Journal of Theoretical and Applied Information Technology
20th March 2016. Vol.85. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A DYNAMIC APPROACH TO TASK SCHEDULING IN CLOUD COMPUTING USING GENETIC ALGORITHM

R. DURGA LAKSHMI [1], N SRINIVASU[2*]

R.Durga Lakshmi[1], Research Scholar, Department of Computer Science . K L University, Vijayawada, India
N Srinivasu[2], Professor, Department of Computer Science and Engineering, K L University, Vijayawada, India
E-Mail:durgalakshmikluniversity@gmail.com

ABSTRACT

Cloud computing is one of device technology trends in the future since it combines the advantages of both device computing and cloud, Recent years have seen the massive migration of enterprise applications to the cloud. Cloud computing used in business organizations and educational institutions. One of the challenges posed by cloud applications is Quality-of-Service (QoS) management, which is the problem of allocating resources to the application to guarantee a service level along dimensions such as performance, availability and reliability. To improve the QoS in a system one must need to reduce the waiting time of the system. Genetic Algorithm (GA) is a heuristic search technique which produces the optimal solution of the tasks. This work produces one scheduling algorithm based on GA to optimize the waiting time of overall system. The cloud environment is divided into two parts mainly, one is Cloud User (CU) and another is Cloud Service Provider (CSP). CU sends service requests to the CSP and all the requests are stored in a Request Queue (RQ) inside CSP which directly communicates with GA Module Queue Sequencer (GAQS). GAQS perform background operation, like daemon, with extreme dedication and selects the best sequence of jobs to be executed which minimize the Waiting time (WT) of the tasks using Round Robin (RR) scheduling Algorithm and store them into Buffer Queue (BQ). Then the jobs must be scheduled by the Job Scheduler (JS) and select the particular resource from resource pool (RP) which it needs for execution.

***Keywords**: Genetic Algorithm, Cloud computing, Quality of Service, Cloud User, Cloud Service Provider, Request Queue, GA Module Queue Sequencer, Buffer Queue, Waiting Time, Round Robin Scheduling Algorithm, , Resource Pool.*

1. INTRODUCTION

Cloud computing, often referred to as simply "the cloud," is the delivery of on-demand computing resources, everything from applications to data centers, over the Internet on a pay-for-use basis. Cloud computing environment is highly dynamic; the system load and computing resource utilization exhibit a rapidly changing characteristic over time. Therefore Cloud service provider normally over-position computing resources to accommodate the peak load and computing resources are typically left under-utilize in nonpeak time. Cloud environment allows users to use applications without installation and access their personal files at any computer with Internet access. End users access cloud based applications through a web browser or a light weight desktop

*Corresponding author: Professor N. Srinivasu

or mobile app while the business software and data are stored inside CSP at a remote location. Cloud application providers strive to give the better service and performance than if the software programs were installed locally on end-user machines. Cloud environment is used in lot of fields like in IT industries, educational institute as well as in other industries. In this paper we have proposed Cloud Service Provider, figure 1, which includes mainly three parts- GA Module Queue Sequencer, Job Scheduler (JS) and Resource Pool (RP). All service requests which are coming from Cloud Users domain are stored in RQ which is in GAQS. Now the requested processes must communicate with GAQS processor (GAP) and the processor finds out the appropriate sequence of tasks which reduce the waiting time of the tasks. GAQS processor then communicate directly with JS which schedules the tasks using Round Robin scheduling algorithm and communicate with RP and tries to assign each of these jobs as per their requirement to the resources. But the main problem here is that to find out the best sequence of the tasks from all possible sequences of tasks and JS schedules those tasks and optimize total Waiting time of those jobs.

The jobs assignment task is done by JS. So JS must need to assign the task such a way that assignments of the jobs to the resources must be fruitful as per as CU requests and the total execution time must be optimal of the whole operations. In next two sections discuss about our proposed model of CSP and one Genetic based scheduling an algorithm which assigns the task to the resource as per the CU's demand and also to optimize the total waiting time of those tasks.

1.1 Scheduling Algorithms

There has been various types of scheduling algorithm exist in distributed computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm.
By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud
environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is
suitable example of On-line mode heuristic scheduling
algorithm.

a. First Come First Serve Algorithm:
Job in the queue which come first is served. This algorithm is simple and fast.

b. Round Robin algorithm:
In the round robin scheduling, processes are dispatched in a
FIFO manner but are given a limited amount of CPU time
called a time-slice or a quantum. If a process does not complete before its CPU-time expires, the CPU is pre-empted and given to the next process waiting in a queue. The preempted process is then placed at the back of the ready list.

c. Min–Min algorithm:
This algorithm chooses small tasks to be executed firstly,
which in turn large task delays for long time.

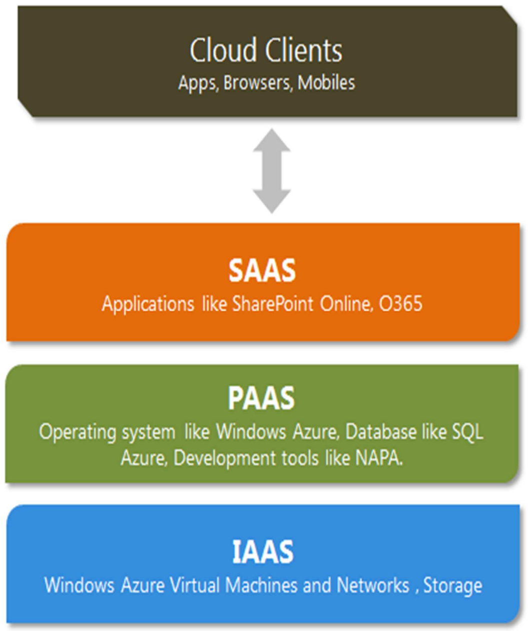
d. Max – Min algorithm:
This algorithm chooses large tasks to be executed firstly,
which in turn small task delays for long time.

e. Most fit task scheduling algorithm:
In this algorithm task which fit best in queue are executed
first. This algorithm has high failure ratio.

f. Priority scheduling algorithm:
The basic idea is straightforward: each process is assigned a
priority, and priority is allowed to run. Equal-Priority processes are scheduled in FCFS order. The shortest-Job-First (SJF) algorithm is a special case of general priority scheduling algorithm. An SJF algorithm is simply a priority algorithm where the priority is the inverse of the (predicted) next CPU burst. That is, the longer the CPU burst, the lower the priority and vice versa. Priority can be defined either internally or externally. Internally defined priorities use some measurable quantities or qualities to compute priority of a process.

1.2 Cloud Computing

Cloud computing, also known as 'on-demand computing', is a kind of Internet-based computing, where shared resources, data and information are provided to computers and other devices on-demand


Cloud Clients
Apps, Browsers, Mobiles
SAAS
Applications like SharePoint Online, O365
PAAS
Operating system like Windows Azure, Database like SQL Azure, Development tools like NAPA.
IAAS
Windows Azure Virtual Machines and Networks , Storage

Cloud computing service models

Cloud computing promises several attractive benefits for businesses and end users. Three of the main benefits of cloud computing include:

a. Infrastructure As A Service (IaaS)

In the most basic cloudservice model, providers of IaaS offer core IT services, for example, computer (physical or virtual machines) networks, compute, security, operating systems, middleware devices, load balancers and block and file or object storage. IaaS clouds regularly provide extra resources such as a virtualmachine disk image library, IP addresses, firewalls, virtual local area networks (VLANs) and other software bundles. IaaScloud providers provide these resources ondemand from their big pools installed in data centres. For widearea network connectivity, businesses can use either the Internet or carrier clouds (dedicated virtual private networks) that IaaS clouds offer. For the deployment of their applications, users install operatingsystem instances and their application software on the cloud infrastructure. The cloud user also patches and maintains the operating systems and the application software. Cloud providers normally invoice IaaS services on a usagebased computing where the cost reflects the size of resources allocated and consumed.

b. Platform As A Service (PaaS)

In the PaaS models, cloud providers provide a computing platform, usually including operating systems, programming language running environment, database and web servers. Application developers can develop and run their software solutions on a cloud platform minus the cost and complexity of purchasing and managing the underlying hardware and software layers.As per some PaaS offers like Microsoft Azure and Google App Engine, the core computer and storage resources scale by design to meet application demand so that the cloud user does not need to allot resources manually. This has also been planned by an architecture targeting to facilitate realtime in cloud environments. Media encoding as an application can also be provided via PaaS.

PaaS platforms are not very common because service providers often cannot offer customers the control and variety that they need for their applications. Furthermore, vendor lockin is another concern. Once the application starts to use any proprietary tools or interfaces that a PaaS provider makes accessible, migration to another provider may become challenging.

c. Software As A Service (SaaS)

In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud service providers manage the infrastructure and platforms that run the applications. SaaS is on occasions cited to as “ondemand software” and is typically rated on payperuse or on a subscription fee basis. In the SaaS model, cloud providers manage the application software in the cloud and cloud users access the software via cloud clients. The cloud infrastructure and platform where the application runs is transparent to the cloud users. The need to install and run the application on the cloud user’s own computers is not needed, which simplifies maintenance and support. Cloudbased applications are unlike other applications in their scalability, which is achieved by duplicating tasks onto several virtual machines at runtime to meet varying demand with load balancers distributing the work. The user sees only a single access point. Cloud applications can be multitenanted to accommodate a huge number of cloud users having any machine serving more than one cloud user organisation.The pricing model for SaaS applications is usually a monthly or yearly flat fee per user, consequently price is scalable and flexible if users are added or removed at any point.Supporters claim SaaS lets a business the potential to lessen IT operational costs by subcontracting hardware and software maintenance and support to the cloud provider. This facilitates the business to reapportion IT operations expenses away from hardware/software and staff costs. Besides when

applications are hosted centrally, updates can be done without the need for users' intervention.
One shortcoming of SaaS is that the users' data are stored on the cloud provider's server which could be subject to unauthorised access to the data. Hence, users are progressively implementing intelligent thirdparty key management systems to protect their data.

2. LITERATURE REVIEW

Various modifications to task scheduling in cloud computing and genetic lgorithm have been proposed by several authors. These modifications can be classified as follows

In 2003,H.xiaoshan et al [3] suggested a QoS Guided Min-Min heuristic [Batch mode heuristic algorithm] was introduced in that some task require higher network bandwidth to exchange a large amount of data among processors, whereas some can be satisfied with the lower network bandwidth. In this algorithm the matching of the QoS request and services between the tasks and hosts based on conventional Min-Min. Firstly each task with the high QoS request in the Meta task, the algorithm finds the earliest completion time and the host that obtains it, in the entire QoS Qualified host. Secondly find the task with the minimum earliest completion time and assigns the task to the host that give the earliest completion time to task. In this algorithm they have addressed only one-dimension QoS issue, because they worked only bandwidth constraint

In 2006 F.Dong et al.. [4] proposed a QoS priority grouping
algorithm which considers deadline and acceptation rate of the task and the makespan as main factor of task scheduling in whole system. It achieves better acceptance rate and
completion time for the submitted task then Min-Min and QoS Guided Min-Min.

In 2008, C.Hsu et al [5] carried out two optimization schemes MOR (Makespan Optimization Rescheduling) and ROR (Resource Optimization Rescheduling). MOR focus on improving the makespan to pull off the better performance and in ROR focus on the redispatch tasks from the machine with the minimum number of tasks to other machine, which is helpful to reduce the resource need. Both this technique achieves low complexity, high effectiveness, good performance than QoS Guided scheduling algorithm and Min-Min algorithm.

In 2008, M.Singh et al [6] proposed a QoS based predictive Max-Min, Min-Min switcher algorithm. In this algorithm, scheduling of the next job is based on appropriate selection among QoS based min-min or QoS max-min algorithm. The effect on the execution time grid jobs has been reduced due to non-dedicated resources. It normally uses the history information about the execution jobs to predict the performance of non-dedicated resources. This algorithm
merges the efficiency of max-min along with min-min and also considers both QoS and non-dedicated property of grid
resources.

In 2009, S.Parsa et al [7] introduced a new task scheduling algorithm called RASA which has the advantage of both Min-Min and Max-Min algorithm. In this first estimate the completion time of the tasks on each resource and then applied both the algorithm. RASA use the Min-Min strategy to execute the small task first then long task and then applied Max-Min to avoid the delays in the execution of large task and support concurrency in the execution of the large and small tasks. It achieves the lower Makespan with good QoS

In2010, Mrs.S.Selvarani et al [8] introduced an improved costbased costbased scheduling algorithm for making efficient mapping of tasks to available resources in cloud. The improvisation of traditional activity based costing is proposed by new task scheduling strategy for cloud environment where there may be no relation between the overhead application base and the way that different tasks cause overhead cost of resources in cloud. This scheduling algorithm divides all user tasks depending on priority of each task into three different lists. This scheduling algorithm measures both resource cost and computation performance, it also Improves the computation/communication ratio.

In 2011, C.Zhao et al [9] proposed a Berger Model in Cloud computing in that algorithm scheduling process establish dual fairness constraint. First constraint is to classify user task by QoS preferences, and establish the general expectation function in accordance with the classification of tasks to restrain the fairness of the resources in the selection process. Second constraint is to define resource fairness justice function to judge the fairness of the resources allocation. According to constraint, the algorithm always assigns tasks on the optimal resources in order to satisfy the QoS requirement of user and it avoid to consider a long task for execution. Experiment result of this algorithm shows effective execution of the user tasks and manifest better performance.

In 2013, X.Wu et al [10] introduce a task scheduling algorithm based on QoS-driven in cloud computing (TS-QoS). In this TS-QoS algorithm compute the priority of the task according to the special attributes of the tasks, and then sort tasks based on priority. Then the algorithm calculate the completion time of each task on different services, and schedule each task onto a service which can complete the task as soon as possible according to the sorted task queue. But in this process priority can change dynamically an increase continuously this can help to solve the ―starvation‖ problem and follow FCFS principle. Experimental result achieves well performance and load balancing by QoS driving form both priority and completion time.

In 1995 Kennedy and Eberhart [11] Particle Swarm introduced The PSO algorithm Optimization (PSO) as a meta-heuristics method is a self-adaptive global search based optimization technique it is alike to other population-based algorithms like Genetic algorithms (GA) but, there is no direct recombination of individuals of the population The PSO algorithm focuses on minimizing the total cost of computation of an application workflow. As a measure of performance, Authors used cost for complete execution of application as a metric. The objective is to minimize the total cost of execution of application workflows on Cloud computing environments. Results show that PSO based task-resource mapping can achieve at least three times cost savings as compared to Best Resource Selection (BRS) based mapping for our application workflow. In addition, PSO balances the load on compute resources by distributing tasks to available resources.

In 2013, Dr. M. Dakshayini, Dr. H. S. Guruprasad [33] introduced this algorithm. The main idea of the Min-Min algorithm is as quickly as possible to dispatch each task to virtual machines as resources which can complete the task in the shortest possible time. Min-Min algorithm will execute short jobs in parallel and the long jobs will follow the short jobs. The shortcoming of this algorithm is the short jobs scheduled first, until the machines are leisure to schedule and execute long jobs. Min-min can cause both the whole batch jobs executed time get longer and unbalanced load. Even long jobs cannot be executed. Compared with the traditional Minmin algorithm, improved algorithm adds the three constraints (quality of service, the dynamic priority model and the cost of service) strategy which can change this condition. The experimental results of improved Min-Min algorithm show it can increase resource utilization rate, long tasks can execute at reasonable time and meet users' requirements.

3. PROPOSED MODEL

Before starting to discuss about Cloud queueing model first we discuss about the Genetic algorithm and then site our proposed scheduling algorithm based on Genetic algorithm.

Genetic algorithms (GA) were first proposed by the John Holland in the 1960s The GA is a heuristic search technique that simulates the processes of natural selection and evolution. Genetic algorithm (GA) is a promising global optimization technique.

It works by emulating the natural process of evolution as a means of progressing towards the optimal solution. A genetic algorithm has the capability to find out the optimal job sequence which is to be allocated to the processor.
General Algorithm perform its general operations using the following steps-

A. Select the fixed size chromosomes from the from the population set.
B. Perform any one type encoding operation on the chromosomes of the chromosome sets.
C. Select the best two chromosomes from the chromosome set using their fitness value.
D. Perform the crossover between two chromosomes and get two different offspring.
E. Perform the mutation operation on those offspring just interchanging the bit positions.
F. Continue the steps A to B until get the best solution of the population.
G. Finally perform the elitism operation of the chromosomes means store the best chromosomes in to the system for future use.

Now we discuss our algorithm based on Genetic Algorithm step by step-.

a) Cloud users send the request to the Cloud service provider for the resource or resources.

b) CSP stores the request initially into request queue of GAQS.
c) GAQS processor then select set of tasks from the RQ and rearrange them until it gets the best arrangement of the tasks.
d) GAQS store the final task set into the buffer queue.
e) Then job scheduler execute the tasks one by one using round robin scheduling algorithm and select the resource or resources to the cloud users.

Figure 1 describes the architecture of the GA guided scheduling mechanism and the execution steps also shown by the numbering.

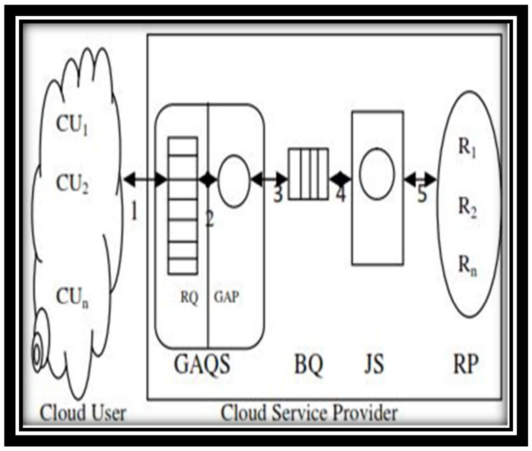


Figure 1: *Architecture of Cloud queuing model using Genetic Algorithm*

Now we discuss the details of GAQS operation step by step-

1)Request initially comes from the CU for the resources andstore into the resource pool.

2) Now GA processor execute the following steps untilthebest sequence of tasks are produced-

a)Select the suitable number of tasks from the request pool using their fitness value (means the tasks those who are ready to execute has the higher fitness value compare to the no ready tasks) and create one chromosome.

b)Now perform the mutation operation on the tasks, just to interchange the positions of them and find out the waiting time of that sequence using round robin scheduling algorithm individually.

c)Choose the best sequence of tasks from the task sets whichhave least waiting time, this step is known as elitism.

3)Finally the tasks that produced by the GAQS must be stored in the buffer queue and latter the JS execute the operation using those tasks.

In Figure 2 diagrammatically shown the basic architecture of GAQS which is already discuss in previous part

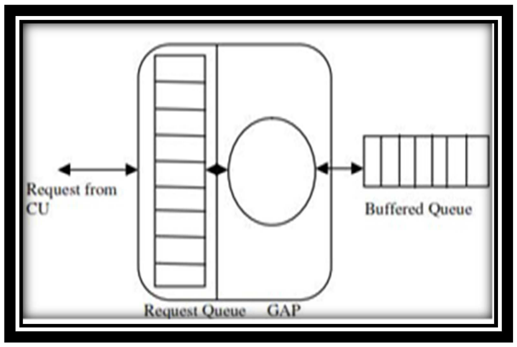


Figure 2: *Diagram of GA Module Queue Sequencer*

In the next section we discuss our propose algorithm using one example.

4. PERFORMANCE ANALYSIS

In this section we elaborately discuss our propose algorithm using one example. Suppose Cloud users sends n number request for the resources and those resources initially store into the request queue in GAQS like

$P_1, P_{2...} P_n$ as the request come from CU. Now GA processor of GAQS select the tasks from the RP those are ready to execute. Suppose first time tasks P_1, P_2, P_3 are ready to execute and their burst times are 20, 26 and 12

respectively. Now GAP executes all possible sequences of task one by one using Round Robin scheduling. If there are n number of tasks are ready to execute, so the number of possible ways are n!. Here three tasks are ready to execute, so the possible way to execute of the tasks into JS are 3! Or 6 way. We discuss all of them one by one. Here we mention the time quantum of the tasks is 10 for Round Robin scheduling operation.

Table 1. Process table case i

The table 1 is depicting here a sample snapshot of three processes whose CPU burst are as follows 20, 26 and 12. Here we have mentioned the burst table according to the present scenario of system state.

P1	*P2*	*P3*	*P1*	*P2*	*P3*	*P2*
0	*10*	*20*	*30*	*40*	*50*	*52* *58*

Table 1.1. Burst table for the above processes

The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

Process name	**Calculation of waiting time**	**Output**
P_1	(0+(30-10))	20
P_2	(10+(40-20))	30
P_3	(20+50-30))	40

Table 1.2. Waiting time of several processes

Here the average Waiting time = (20+30+40)/3 = 90/3 = 30 The second option is being explained in the next table of process pool. Here a slight variation can be seen using different notion of process shuffling inside CSP.

Process	**Burst time**
P_2	26
P_1	20
P_3	12

Table 2. Process table case ii

Table 2 in the above is defining the process pool with several Burst Time 26, 20 and 12. Here we have mentioned the burst table according to the present scenario of system state.

P2	P1	P3	P2	P1	P3	P2
0	10	20	30	40	50	52 58

Table 2.1. Burst table for the above processes

The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

Process name	**Calculation of waiting time**	**Output**
P_1	(10+(40-20))	30
P_2	(0+(30-10)+(52-40)	32
P_3	(20+(50-30))	40

Table 2.2. Waiting time of several processes

Here the average Waiting time = (30+32+40)/3 = 102/3 = 34. The third option is being explained in the next table of process pool. Here a slight variation can be seen using different notion of process shuffling inside CSP.

Process	**Burst Time**
P_1	20
P_3	12
P_2	26

Table 3. Process table case iii

Table 3 in the above is defining the process pool with several Burst Time 20, 12 and 26. Here we have mentioned the burst table according to the present scenario of system state.

P1	P3	P2	P1	P3	P2	P2
0	10	20	30	40	42	52 58

Process	**Burst Time**
P_1	20
P_2	26
P_3	12

Table 3.1 Burst table for the above processes

The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

Process name	Calculation of waiting time	Output
P_1	(20+(42-30))	32
P_2	(0+(30-10)+(52-40))	32
P_3	10+(40-20)	30

Table 3.2 Waiting time for several processes

Here the average Waiting time = (20+42+30)/3 = 92/3 = 30.6

The fourth option is being explained in the next table of process pool. Here a slight variation can be seen using different notion of process shuffling inside CSP.

Process	Burst Time
P_2	26
P_3	12
P_1	20

Table 4. Process table case iv

Table 4 in the above is defining the process pool with several Burst Time 26, 12 and 20. Here we have mentioned the burst table according to the present scenario of system state.

P2	P3	P1	P2	P3	P1	P2
0	10	20	30	40	42	52 58

Table 4.1 Burst Table for the above processes

The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

Process name	Calculation of waiting time	Output
P_1	(0+(30-10))	20
P_2	(20+(42-30)+(52-42)	42
P_3	10+(40-20)	30

Table 4.2 Waiting time for several processes

Here the average Waiting time = (32+32+30)/3 = 94/3 = 31.3.

The fifth option is being explained in the next table of process pool. Here a slight variation can be seen using different notion of process shuffling inside CSP

Process	Burst Time
P_3	12
P_2	26
P_1	20

Table 5. Process table case v

Table 5 in the above is defining the process pool with several Burst Time 12, 26 and 20. Here we have mentioned the burst table according to the present scenario of system state.

Process name	Calculation of waiting time	Output
P_1	(20+(42-30))	32
P_2	10+((32-20)(52-42))	32
P_3	0+(32-10)	22

Table 5.1 Burst table for the above processes

The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

P3	P2	P1	P3	P2	P1	P2
0	10	20	30	32	42	52 58

Table 5.2 Waiting time for several processes

Here the average Waiting time = (32+32+22)/3 = 86/3 = 28.6.

The sixth option is being explained in the next table of process pool. Here a slight variation can be seen using different notion of process shuffling inside CSP

Process	Burst Time
P_3	12
P_1	20
P_2	26

Table 6. Process table case vi

Table 6 in the above is defining the process pool with several Burst Time 12, 20 and 26. Here we have mentioned the burst table according to the present scenario of system state.

P3	P1	P2	P3	P1	P2	P2
0	10	20	30	32	42	52 58

Table 6.1 Burst Table for the above processes

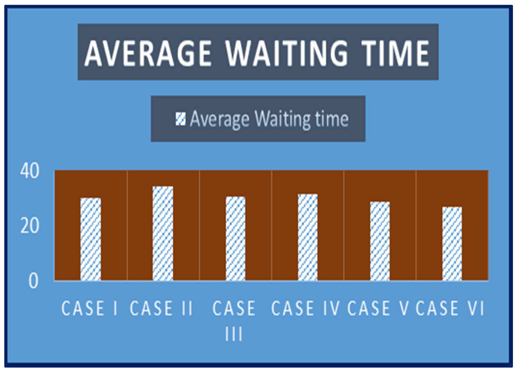
The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

Process name	Calculation of waiting time	Output
P1	(10+(32-20))	22
P2	(20+(42-30)(58-52)	38
P3	0+(30-10)	20

Table 6.2 Waiting time for several processes

Here the average Waiting time = (22+38+20)/3 = 80/3 = 26.6

Here the average waiting time of all possible arrangement of three processes those are ready to execute. From that we see that if we arrange the processes using sixth table and then execute them the average waiting time of the tasks must be optimal. So this particular sequence must be initially store into the buffer queue and JS execute the tasks using the sequence which is stored into BQ and select the resources as CUs demands. So, using GA we reduce the waiting time of the overall system.



Graph 1 Compression of Average Waiting Time in different cases.

5. COMPARISION OF AVERAGE WAITING TIME BETWEEN ADDR ALGORITHM AND NEW PRAPOSED GACQM ALGORITHM

i. ADRR ALGORITHM

An Augmented Dynamic Round Robin (ADRR) CPU scheduling algorithm works similar to Round Robin (RR) with an improvement. ADRR picks the first process from the ready queue and allocate the CPU to it for a time interval of up to 1 time quantum. After completion of process's time quantum, it checks the remaining CPU burst time of the currently running process. If the remaining CPU burst time of the currently running process is less than equal half of the time quantum, the CPU again allocated to the currently running process for remaining CPU burst time. In this case this process will finish execution and it will be removed from the ready queue. The scheduler then proceeds to the next process in the ready queue. Otherwise, if the remaining CPU burst time of the currently running process is longer than 1/2 time quantum, the process will be put at the tail of the ready queue. The CPU scheduler will then select the next process in the ready queue.

If the above processprs example is applied to this ADRR algorithm then take

Process	Burst Time
P_1	20
P_2	26
P_3	12

Table 7. Process table

The table 1 is depicting here a sample snapshot of three processes whose CPU burst are as follows 20, 26 and 12. Here we have mentioned the burst table according to the present scenario of system state.

P1	P2	P3	P3	P1	P2	P2
0	10	20	30	32	42	52 58

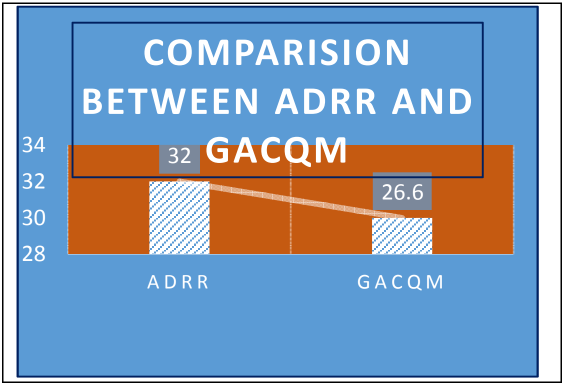
Table 7.1. Burst table for the above processes

The estimated waiting time of the several processes and the average waiting time of the cloud service provider system will be as follows

Process name	Calculation of waiting time	Output
P_1	(0+(32-10))	30
P_2	(10+(42-20)+(52-42)	42
P_3	(30+(32-30))	32

Table 7.2. Waiting time of several processes

Here the average Waiting time = (22+42+32)/3 = 96/3 = 32



Graph 2 *Compression of Average Waiting Time Between ADRR and GACQM Algorithms.*

By the above chart we observed that the GACQM algorithm is giving less average waiting time than ADDR algorithm.so the new proposed algorithm is better than ADDR algorithm.

6. ADVANTAGES, DISADVANTAGES, ISSUES ASSUMPTIONS, ARISING RELATED TO GACQM ALGORITHM

i. The algorithm must operate on multi-resource based CSP system. It must not operate on accessory system because it creates an overhead of the whole system.

ii. The GA processor must operate on the tasks those are ready in RQ to execute. The tasks those are recently come to RQ or the tasks those are come after the execution start by the GA processor must not execute at that time.

iii. The algorithm must be executed in background. So it must not hamper the execution of the selected tasks on the other processors.

iv. First time it requires some time to find out the sequence of the tasks those are executed in other processors. But from next step it must not take any extra time because GAP find out the exact sequence of the task in the background and other processor must execute in foreground.

v. This algorithm reduces the overall average waiting time of the system.

vi. This algorithm increases the throughput of the system.

vii. The algorithm is not applicable in single resource based CSP system because the throughput of the system must be degraded a lot.

viii. The algorithm is not applicable in online application, means when any new process arrives at RQ the GAP must not use it in current operation which is already started. So the algorithm works as static mode.

7. CONCLUSION

In this paper GA based scheduling algorithm is proposed. It gives better performance than conventional round robin algorithm and augmented dynamic round robin algorithm also, it gives minimum average waiting time. Genetic algorithm is a heuristic search algorithm and it always find out the solution which takes minimum time to execute or find optimal solution from the set of possible solutions. This algorithm must increase the throughput of the system. Here the tasks those are ready to execute are operate in all possible way and find the best sequence of the tasks which average waiting time must be optical one. So the waiting time of the system must be optimum. This algorithm can be implemented to improve the performance in the systems in which RR is a preferable choice.

8. FUTURE WORK

In future we are trying to reduce the problems of that algorithm and try to create another algorithm which also produces an optimal solution. In future we also create a simulator of that algorithm and implement it in our environment.

REFERENCES

[1] Kaiqi Xiong, Harry Perros "Service Performance and Analysis in Cloud Computing" 978-0-7695-3708-5/09 $25.00 © 2009 IEEE page- 693-700

[2] Borja Sotomayor, Rubén S. Montero and Ignacio M. Llorente, "Virtual Infrastructure Management in Private and Hybrid Clouds" Ian Foster 1089-7801/09/$26.00 © 2009 IEEE

[3] XiaoShan He,Xianhe Sun and Gergor von Laszewski. QoS guided Min-Min heuristic for grid task scheduling‖. Journal of Computer Science and Technology, 2003, 18(4), p.442-451.

[4] Dong. F, Luo. J, Gao. L and Ge. L, "A Grid Task Scheduling Algorithm Based on QoS Priority Grouping," In the Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06), IEEE, 2006.

[5] Ching-Hsien Hsu,Zhan. J.,Wai-Chi Fang, et al. ―Towards improving QoS-guided scheduling in grid‖ Third ChinaGrid Annual Conference(CHINAGRID). Dunhuang, Gansu,China, 2008, p.89-95.

[6] M.Singh and P.K.Suri; ―QPSMax-Min<>Min-Min : A QoS Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid‖, ―International Technology Journal7(8)‖ :p.1176-1181,2008

[7] Saeed Parsa and Reza Entezari-Maleki,‖ RASA: A New Task Scheduling Algorithm in Grid Environment‖ in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009. [8] Mrs. S.Selvarani1, Dr.G.Sudha Sadhasivam,‖ improved cost-based algorithm for task scheduling in Cloud computing ―in IEEE 2010.

[9] Baomin Xu,Chunyan Zhao,Enzhao Hua,Bin Hu. ―Job scheduling algorithm based on Berger Model in Cloud Environment‖ Advance in Engineering Software, 2011, 42(7), p.419-425.

[10] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou. ―A task scheduling algorithm based onQoS-driven in Cloud Computing‖ Information Technology and Quantitative Management(ITQM2013).2013,p.1162-1169 [11] J. Kennedy and R. Eberhart, (1995), Particle swarms optimization In IEEE International Conference on Neural Networks, volume 4, pages 1942–1948.

[12] Pai-Chou Wang, W. Korfhage, "Process Scheduling with Genetic Algorithms", Proceedings of the 7th IEEE Symposium on Parallel and Distributed Processing, Page: 638, ISBN: 0-8186- 7195-5, October 2005, IEEE Computer Society Washington, DC, USA.

[13] T.Gunasekhar, K Thirupathi Rao,et,at.," EBCM: Single Encryption, Multiple Decryptions" International Journal of Applied Engineering Research,vol .9(19), 2014 [14]T.Gunasekhar, K.Thirupathi Rao, et al. "Mitigation of Insider Attacks through MultiCloud."International Journal of Electrical and Computer Engineering (IJECE) 5.1 (2015). [15] G Siva Nageswara Rao,N. Srinivas, "Comparison of Round Robin CPU Scheduling Algorithm with Various Dynamic Time Quantum", International Journal of Applied Engineering Research, ISSN 0973- 4562 Volume 9, Number 18 (2014) pp. 4905-4916.

[16] G Siva Nageswara Rao, A New Proposed Dynamic Dual Processor Based Cpu Scheduling Algorithm, Journal Of Theoretical And Applied Information Technology 20th March 2015. Vol.73 No.2.

[18] GAN Guo-ning, HUANG Ting-lei, GAO Shuai, "Genetic Simulated Annealing Algorithm for task scheduling based on Cloud Computing Environment", 978-1-4244-6837-9/10/$26.00 2010 IEEE.

[19] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, " A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", Procedia Technology, 10 (2013) 340 – 347.

[20] Monika Choudhary, Sateesh Kumar Peddoju, "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment", IJERA, Vol. 2, Issue 3, May-Jun 2012, pp.2564-2568.

[21] Ji Lia;b, Longhua Fenga;b, Shenglong Fan" An Greedy-Based Job Scheduling Algorithm in Cloud Computing" JOURNAL OF SOFTWARE, VOL. 9, NO. 4, APRIL 2014.

[22] Stuti Dav Prashant Maheta "Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment" International Journal of Computer Applications (0975 – 8887) Volume 94 – No 4, May2014.

[23] Lizheng Guo1,2 Shuguang Zhao1, Shigen Shen1, Changyuan Jiang1" Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm" JOURNAL OF NETWORKS, VOL. 7, NO. 3, MARCH 2012.

[24] Jia Yu, Rajkumar Buyya and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids", In 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, Dec. 5-8, 2005

[25] Van den Bossche, R., Vanmechelen, K., Broeckhove, J, "Cost Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads", in 3rd IEEEInternational Conference on Cloud Computing, Miami (July 2010)

[26] Savitha. P, J Geetha Reddy ,"A Review Work On Task Scheduling In Cloud Computing Using Genetic Algorithm",International Journal of Sciences & Technology Research, vol 2, Issue 8, August (2013)

[27] Ankur Bhardwaj " Comparative Study of Scheduling Algorithms in Operating System", International Journal of Computers and Distributed Systems, Vol. No.3, Issue I, April-May 2013.

[28] V. Krishna Reddy, L.S.S.Reddy, "A Survey of Various task Scheduling Algorithms in Cloud Environment", Global Journal Engineering and Applied Sciences (GJEAS),Volume 2.No1,January,2012.

[29] Arash Ghorbannia Delavar,Mahdi Javanmard ,Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi "RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3,June,2012

[30] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.

[31] Dr. M. Dakshayini, Dr. H. S. Guruprasad "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment" International Journal of Computer Applications (0975 – 8887) Volume 32–No.9, October 2011

[32] Shamsollah Ghanbari, Mohamed Othman "A Priority based Job Scheduling Algorithm in Cloud Computing" International Conference on Advances Science and Contemporary Engineering 2012 ICASCE 2012