=1}^{M}\sum_{m=1}^{N}\sum_{j=1}^{K}[P_{im} * (IC_j / NI_{im}) * \alpha + (1-\alpha)\, d_{im}]^* \quad (1)$$

$$Make\ span\ M = \sum_{i=1}^{M}\sum_{m=1}^{N}\sum_{j=1}^{K}(IC_j / NI_{im})^{*}\, Y_{jim} \qquad (2)$$

Our objectives are minimizing the execution cost C and make span M simultaneously subject to the following constraints:

$$\sum_{i=1}^{M}\sum_{m=1}^{N}\sum_{j=1}^{K}(IC_j / NI_{im}) \leq \sum_{c=1}^{K} W_c$$

$$\sum_{i=1}^{M}\sum_{m=1}^{N} M_{req} \leq M_c$$

where Mreq is memory required for that instance and M_c is memory available in cloud

5. PROPOSED ALGORITHM

Genetic Algorithm is adaptive heuristic algorithm that tries towards generate the best solutions based on the principle of "Survival of the fittest". Each candidate solution has some properties that altered and swapped to generate solutions. Genetic Algorithm starts with the initial population that randomly generated solutions. If the initial solution preferred with some knowledge about solutions, then it makes the algorithm to produce the solutions faster. Each candidate or chromosome shows the one solution which represented with some encoding format like binary encoding tree encoding etc.

Genetic algorithm performs the following operations on initial population: Selection, Crossover, and Mutation. In selection process, we selecting the fittest chromosome from the population based on fitness function. Some of the selection techniques are Roulette Wheel selection [4], Tournament Selection, Rank selection etc.

In Crossover, two populations are selected randomly and solutions are interchanged to generate more than one new population. It is done with crossover probability. The new crossover population is used to new population along with selected population. Some of the Crossover operators are one point, two point, and uniform crossover. Mutation is defined as a small random change in the population to generate a new solution. Mutation is performed with low probability rate. Some of the mutation operators are bit flip, swap, and inversion mutation etc.

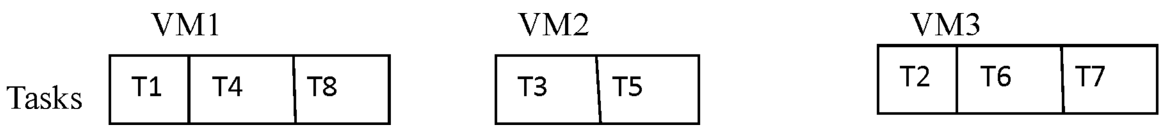
The proposed Multi objective Genetic algorithm for the scheduling is defined as follows in Figure 2.

Proposed Algorithm
Step 1: Create Initial Parent Population Step 2: Evaluate the fitness( ) for each chromosome in the parent population Step 3: Assign the rank and crowding distance for population Step 4: Repeat the following steps till the stopping criteria 4.1: Create the Child population( ) using selection and crossover 4.2: Apply the Mutation(child population) 4.3: Evaluate the fitness(child) for each chromosome in child 4.4: Apply merge(Parent, child, mixed) 4.5: Assign the rank and crowding distance 4.6: Apply Nondominationsort(mixed, parent) to select N individuals Step 5: Add final schedule to scheduling Queue

Figure 2: Proposed Algorithm

Initial Population: In this proposed algorithm, initial parent population is generated randomly. It generates the population size number of chromosomes which defines length of task set. Each chromosome defines solution schedule of the task set in which each gene defines the assigned virtual machine.

For example, we have 8 tasks and 3 resources then sample encoding representation is as follows:



Chromosome: 1 3 2 1 2 3 1 3

Fitness Evaluation

This step evaluates the fitness values of the each population in the population list based on the fitness function which is defined in the system model.

Assign rank and crowding distance

This step assigns the rank to the individuals based on the domination level of the fitness functions. The individual chromosome is compared with other individuals and counts the number of dominances. If the count is zero it dominates all the individuals, then assign the rank 1 and put in to the front. Likewise, repeat the same process to assign the rank for the remaining individuals after discarding the front individual. The crowing distance is calculated from the objective functions.

Non-domination sort

It is used to select the best N individuals from the mixed chromosomes. After creates the child population it combined with parent population using merge function. The given set of individuals is sorted based on the rank value. We applied the Quick sort algorithm to sort the population.

Child Population Creation

From the mixed number of individuals, the child population is created by applying the Selection, Crossover, Mutation process.

Selection

In our algorithm Rank Selection is used to select the N individuals from the mixed set. We are selecting the individual which has less rank. If both the individuals have same rank when consideration, the individual which has high crowding distance [11].

Cross over

In our algorithm, single point cross over operator is applied to the selected two individual chromosomes with the cross over probability 0.9. Cross over point is selected randomly with the range of chromosome length.

Mutation

The mutation operator randomly picks up the one gene from the chromosome and inverts it to new picked value. The mutation probability is (1/task number).

Stopping criteria

In our algorithm stopping condition is described by maximum number of generations which is described as 200 generations.

6. EXPERIMENTAL SETUP & RESULTS

The CloudSim Simulation toolkit [14] is used to evaluate our proposed multi objective genetic algorithm based task scheduling in hybrid cloud environment. CloudSim is a toolkit that used or modelling and simulating the large scale infrastructure and services for cloud computing environment. It provides basic classes for describing the physical computing data centres, self contained clouds, service brokers, service provisioning, allocation policies, users and applications for managing the various parts of the cloud system. It also provides the facility to simulate the hybrid cloud environment that interconnects the resources from the private and public cloud. Parameters used in the proposed algorithm are described in the table Table 1.

Table 1
Proposed Algorithm Parameters

Parameters	*Value*
Population Size	100
Generations	200
Cross over Operator	Single Point Random
Crossover Probability	0.9
Mutation Operator	Random
Mutation Probability	1/task number
Selection Operator	Rank selection

For performance evaluation of the framework (GAF), it has been compared with P-Aware [18], DAS [19] and OS [20]. The performance parameters are allocation efficiency, execution time and failure rate under overhead.

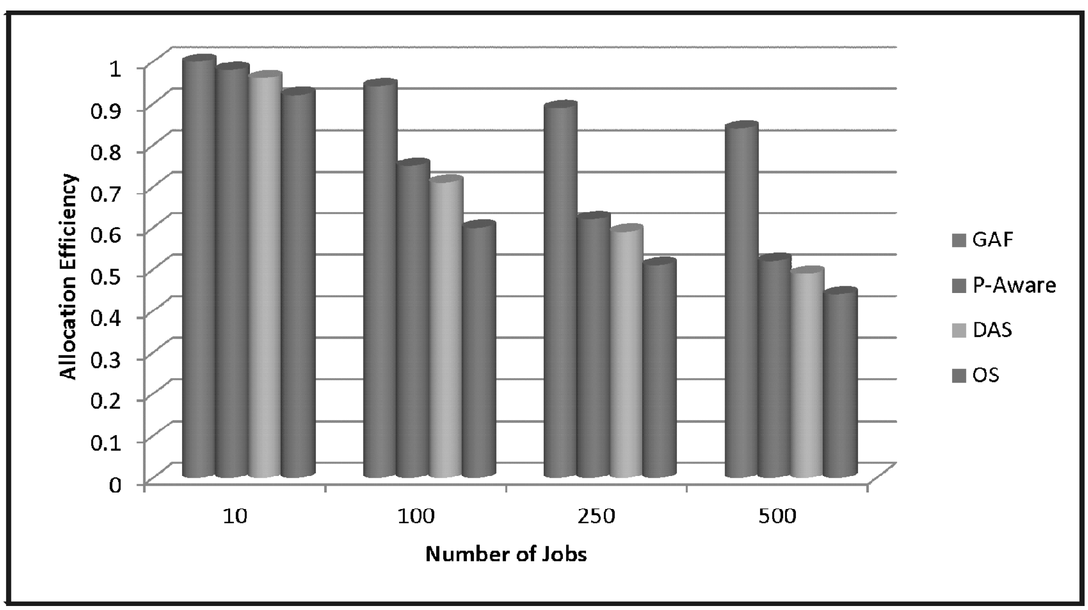


Figure 3: Allocation Efficiency

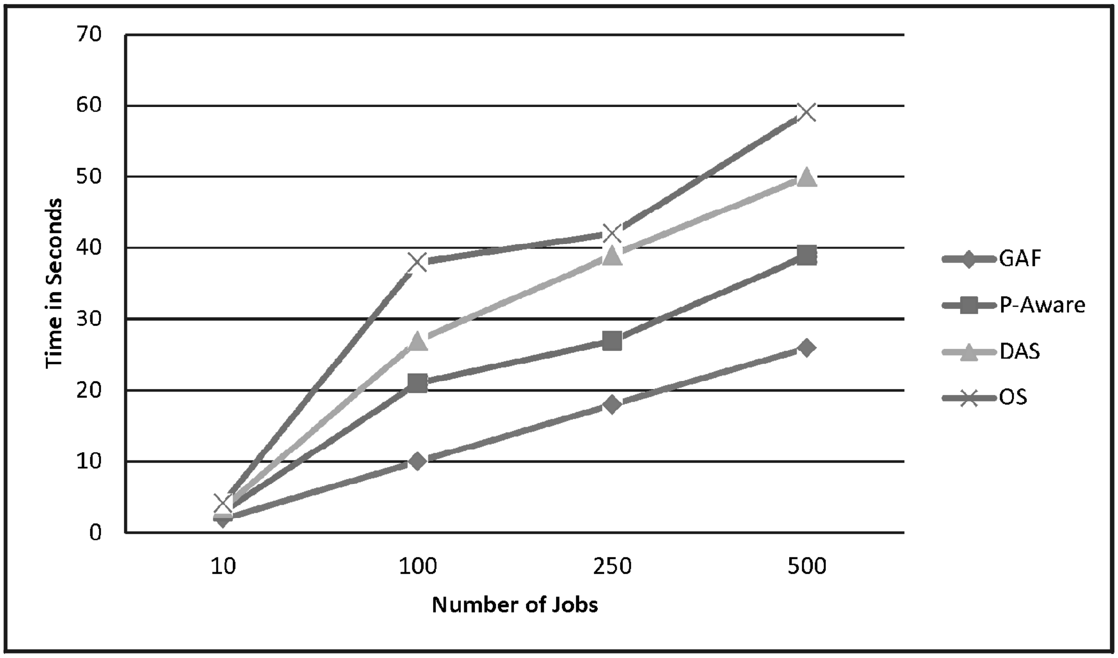


Figure 4: Execution Time of Jobs

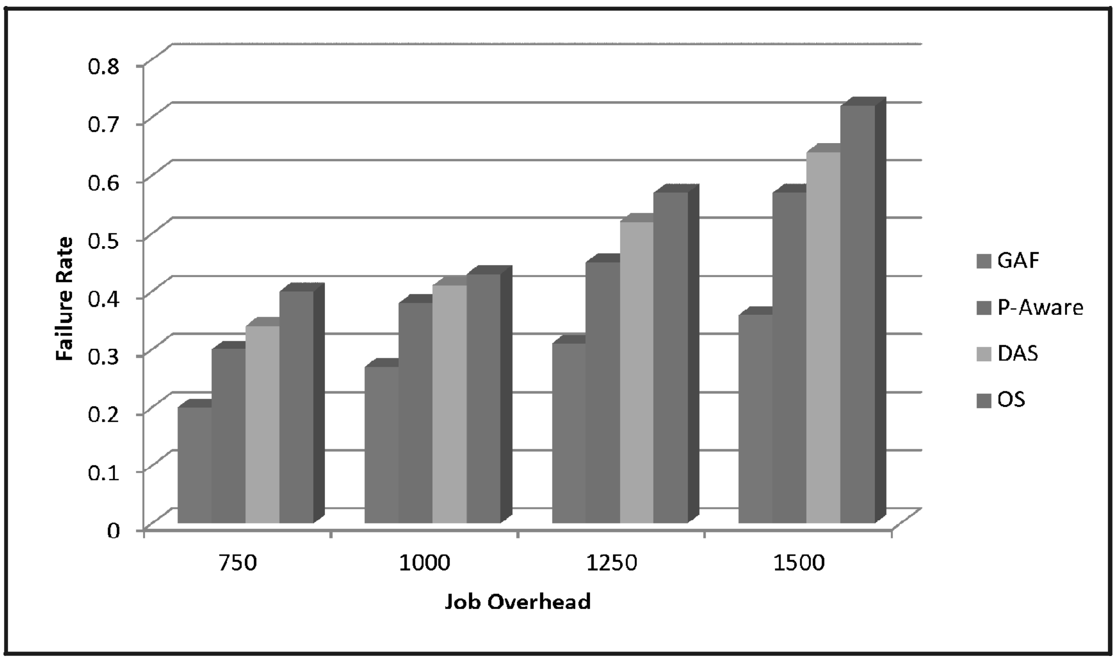


Figure 5: Failure Rate under Overhead

From the above performance measures, it is evident that the proposed framework outperforms the existing systems in all dimensions.

7. CONCLUSION

In cloud computing, a set of task set allocated and utilized the virtual resources dynamically is one of the important issue. In this paper, we proposed efficient resource allocation policy that performs efficient scheduling of jobs on hybrid clouds. In the proposed framework, we applied Genetic Algorithm to compute the multi objective solution which minimizes the execution cost and make span. When GA reaches the maximum number f generations, the final optimal schedule is obtained. From the performance evaluation of the system it is clear that the system provides promising scheduling strategy over the existing systems. For the future extension of the framework, we plan to modify the framework which accepts the dynamic resource pooling so as to avoid the failure rate of the framework.

REFERENCES

[1] Karger D, Stein C, Wein J. Scheduling Algorithms. Algorithms and Theory of Computation Handbook: special topics and techniques. Chapman & Hall/CRC; 2010.

[2] Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. A Genetic Algorithm (GA) based load balancing strategy for cloud computing. Proc Technol 2013;10:340–7.

[3] Ajiro Y, Tanaka A. Improving packing algorithms for server consolidation. In: Int C conf; 2007.

[4] Ge Y, Wei G. GA-based task scheduler for the cloud computing systems. In: Proc int conf web inf syst min, vol. 2; 2010. p. 181–6.

[5] Gu J, Hu J, Zhao T, Sun G. A new resource scheduling strategy based on genetic algorithm in cloud computing environment. J Comput 2012;7:42–52.

[6] Sellami K, Ahmed-Nacer M, Tiako PF, Chelouah R. Immune genetic algorithm for scheduling service workflows with Qos constraints in cloud computing. South African J Ind Eng 2013;24:68–82.

[7] Wu G, Maolin T, Tian Y-C, Li W. Energy-efficient virtual machine placement in data centers by genetic algorithm. In: Vmslv A, editor. Neural inf process. Springer; 2012. p. 315–23.

[8] Tao F, Feng Y, Zhang L, Liao TW. CLPS-GA: a case library and Pareto solution-based hybrid genetic algorithm for energyaware cloud service scheduling. Appl Soft Comput J 2014;19:264–79.

[9] Wang T, Liu Z, Chen Y, Xu Y, Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. In: IEEE 12th int conf dependable auton secur comput; 2014. p. 146–52.

[10] Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 2009, 25(6), 599–616.

[11] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Journal of Evolutionary Computation*, vol. 2, no. 3, 1994.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, April 2002.

[13] S. H. Jang, T. Y. Kim, J. K. Kim, and J. S. Lee, "The study of genetic algorithm-based task scheduling for cloud computing," International Journal of Control and Automation, vol. 5, pp. 157-162, 2012.

[14] R. N. Calheiros, R. Ranjan, C. A. De Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," arXiv preprint arXiv:0903.2525, 2009.

[15] D. M.Abdelkader, F.Omara," Dynamic task scheduling algorithm with load balancing for heterogeneous computing," system Egyptian Informatics Journal, Vol.13, PP.135–145, 2012.

[16] Giuseppe Portaluri, Stefano Giordano, Bernab´e Dorronsoro, "A Power Efficient Genetic Algorithm for Resource Allocation in Cloud Computing Data Centers" IEEE 3rd International Conference on Cloud Networking (CloudNet), PP.58-63, 2014.

[17] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., 2009.

[18] Hang Zhou, Qing Li, Weiqin Tong, Samina Kausar, Hai Zhu, "P-Aware: a proportional multi-resource scheduling strategy in cloud data center", Cluster Computing, Volume 19, issue 3, 2016, pp 1089-1103.

[19] Elias De Coninck, Tim Verbelen, Bert Vankeirsbilck, Steven Bohez, Pieter Simoens, Bart Dhoedt, "Dynamic Auto-scaling and Scheduling of Deadline Constrained Service Workloads on IaaS Clouds", Journal of Systems and Software, Volume 118, August 2016, pp 101–114.

[20] Li Chunlin, Li LaYuan, "Optimal scheduling across public and private clouds in complex hybrid cloud environment", Information Systems Frontiers, 2015, pp 1-12.