IJCTA, 9(25), 2016, pp. 323-335 ISSN: 0974-5572
© International Science Press

Optimization of Task Scheduling Using Improved Clonal Selection in K-Level Architecture

A. Kousalya* and R. Radhakrishnan**

ABSTRACT

Cloud computing is the promising technology in currentand future computing platforms. Cloud computing enablesthe user to run their applications in remote data centers and accessing resources that are locally limited becomes easier. Cloud computing deals with many virtualized resources and therefore scheduling plays an important role.Scheduling is an active research area in cloud computing. The aim of task scheduling is to reduce operational costs, reduce queue waiting time, increase resource utilization and to achieve high system throughput.This can be achieved by Load Balancing. Load balancing is an important aspect in cloud where the load has to be balanced for optimal resource utilization.The computing capacity of each node is partitioned in the data centers using the K-Level mechanism. In this paper, improved PSO is implemented in the K-level architecture.Therefore Queue waiting time will be reduced and optimal resource utilization can be achieved.

Indexterms-Cloud Computing, Load-balancing, Scheduling, Virtual Machine, Particle swarm optimization, Improved Clonal Selection

1. INTRODUCTION

With the enormous growth of internet access, cloud computing becomes the hottest emerging fields in the IT industry and society. With infinite number of computers interlinked together, task scheduling becomes the main concern in cloud environment.So large-scale task scheduling occurs more frequently for which the user needs an optimal solution to solve this.Task scheduling is to assign a certain number of tasks to the resource that will reduce the total completion time as possible, which is an NP-complete problem.

Task scheduling is the process of assigning proper resources thereby reducing the waiting time of a user to access the resource which may be a physical resource, software, hardware or may be a platform. Inefficient task scheduling affects performance and cost and throughput. Task scheduling algorithms have not considered the impact of bandwidth as they are concerned mainly with CPU resource.With the algorithms considering only the availability of Virtual machine, the insufficient bandwidth results in the delay of resources.

Among task scheduling algorithms, the best known is Genetic Algorithm(GA). This algorithm evaluates many points in the search space simultaneously thus giving the efficient solution for the given problem [1]. Another algorithm Particle Swarm Optimization (PSO) is invented by Kennedy and Eberhart[2,3] considering the interaction of individual particles. This algorithm is developed by observing social attributes of animals likely bird flocking and fish schooling. PSO possess more pleasant characteristics than GA. In PSO, knowledge of good solutions are retained by all the particles as it has memory whereas in GA once the population is changed all the previous knowledge of a problem are discarded.

* Assistant Professor, Computer Science and Engineering, United Institute of Technology, Tamilnadu, India, *Email: kousalya198710@gmail.com*

** Principal, Computer Science and Engineering,Vidhya Mandhir Institute of Engineering and Technology, Tamilnadu, India, *Email: rlgs14466@gmail.com*

In Clonal Selection Algorithm a random antigen is selected from the pool. Instead, the best antigen can be selected using the Fitness value using Improved clonal selection algorithm.In this paper,Capacity of Virtual Machine, bandwidth, Number of Loads in Virtual Machine are the parameters taken into consideration and Improved clonal algorithm is proposed for optimizing Task scheduling.

2. RELATED WORK

Cloud computing is called as Heterogeneous system as it contains Large amount and different types of data. Inorder to enhance the utilization of resource, to increase the performance, for minimizing the cost, reducing the processing and completion time it is essential to perform task scheduling in the cloud environment. Various scheduling mechanisms proposed by several authors to solve various problems are as follows:

The authors of paper[4],[5],[6],[7] presenteda thorough survey about cloud computing. Basic fundamentals with various applications of cloud computing were discussed in these papers.

Jeffrey Dean and Sanjay Ghemawat[8] was about data processing in cloud computing. The author proposed MapReduce that is associated with implementation for processing and generating large data sets. MapReduce model is widely used for many heterogeneous purposes likely: it is easy to use as it hides fault-tolerant details, parallelization details, load-balancing and locality optimization even for new users. The authors [8] implemented MapReduce scaling large groups of machines that includes enormous number of machines. Author [8] exploits how programming model is restricted so that it makes way for easy parallel and distributes computations, making fault-tolerant computations, how redundant execution reduces the slow machines impact and how to react if there occurs any data loss or even if there is a machine failure.

LizhengGuo [9] elaborated regarding optimizing transfer and processing time is needed to an application program in the cloud environment even cloud holds huge data and many process that can schedule the services to the user. To minimize the processing cost and to optimize task scheduling LizhengGuo [9] have proposed particle swarm optimization (PSO). As resources being distributed all over the worldand data often bigger with narrow bandwidth, this PSO algorithm solves cost minimization problem. Results convey PSO produces optimal solution added with converging faster in large data sets than the other algorithms. Also it minimizes the completion time and hence PSO is more efficient for scheduling in cloud.

It is important to manage computing resources for complex applications in parallel computing. These kind of applications results in less resource utilization.XiaochengLiu[10] author proposed a Priority Algorithm for Task execution which is efficient than FCFS and Round Robin scheduling. Virtualization concept is implemented for improving resource utilization. Here priority-based allocation method is used for scheduling parallel jobs that can run jobs with low priority in a virtual machine with low capacity. LizhengGuo [9] have partitioned computing capacity into two tiers : foreground VM and background VM. Foreground VM processes the job with high CPU priority whereas jobs with low CPU priority is allocated to background VM. Backfilling technique [10] concern about low utilization caused by different node requirements of parallel jobs and it does not deal with low resource utilization because of parallel jobs.

A Priority based scheduling [10] is proposed to strengthen the workload of the parallel job in the cloud. This algorithm schedules the parallel jobs in the two tier architecture effectively.The author have combined backfilling and migration technique to schedule the process into two types of VMs based on priority thatwill reduce the waiting time and completion time of processes.Also author have shown through simulation that, Aggressive Migration Supported BackFilling (AMCBF) a consolidation based algorithm, shows better performance than other algorithms especially in terms of response time and bounded slowdown. Also it outperforms the commonly used EASY algorithm and AMCBF is robust as it allows inaccurate CPU usage estimation of parallel processes.

3. PROPOSED METHODOLOGY

Efficient resource utilization of parallel workload in cloud computing is discussed in this section. The objective of the proposed work is to identify the available resources and allocate the job in appropriate virtual machine. The computing capacity of VM is partitionedinto foreground and background. These portions are further classified into k blocks. The Improved Clonal Selection with PSO is a novel task allocation model which is utilized for optimize task scheduling. The parameters such as VM capacity, bandwidth, number of processors needed and number of loads for completion of the task are considered for determining the states such as neutral, loaded and low loaded. Based on these states, the Improved CS with PSO schedules the incoming jobs in the available blocks of virtual machine..

3.1. MATHEMATICAL MODEL

Based on the Capacity of VM, current workload assigned and the time taken for execution of a process, the works are assigned to the available VMs. VM can be categorized as loaded, low loaded and neutral. The highest affinity of a virtual machine is selected from low loaded virtual machine and loaded VM is considered as low affinity VM. In each iteration the job is assigned to low loaded machine from loaded machinebased on affinity measures. After selecting the highest affinity job, it is assigned to the respective virtual machine. It is repeated until all the jobs are assigned to the Virtual machine.

Makespan must be less for an optimizedscheduling. Let $T = \{T_1, T_2, \ldots, T_n\}$ be the total number of tasks and virtual machine processing J_n jobs be $M = \{M_1, M_2, \ldots, M_m\}$.

The makespan *ms* for task T_n is Cp_{ms}. Therefore,

Makespan Ms = max $\{Cp_{\text{ms.}} \mid n \in T, n = 1, 2 \ldots m \text{ and } s \in M, s = 1, 2, \ldots, m\}$

The waiting time of a job must be reduced to increase the makespan. Scheduling of preemptive taskare denoted as p_t.

Let us denote the time taken to complete all jobs on virtual machine be et_{ms}.

$$et_s = \sum_{m=1}^{i} et_{ms}, \text{ where} s = 1, 2, ..., t$$

Our main objective is to reduce the makespan that is done by comparing the execution time of a job and execution time of all jobs.

$$\sum_{m=1} et_s \leq Cp_{msi.} \text{ where} s = 1, 2, ..., t$$

$$\Rightarrow \qquad et_s \leq Cp_{msi.} \text{ where } s = 1, 2, .., t$$

Total number of nodes needed by job $= N(et_s, S_i)$

Load balancing reduces the waiting time of a job by executing it in low loaded VM.

$$Cp_{msi.} = \left\{ msi^{i}_{m=1} Cp_{m.}, msi^{j}_{s=1} \sum_{m=1}^{i} et_{ms} \right\}$$

Before assigning a job to the VM, its capacity, existing workload, bandwidth, speed at which it can execute and the execution time must be calculated.

$$V_k = et_{s_{nopk}} + et_{s_{mps}} + M_{ba_k}$$

where, $et_{s_{nopk}}$ is the number of processors in M_m, $et_{s_{mps}}$ is million commands per second of all processors in M_m and M_{ba_k} is the communication bandwidth of M_m.

Overall capacity of Ms is calculated by summation of capacity of each C

$$C = \sum_{m=1}^{i} C_m$$

Load balancing is done by calculating the overall workload of VM. For that, Individual VM workload M is calculated by,

$$W_{M_s,e} = \frac{Num\ of\ jobs(J,\ e)}{Serive\ Rate(M_s,\ e)}$$

Overall workload of M is calculated by adding workload W of all VM.

$$W = \sum_{p=1}^{n} W_{M_p}$$

Completion time of a VM is calculated by,

$$et_m = \frac{W_{M_s}}{C_m}$$

i.e. $et = \frac{w}{c}$

3.2. Neutral, Loaded and Low loaded

Load can be balanced by checking the state of virtual machine whether it is neutral, overloaded or Expected.

a. For neutral state *n*, find the standard deviation for the difference between execution time of each M_s and execution time of *M*.

$$\sigma = \sqrt{\frac{1}{i}\sum_{m=1}^{i}\left(et_m - et\right)^2}$$

```
If the standard deviation is equal to 0, VM is balanced.
If standard deviation ≤ Threshold condition
setVM asneutral
End if
```

b. For loaded state l, the workload of M is higher than its actual capacity.

```
    If (workload of M > Capacity of machine C)
        VM is loaded.
    End if
```

c. For alowloaded state u, the job can run on a particular VM, because total capacity of the machine is greater than the demand.

$$\rho = capacity - \left(workload\ +\ demand\right)$$

```
If $(\rho \le 0)$
VM is in lowloaded state
End if
```

Algorithm

```
Calculate Standard deviation of execution time of $M = \sqrt{\frac{1}{i}\sum_{m=1}^{i}\left(ex_m - ex\right)^2}$
If standard deviation $\leq$ Threshold condition set
    Data center is balanced
End if
If (workload of M > C)
    Data center is loaded.
End if
If $(\rho \geq 0)$
    Data center is low loaded condition
End if
        For each machine
        Supply of = $M_k = Max(V_k) - \frac{W_k}{V_k}$
        Demand of = $M_k = \frac{W_k}{V_k} - Max(V_k)$
```

4. K-LEVEL TASK ALLOCATION ALGORITHM USING IMPROVED CLONAL SELECTION (ICS) WITH PSO

4.1. Clonal selection Algorithm

Clonal Selection Algorithm is a type of immune algorithm that follows the principle of immune system. It is one of the best method for optimization. It is characterized by the population of Antigens and population of Antibodies.

The following steps provide the overview of CS Algorithm [13]

1. Initialization: A fixed N antibody contains antibody *m* and remaining antibody pool *r* where *m* is the representative of algorithms and *r* introduces additional diversity.
2. Loop: A single iteration is said to be as generation.G refers to the number of generations and is iterated until all antigens are exposed.

a. Selection –A random antigen is selected without replacement from the whole.
b. Exposure – The selected antigen is compared with all other antigens in antibodies forsimilarity values
c. Cloning – The antigen with highest similarity value in the pool is selected.
d. Mutation – The clone isagain compared to get the better antigen with high value.
e. Clone Exposure – Clone is compared with the antigen and similarity values are calculated.
f. Candidature – Antibody with highest similarity is selected as a part of *m*. if the selected clonehas more similarity values than that of highest similarity antigen that *m* contains, then it replaces the mentioned antigen.
g. Replacement –new antibodies will replace the individual antigens with low similarity values in *r* pool.

3. Finish: The memory *m* of antigen provides the solution for this algorithm. Hence the solution may be a single best antigen or group of antigens depending upon the problem domain.

The Clonal selection algorithm for assigning job among VM is given below.

```
Select job from job queue
Initialize Virtual machine VM
Determine the vector which contain affinity of VM
Select the $n$ highest affinity VM from VM in cluster.
Select the VM, $\sum_{i=1}^{n} VM_n$
Produce the number of VM for each of the $VM_n$
Assign VM to an affinity maturation process
Produce matured VM's.
Determine the affinity of the matured VM.
Among mature VM,
Elect VM by choosing best highest affinity one in the cluster.
Replace the $d$ lowest affinity VM from $VM_n$.
```

4.2. Clonal Selection Architecture

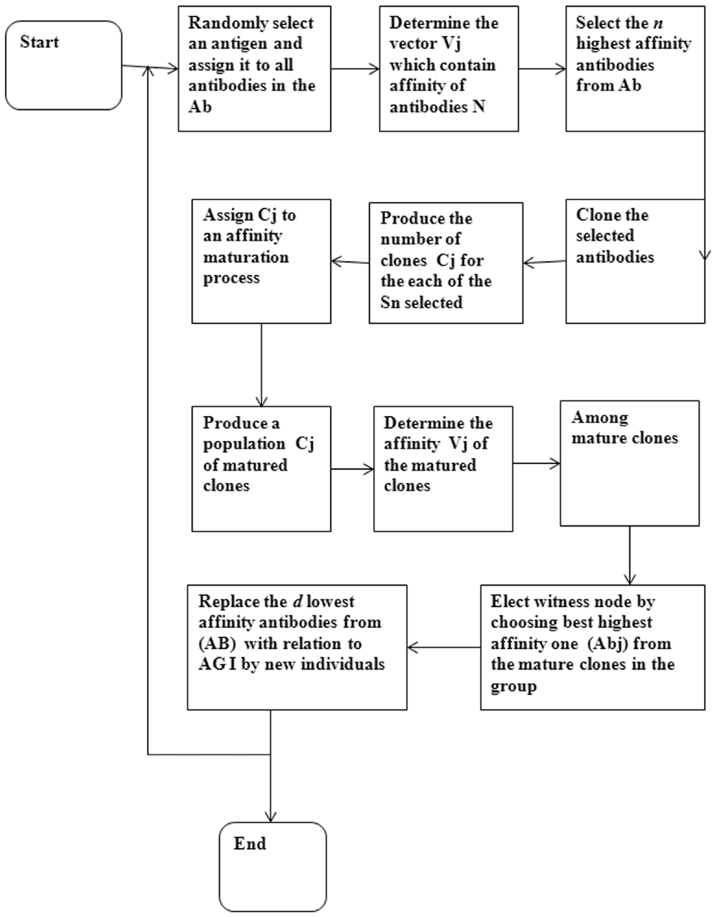
Start
Randomly select an antigen and assign it to all antibodies in the Ab
Determine the vector Vj which contain affinity of antibodies N
Select the n highest affinity antibodies from Ab
Clone the selected antibodies
Produce the number of clones Cj for the each of the Sn selected
Assign Cj to an affinity maturation process
Produce a population Cj of matured clones
Determine the affinity Vj of the matured clones
Among mature clones
Elect witness node by choosing best highest affinity one (Abj) from the mature clones in the group
Replace the d lowest affinity antibodies from (AB) with relation to AG I by new individuals
End

Figure 1: Architecture diagram for Clonal Selection algorithm

4.3. Particle Swarm Optimization

PSO is an optimization algorithm developed by Eberhart and Kennedy [11]. PSO is similar to Genetic Algorithm (GA) except PSO has no crossover and mutation. It is a technique evolved by the behavior of swarm of birds. In PSO "bird" is referred as "particle". The individual particles [12] adapt with their environment, search for food and share the information they gather. The information includes velocity, position and each particle finds its best position using fitness values and share the information. Each individual particle adds velocity with their position to reach the optimal point. The same is repeated by collective particles to find the global best solution.

As stated above, PSO is inspired by bird flocking. Consider group of birds searching for food randomly in an area. They do not know exactly where the food is. The bird which is near to the food shares the knowledge. In each iteration all the birds know how far the food is placed. So all the birds follow the bird that is nearest to the food area.calculate the fitness function f(x), for each particle x=(x1,x2,..., xn). Eachparticle hasvelocities and it follow the particle which has best fitness value. In each iteration, particles update two best values. pbest is the best fitness value (solution) of the particle has and gbest is the best fitness value ofthe population after a maximum number of iterations.Velocity and Position of particles are updated when the best values are found.

Algorithm

```
Step 1:    Initialize the population of particles
Step 2:    Fitness value of all particles is calculated by the formula $f(x) = (x1^2 + x2^2 + ... + xn^2)$
Step 3:    Best fitness valueof the particleis chosen and it is chosen as pbest and in next iteration if
fitness value is found better than previous fitness value, replace the newer one as pbest.
Step 4:    The best fitness value of the particle in the populationis considered asgbest
Step 5:    Calculate and update the velocity and position of particle
               $vel[\,] = vel[\,] + c1 * u1 * (pbest[\,] - present[\,] + c2 * u2 *$
               $(gbest[\,] - present[\,])\ present[\,] = present[\,] + vel$
Step 6:    Check for the target or Maximum iterations
Step 7:    If Maximal iterations is reached halt the loop else repeat from step 2.
```

4.4. Improved Clonal Selection

Clonal selection is a selection of superior clone among the mixed population. In the selection process the CS algorithm chooses a random antigen from the population. The pool of antigens is formed in a random manner. Hence for optimal selection, the selection process can be done using particle swarm optimization algorithm. In this proposed approach, Improved Clonal selection algorithm is implemented to achieveoptimal solution.

```
Select job from job queue
Initialize Virtual machine VM
Determine the vector which contain affinity of VM
Select the $n$ highest affinity VM from VM in cluster.
Check $n \nexists l$ and $n \nexists b$
then
```

```
Calculate the fitness value for all VM
Assign best fitness value as pbest
best fitness value o all VM is the gbest
Assign the gbest VM for execution of task
Electgbest VM and execution is done in gbest VM.
Replace the $d$ lowest affinity VM from $VM_n$.
```

The above algorithm is used for selecting the virtual machine considering the three states neutral, loaded and low loaded.For each M in foreground and back ground,

Virtual machine which has high volume, supply,completion time, fast speed of node in M and minimum current workload is thelow loaded virtual machine(u).

Virtual machine which has insufficient volume, demand, high completion time, low speed of node in M and maximum current workload is the loaded virtual machine (l).

Virtual machine which has sufficient volume, high demand, high completion time, low speed of node in M and sufficient current workload is the neutral virtual machine (n).

New jobs with high demand can be executed only in low loaded virtual machine and not in neutral or loaded machine.

5. RESULTS

The performance of proposed methodology and the results are shown in this section. The simulation is done using CloudSim. It is a familiar tool for the simulating cloud infrastructure and services. It provides the fundamental requirements to describe virtual machine, data centers and computing resources. The primary work of proposed method is to effectively schedule the tasks and to reduce the waiting time of jobs that enter into the virtual machine. This can be done by considering the factors throughput, makespan, flow time,waiting time, execution time and CPU memory.

5.1. Throughput

Throughput is a measure of number of jobs executed in a given amount of time. Throughput determines the efficiency of the work done. Using Execution time and number of jobs executed, throughput can be calculated.

Throughput = Number of jobs Executed/Total Execution time

From the below table and figure, it is clear that our proposed algorithm achieves high throughput.

Table 1
Throughput Report of various algorithms

Algorithm	*Throughput*
AMCBF	2.1558897047
CMCBF	0.1758047863
Gauss Newtons and k-means ++	0.0072534051
Buddy Allocation	13.885041680
Proposed Scheduling Method	15.455856860

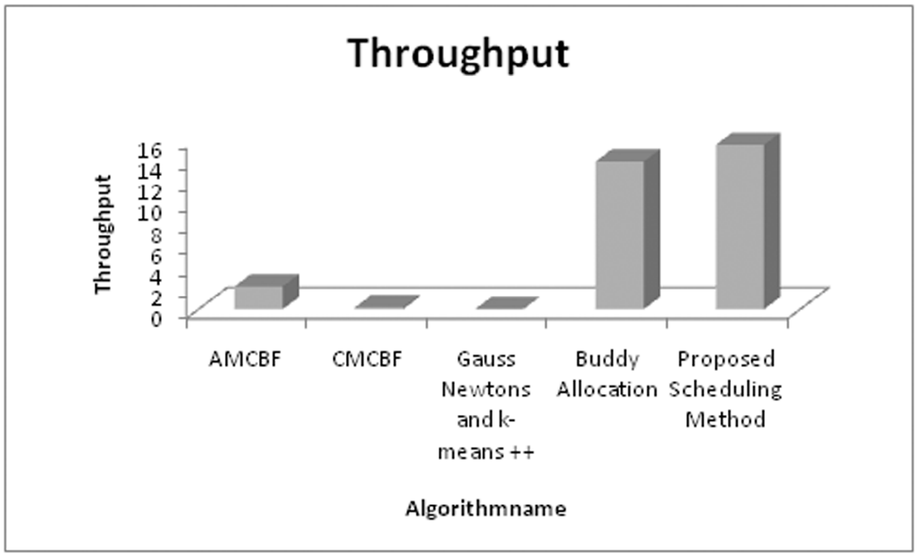

Figure 2: Throughput Result

5.2. Makespan

Makespan is the overall execution time of a machine. It is one of the performance measures of the machine. Lesser makespan indicates efficient scheduling. Makespan can be minimized by scheduling longest job on the longest resource. Minimizing makespan supports less computational cost and speed in execution.

Makespan = completion time of last job – starting time of first job

Table 2
Makespan report

Algorithm	*Makespan*
AMCBF	945.69230769
CMCBF	683.5
Gauss Newtons and k-means ++	130.781
Buddy Allocation	86.307692307
Proposed Scheduling Method	74.6502

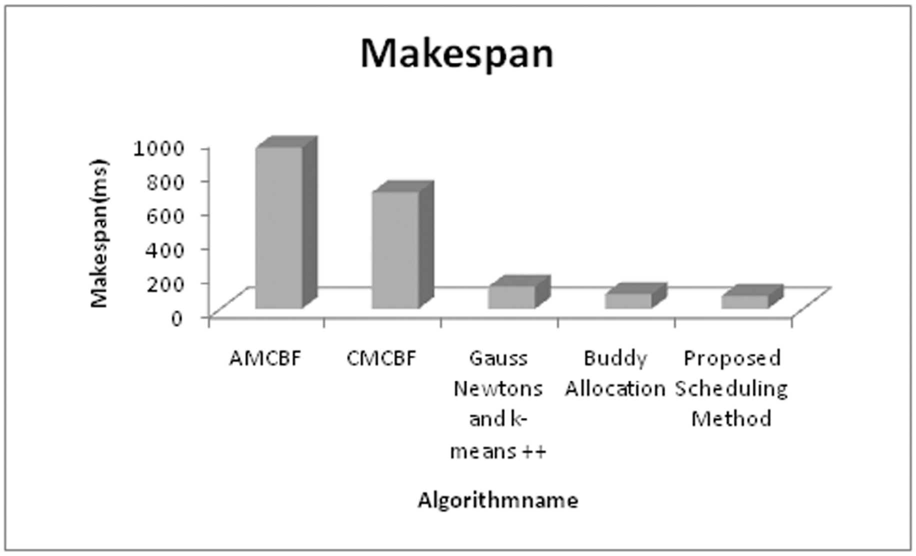

Figure 3: Makespan results

Makespan values are found for the proposed work and compared with other algorithms like AMCBF, CMCBF, Buddy Allocation. Improved clonal selection produces lesser makespan value which means the overall completion time is lesser than the other mentioned algorithms.

5.3. Flowtime

Flowtime refers to the time taken for the task to complete after the user requests for the job. Flowtime is minimized by executing the shortest job on the fastest resource.

Flowtime = Waiting time + Execution time

The below figure shows the flowtime of our proposed algorithm and other algorithms. The proposed algorithm achieves the minimum flowtime.

Table 3
Flowtime report

File name	*AMCBF*	*CMCBF*	*Gauss Newtons*	*Buddy Allocation*	*Proposed Scheduling Method*
05432179.pdf	359	281	262	53	48
Mk-TPDS08.pdf	437	844	345	56	51
Phr1.txt	531	495	422	57	54
Assembling Learning Objects.pdf	719	515	675	81	66
Recommendation Framework.pdf	812	1094	735	92	75
Images1.jpg	890	880	845	93	77
Phr2.txt	984	925	936	94	79
Phr5.txt	1062	975	1023	95	81
Phr5 9.txt	1140	1045	1163	95	82
Phr8.txt	1219	1126	1195	96	84
Phr7.txt	1297	1256	1213	97	88
Phr4.txt	1375	1324	1396	98	90
Phr3 .txt	1469	1425	1450	115	94

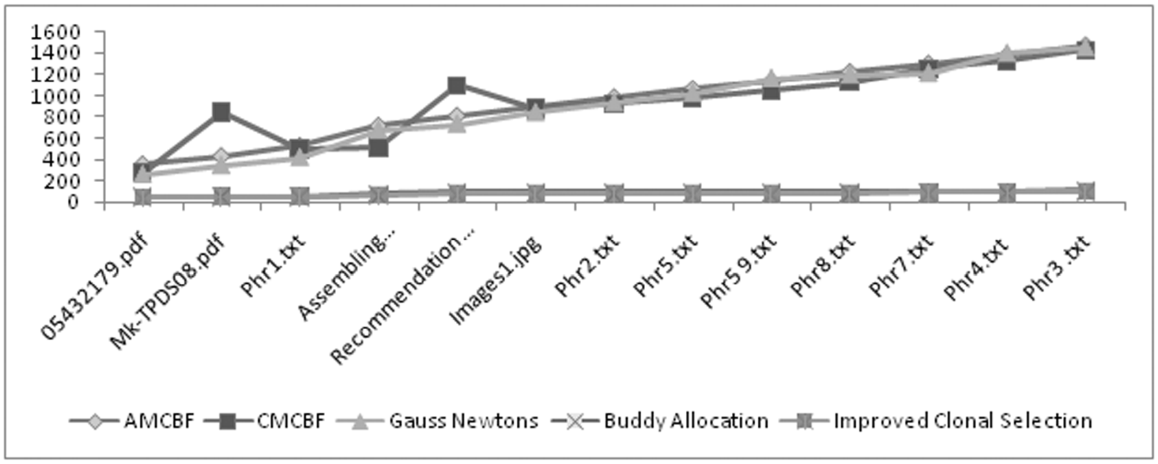

Figure 4: Flowtime result

5.4. Waiting time

It is the time taken for the process to start executionafter the user requests for it.

Waiting time = completion time of the job – execution time

The waiting time of the job in queue before it gets executed is calculated using the above formula. The waiting time for various workloads is calculated for AMCBF, CMCBF, Gauss Newtons and k++ scheduler, Buddy allocation and Improved CS algorithm. From the figure it is clear that the proposed method takes minimum waiting time.

Table 4
Waiting Time Report

Workload	*AMCBF*	*CMCBF*	*Gauss Newtons*	*Buddy Allocation*	*Proposed Scheduling Method*
1	38	49	42	25	20
2	40	43	38	24	19
3	44	43	40	27	21
4	40	49	40	27	22
6	42	46	42	27	21
7	44	40	45	29	22
8	32	46	45	28	23
9	42	39	50	29	24
10	38	43	60	28	20
11	43	51	100	28	22
12	45	44	140	30	24
13	36	53	160	33	25
14	40	46	160	30	22
15	36	51	170	29	24
16	41	44	170	33	26

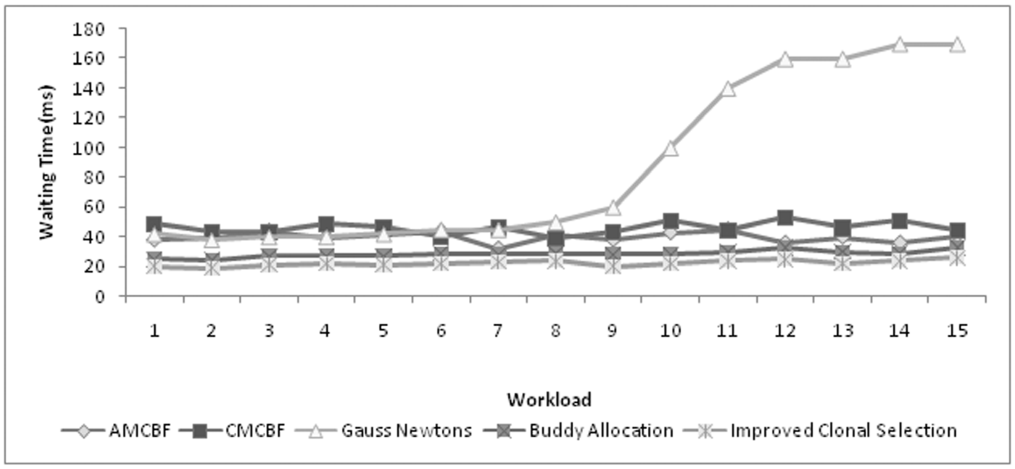

Figure 5: Waiting time result

5.5. CPU memory

It is the total number of memory utilized by thevirtual machine to completea task efficiently.

Table 5
CPU memory report

Workload	*AMCBF*	*CMCBF*	*Gauss Newtons*	*Buddy Allocation*	*Proposed Scheduling Method*
1	23	33	27	20	18
2	23	23	32	29	22
3	23	23	28	42	22
4	23	23	24	19	17
6	23	24	27	26	23
7	23	24	32	18	20
8	23	25	22	18	18
9	23	24	30	19	19
10	26	28	38	24	22
11	28	29	18	26	20
12	32	24	30	24	24
13	30	22	19	53	18
14	28	26	35	25	23
16	26	30	25	25	24

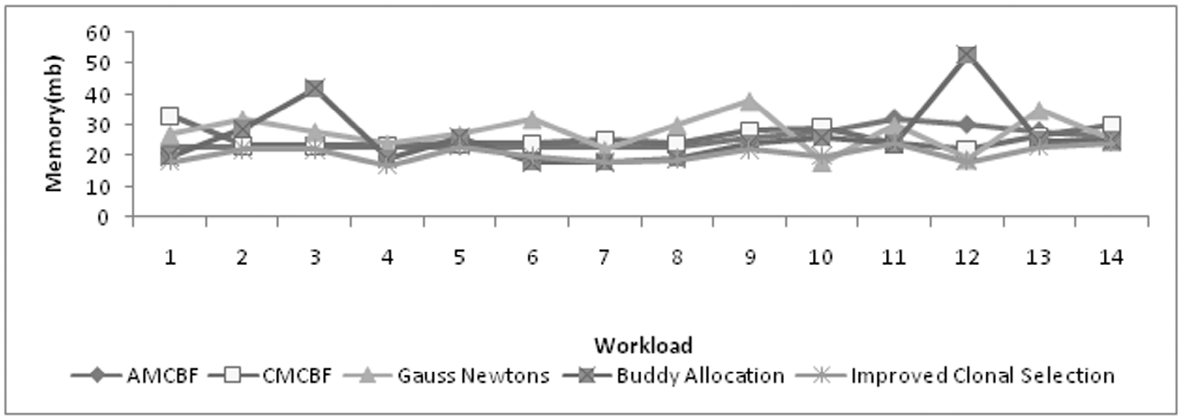

Figure 6: CPU Memory result

The table shows the CPU memory utilization by existing and proposed algorithms. In the fig. it is shown that the proposed approach obtains best result.

6. CONCLUSION

In cloud, for an optimal task scheduling, the execution time of a particular job must be minimized that it turn will reduce the makespan of the machine.We designed a global task scheduling algorithm combining Clonal selection algorithm and particle swarm optimization. We compared our work under parameters such as throughput, makespan, flowtime, waiting time and CPU memory. The experimental results show that the proposed algorithm produces better results than other algorithms. Scheduling is done by considering the workload of the machine and best virtual machine is allocated using best affinity values. The Proposed method clearly focused on creating simulation based evidences for evaluating the performance of the proposed algorithm. In future work, we focus to construct and implement in a real cloudenvironment.

REFERENCE

[1] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machinelearning, Addison-Wesley, New York, 1998.

[2] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory,in: Proceedings of the Sixth International Symposium on Micro Machineand Human Science, Nagoya, Japan, (1995), pp. 39–43.

[3] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. IEEEInt'l Conf. Neural Networks, Piscataway, NJ, USA, (1995), pp. 1942–1948.

[4] B. Ramamurthy & K. Madurai : Cloud Computing: Concepts, Technologies and Business Implications

[5] Dan C. Marinescu : Cloud Computing Theory and Practice. Pp. 1-50 .

[6] School of Software, Sun Yat-senUniversity : Introduction to Cloud Computing .

[7] Introduction to Cloud Computing .Fact Sheet , Fiche d'information. Page 1- 6..

[8] Jeffrey Dean and Sanjay Ghemawat : MapReduce: Simplified Data Processing on Large Clusters

[9] Lizheng Guo,Shuguang Zhao, Shigen Shen, Changyuan Jiang : Task Scheduling Optimization inCloud Computing Based on Heuristic Algorithm. In IEEE Journal of Networks, Vol. 7, No. 3, March2012

[10] Xiaocheng Liu, Albert Y. Zomaya, Fellow IEEE, Chen Wang, Bing Bing Zhou, Junliang Chen, TingYang, : Priority-Based Consolidation of Parallel Workloads in the Cloud. IEEE Transactions OnParallel And Distributed Systems, Vol. 24, No. 9, September 2013

[11] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory,in: Proceedings of the Sixth International Symposium on Micro Machineand Human Science, Nagoya, Japan, (1995), pp. 39–43.

[12] Brandstätter, Bernhard, and Ulrike Baumgartner. "Particle swarm optimization-mass-spring system analogon." *Magnetics, IEEE Transactions on* 38.2 (2002): 997-1000.

[13] Jason Brownlee "Clonal Selection Theory & Clonalg The Clonal Selection Classification Algorithm (CSCA)" Technical Report No. 2-02 January 2005