Journal of Advances in Information Technology Vol. 8, No. 4, November 2017

# **Standard Deviation Based Modified Cuckoo Optimization Algorithm for Task** Scheduling to Efficient Resource Allocation in Cloud Computing

Mahendra Bhatu Gawali
IT Dept, Thadomal Shahani Engineering College, Bandra (W), University of Mumbai, Mumbai, MS, India
Email: gawali.mahen@gmail.com

Subhash K. Shinde
Lokmanya Tilak College of Engineering, Kopar Khairane, Navi Mumbai, University of Mumbai, Mumbai, MS, India
Email: skshinde@rediffmail.com

***Abstract*—The Cloud Computing has an epochal technology now a day. Managing the incoming request (tasks) to avail-able resources is a challenge for scientist and researchers. This paper proposes a Standard Deviation based Modified Cuckoo Optimization Algorithm (SDMCOA) for task scheduling to efficiently manage the resources. The proposed sys-tem works, in two phases. In the first phase, the sample initial population have been calculated among**

**the available number of task's population. Rather to take**

**the sample randomly, if an appropriate population's sample for an experiment are chosen then there are more chances to get optimal result. In second phase, the Cuckoo Optimization Algorithm has been modified with respect to immigration and laying stage. This helps to improve the performance of the system. The experimental results using Cybershake Scientific Workflow shows that the proposed SDMCOA performs better than existing methods BATS, COA in terms of finish time and response time.**

***Index Terms*—Cloud Computing, task scheduling, modified cuckoo optimization, resource utilization**

## I.INTRODUCTION

Now days, industry and academia both are shifting their traditional way to utilize services offline to online. The technology which can make it possible is known as Cloud Computing. The Cloud Computing centres are responsible to hosts the applications and services such as Software as a Service, Platform as a Service, and Infrastructure as a Service. The Cloud Computing centres are builds of various specification computers or servers which are connected together. The Cloud Computing is the next paradigm of parallel and distributed computing to provide the resources. Utilization of the services by the service user and service provider both collectively

Manuscript received July 16, 2017; revised November 28, 2017.

formed Service Level Agreement [1]. The Cloud Computing system which is based on “pay-as-you-go” model makes it more powerful than others. The Virtualization [2] is the technology which adds strong corner to Cloud Computing. The Virtualization is actually abstracts the computing resources such as CPU, memory and other physical devices. Whenever, user submits a request to the cloud computing then such virtualization generates virtual machines to fulfil it. The Cloud Computing is basically providing any type of service (software or hardware) over Internet. To provide soft-ware or hardware services to the service user the Cloud Computing should balance thencoming request load with avail-able infrastructure. The 1and1 [3] developed a load balance system where user has the provision to shift one server to another manually. The Amazon Web Service (AWS) cloud provider [4] implemented task placement strategies by Bin-pack algorithm. This Binpack algorithm placed tasks based on the demanding percentage of computing resources such as CPU and memory. It randomly placed the tasks for execution.

The Microsoft Azure Scheduler [5] schedules the jobs by kept the job execution result history. The scheduler REST (Representational State Transfer) API is responsible to man-age the interactions between scheduling activities. Round Robin and Least Connection Algorithms are developed by the Century Link [6] Cloud Service Provider. The Rackspace has utilized the Random, Round Robin or Least Connection algorithm to manage the incoming traffic over the avail-able computing infrastructure. If computing resources such as CPU or RAM were not sufficient to incoming tasks demanded then a weighted algorithm has used to handle such situation [7].

In this paper, we focused on task scheduling and resource allocation in Cloud Computing. From the





performance and profit point of view these two are milestones of Cloud Computing. We have used scientific applications [8] as an input to the proposed system.

In order to minimized the response time and maximize the utilization of Cloud Computing resources the scheduling has performed the keen role. Scheduling is properly managed the incoming request (tasks) over the available resources with having some constraints. As far as scheduling is concerned lot of work have been noted on it, but still there is scope for improvement. Further, nature-inspired techniques such as Particle Swarm Optimization (PSO) [9], another nature-inspired Ant Colony Optimization [10] apart from this Honey Bee optimization [11] are some of existing optimization methodologies utilized to solve the scheduling and resource allocation problem in Cloud Computing to minimized the makespan, response time, throughput and maximized the utilization of computing resources such as CPU, memory and bandwidth etc. Basically, we focused and modify the scheduling of tasks by using Cuckoo Optimization Algorithm [12].

Our proposed SDMCOA has given more fruitful result in terms to minimize response time and finish time. The pro-posed SDMCOA approach also more efficient to maximize utilization of Cloud Computing resources.

> Our key contributions in this paper are as follows: 1.Mathematical based population selection adds strong corner in proposed system.
>
> 2.Iteration method has been developed in proposed SDM-COA.
>
> 3.The modified operators such as immigration and laying are utilized to schedule the tasks in optimal way.
>
> 4. The performance has been evaluated of proposed SDM-COA with existing system by Cloudsim Simulator.

The rest of this paper is organized as follows. Section II focuses up on related work of scheduling. Section III elaborates the proposed system architecture. Section IV form the task scheduling problem. Section V describes the steps of Standard Deviation based population selection. Section VI de-scribes the proposed SDMCOA for tasks scheduling. Section VII evaluates the SDMCOA approach. Demonstrates the simulation result and valuation with existing system in Section VIII. Finally, we conclude with future direction in respective Section IX.

## II. RELATED WORK

This section briefly describes the state-of-the-art for task and resource allocation in the Cloud Computing. Every existing system tried to achieve the optimum results by applying their own point of view. Even though the system given optimum result but still there is a scope for better optimum values to make system more appropriate.

Author Lizheng Guo *et al.* [13] proposed a system which optimized transfer and processing time of an application program. The system has based on the natured inspired particle swarm optimization. Here,

> authors applied optimization method to minimize the processing cost of the task.

Kun Li *et al.* [10] proposed a system to balance the load of the entire system and the incoming requests. The major aim of authors was to minimize the makespan of the input tasks. To achieve this aim authors has modified ant colony optimization algorithm. This approach mainly responsible to decreased the computation time of the tasks. Apart from this author has also considered the load up on the virtual ma-chine.

Author Dhinesh Babu L. D. and P. Venkata Krishna [11] have been proposed a system by considering the current load of virtual machine. If any virtual machine was overloaded and at the same time other virtual machine was under loaded then such system has optimized this by honey bee behaviour. This system has increased the throughput by such optimization method. Jinn-Tsong Tsai *et al.* [14] proposed a system by combination of differential evaluation algorithm with Taguchi method. This system has described the cost for processing and waiting time model for tasks. To reduce the makespan and cost of task processing was an aim of this system. Mohand Mezmaz *et al.* [15] proposed a system of parallel bi-objective hybrid genetic algorithm that considered makespan and energy consumption. To reduce the overall energy authors have utilized Dynamic Voltage Scaling. The limitation of this sys-tem is that author has focused only precedence-constraints parallel application. Author Luiz Fernando Bittencourt and Edmundo Roberto Mauro Madeira [16] proposed a system to optimize the cost of real time application such as work-flow. To minimize the cost authors have used The Hybrid Cloud Optimized scheduling algorithm. This system has re-executed the task according to the priority when the makespan deadline is increased. The pre-emption method has increased the complexity which has not focused properly here.

Baomin Xu *et al.* [17] proposed a system to schedule the tasks based on the Berger model. Basically, Berger model is utilized in social wealth. Authors have modified this model and utilized to reduce the makespan of execution of tasks. This system focused the actual demanded resources and actual allotted resources to the specific task. In the second view this system has also considered the type of tasks. As per the types of tasks the system has done classification before demanding the resources. Hong Sun *et al.* [18] proposed a QoS based task scheduling for resource allocation algorithm in cloud environment. Wanneng Shu *et al.* [19] proposed a system considering the energy consumption and makespan by introducing immune clonal optimization method. As new request (task) generated then this proposed system managed the resources based up on the Immune Clonal Selection Algorithm.

Author Mohammed Abdullahi *et al.* [20] focused on task scheduling in cloud computing environment based on Symbiotic Organism Search optimization. The system has reduced the makespan and increased utilization of resources. Optimization of tasks and utilization of resources have been managed in this system. Fan Zhang *et al.* [21] proposed a system which scheduled the



*Journal of Advances in Information Technology Vol. 8, No. 4, November 2017*

various types of tasks on the cloud computing. To scheduled variety of tasks to execute on cloud the system is based up on ordinal optimization. An Ordinal optimization method basically utilized for complex systems. Various workflows have been utilized as an input to the cloud system. Author Zhaobin Liu *et al.* [22] proposed a system has reduced the communication cost of tasks by introducing a fuzzy clustering method.

Tasquia Mizan *et al.* [23] proposed a system to schedule the tasks to gained maximum profit by Modified Bees Life Algorithm. Authors Wang *et al.* [24] considered non pre-empted and independent tasks as an input for the system.

Xiaofeng Wang et al. [25] proposed a system to optimized makespan and reliability by Look Ahead Genetic Algorithm. This algorithm is based on reliability driven reputation which generate the reliability of the allocated resources in distributed system. Author Xiaoli Wang et al. [26] proposed a system to scheduled energy efficient jobs based on map-reduce frame-work. The

system has utilized the Googles massive data as an input. Authors have also utilized the genetic algorithm to schedule Googles massive data. Gang Shen and Yan-Qing Zhang [27] have proposed a shadow price guided algorithm for schedule the tasks to improve the performance of cloud computing. The base of a shadow price guided algorithm is also genetic algorithm. Ye Huang et al. [28] proposed a system to improve the scalability and flexibility of the resources by community aware scheduling algorithm. Wei Wang et al. [29] proposed a system to manage the resources optimally by Dominant Resource Fairness. Such a system has worked on the large number of heterogeneous system. Shridhar G. Domanal and G. Ram Mohana Reddy [30] proposed a system to maintain the load balancing by introducing modified throttle algorithm. Authors [31] have also proposed the VM-assign load balance algorithm to allocate the tasks to virtual machine based in their status. Xiao-long Zheng *et al.* [32] proposed a system to schedule the task to allocate the resources by Pareto based fruit fly optimization algorithm (PFOA). For performance evaluation of this system authors did not considered other existing systems. The authors Madni et al. [33] has presented the detailed work on the scheduling of resources specially Infrastructure as a Service in Cloud Computing. Scheduling of resources strategies have been listed in research work [34]-[36]. Author Abdulhamid S. M. *et al.* focused on the scheduling on resources by League Championship Algorithm [37]. The detailed survey of scheduling has been focused by Abdulhamid *et al.* [38]. S.

M. Abdulhamid *et al.* has implemented the system with secure tolerance of fault [39]. Heuristic algorithms for task scheduling has been performed by S. H. H. Madni [40]. Author Calheiros *et al.* [41] have proposed a system for task scheduling based on Heuristic approach.

## III. PROPOSED SYSTEM ARCHITECTURE

The proposed Standard Deviation based Modified Cuckoo Optimization Algorithm has been shown in Fig.

1. Initially, The Cybershake Scientific Workflows tasks have an input for the system.

Figure1. Proposed SDMCOA system architecture.

The tasks are heavy in size and time consuming while running. These tasks are arranged properly in to task queue and then allocate the task as per the availability of the resources is done by proposed Standard Deviation based Modified Cuckoo Optimization Algorithm (SDMCOA).

## IV. PROBLEM FORMATION FOR TASK SCHEDULING

Cybershake performs lot of computation on very huge datasets which is generated from simulation Strain Green Tensor (SGT). Such generated data in the form of

"master" SGT files for x and y dimensions. This master SGT data quantifies the relationship between motion at a site and motion throughout the region. The ExtractSGT jobs may therefore be considered data partitioning jobs. In next level Synthetic seismogram are generated for each iteration variation by the Seismogram Synthesis jobs. The Peak intensity values are calculated by the PeakValueCalcOkaya jobs for each synthetic seismogram. The resulting synthetic seismogram and peakintensities are collected and compressed by the ZipSeismo-grams and ZipPeakSA jobs to be staged out and archived. These jobs may be considered as simple data aggregation jobs. Of the computational jobs, seismogram synthesis jobs are the most computationally intensive [27]. As shown in Fig. 1 a scheduling of task in Cybershake Scientific workflow in which every node represents a task and each edge represents the dependency between the tasks. Fig. 2 is the Cybershake Scientific Workflow tasks.

The task which has no predecessor is represented as

„Tstart and the task which has no successor is

represented as „Tend. The Cloud Computing has executed various tasks based on virtual machines. The Virtual Machine is basically formed by combination of computing resources such as CPU, memory and bandwidth.

The execution of allocated task by respective virtual machine based the following equ. 1

AST (ti, VMi) = Max(EST (ti , VMi), Ready(VMi)) (1)

This eq. 1 contains Ready (VMi) shows the earliest time when VMi is ready to schedule. The virtual machine is in ready condition that means none of the task is currently running on it. EST (ti, VMi) is the earliest start time of task on a virtual machine.



*Journal of Advances in Information Technology Vol. 8, No. 4, November 2017*

vertopal_f470afb54b004d538d800c66b7504665/media/image2.png

Figure 2. Cybershake scientific workflow.

0 if ti = tempty

Maxtj€ pred (ti) AFT(tj, VM1) if VMi = VM1 (2)

In above equ. 2 pred (ti) is the set of immediate predecessors of ti and RT (tj, ti) is the response time between task tj and task ti respectively.

The actual finish time of task ti is on virtual machine „VMi is determining by the following equ. (3)

AFT (ti,VMi) = AST (ti, VMi) +W (ti,VMi) (3)

*A. Fitness Function*

The Fitness Function for the task scheduling up on virtual machine can be calculated by equ. 4.

*Fitness (i) = maxVMi€ VM (FT (VMti))* (4)

*B. Task's Evaluation Parameters*

The parameters are used to check the performance of the task scheduling in proposed system.

- Response Time: Whenever the demanding resources by tasks are free then to takes the next task for execution is easy. But when actually the demanded computing re-sources are busy then algorithm performs crucial role to schedule incoming task over available resources. Response Time is one of the important evaluation parameter in computer based system. As response time is low the system is better to use.

- Finish Time: This is second parameter which indicates complete execution of set of task over time. As finish time is low then proposed system works better to schedule the tasks, execution of certain tasks and free the resources as soon as execution completed.

V. STANDARD DEVIATION BASED POPULATION SELECTION

The limitation of an existing system is to select the population for experiment randomly. Whereas, the proposed population selection algorithm initially calculates range of population appropriately, this results in the performance of the scheduling algorithm. Notations used in following equations have been listed in Table I.

TABLE I. NOTATION DESCRIPTION

| Notation | Description |
|---|---|
| Ts | A set of tasks of Cybershake Workflow |
| Te | A set of edges among the tasks |
| VMs | A set of Virtual Machine |
| I | no. of Iterations |
| tn | no. of tasks |
| vm | no. of virtual machine |
| te | no. of edges |
| Tstart | The start task in an application |
| Tend | The end task in an application |
| r | no. of repetition |
| LR | Laying Radius |
| LRvm | laying radius of vm |
| LRt | laying radius of task |

The proposed population selection algorithm consists of two steps, determining the sample deviation, Confidence interval to decide the lower-upper range of population for an experiment.

These two steps are described in detail.

*A. Determining the Sample Population*

Instead to process all the population we used to select the sample among the population. So, that the selection of such
Sample is very important for researchers. The selection of sample from population is completely depends on the types of problems. Here, we used the set of Cybershake Seismograph tasks which has content almost 8,00,000 of various sizes of tasks. The calculation steps for sample deviation areas follows.

1.In first step we have calculate the average of available tasks length. Average of this calculation is represented byx.

All tasks length addition

X=------------------------------------------------ (5)

Total Number of tasks
2.Then subtract the average value from the individual value of available set. The number of

task from a set is represented by „xi. Immediately do the square of this result by following equ. 6.

*Result= (xi – x2)* (6)

Then calculate addition of all results generated by equ. (6).

3.Now, we have to divide total number of sample task minus -1 with the result produced by equ. (6)

*Total number of task - 1*
*Answer =* ---------------------------------------- (7)

*Result*

4.Take square root of the sample variance generated in equ (7).

1/N-1=N i=1 (xi– x2) (8)

*B. Calculate the Confidence Interval*

The Confidence Interval is an important term in sample population selection. Whatever the sample we have chosen experiment how much we are confident about it? The answer of this question is to calculate the

Confidence Interval of the Sample population. The procedure to calculate Confidence Interval is as follows. 1.Initial stage is to find out how much percentage of Confidence Interval needed for selected Sample Population.

2.Subtract Confidence Interval percentage from 100 %.

3.Remaining percentage again divide into two parts. i. e. Upper and Lower range.

4.Calculate the Degree of Freedom.

i. e. Total number of tasks - 1;
5.Use T table [33] to find the exact value.

6.Calculate SE= SD/ n
7.Calculate T-value x SE = Result1 8.Subtract Result1 from mean
9.Mean- Result1= Lower Limit
10.Mean+ Result1= Upper Limit

## VI. PROPOSED STANDARD DEVIATION BASED MODIFIED CUCKOO OPTIMIZATION ALGORITHM

The Modified Cuckoo Optimization Algorithm has been implemented to get an optimal result with less number of iterations. Basically, Cuckoo Optimization Algorithm randomly generates an initial population as a habitat matrix that each member shows the current habitat of cuckoos. In this work each cuckoo in initial population represents a complete solution that will schedule the task to available virtual machines. Fig. 3 shows the flowchart of the proposed SDMCOA. In every iteration, immigration and laying stages are calculated with its fitness.

vertopal_f470afb54b004d538d800c66b7504665/media/image3.png

Figure 3. Proposed SDMCOA system flowchart.

*Phase- I Generating Initial Modified Cuckoo Habitat* To manage the task to allocate virtual machine a proper scheduling is needed. To keep this perceptive in

mind some iteration „I has been generated. The iterations „I has been the initial population.

*I= (NImax – NImin) x Rand [0,1] + NImin* (9)

This number „I suggests new iterations number to be generated by parent iterator at laying stage.

Each parent iterator is allowed to change the order of a limited tasks with the constraints of control flow dependency. This limited number is known as laying

Radius (LR). Each parent iteration has computed LRVM and LRt values.

The LRVMand LRt are computed for each iteration by using following equ. 10 and equ.fdg11 respectively.
LRVM = [ x current Iteration I / Total of all Iteration I]

x (varhivm- varlowvm) (10)

LRt = [ x current Iteration I / Total of all Iteration I]

x (varhit- varlowt) (11)

where, = maximum number of possible laying in the order of tasks or virtual machines to achieve schedule.

**Algorithm 1:** Modified Cuckoo Optimization Habitat **Input:** A Cybershake Scientific Workflow
**Output:** The Initial Population
1.Produce Iteration as a member of population 2.Calculate I, LRVM, LRt for this Iteration
3.Repeat
4.Until size of population
5.Calculate the fitness of all iterations
6.Set maximum fitness as Goal point
7.Global points = Goal Point.

8.End
According to the fitness function the values are arranged in ascending order to determine the best schedule.

*A. Modified Laying Stage*

In this stage the Iteration „I and LR (Laying Radius) formed for each iteration in initial population. In every iteration, each virtual machine is replaced randomly by a virtual machine in laying radius virtual machine limit and each task is also replaced randomly. For every iteration the new fitness of population has been calculated.

According to this calculations goal point and global optimum point are updated. The generated population should be sorted according to iterations fitness and then number of maximum Cuckoo Survived (Cmax) has to be selected from the beginning and others were deleted.

**Algorithm 2**: Modified Laying Stage
**Input**: The Initial Population
**Output**: The Laying Population
1. For
2. Each iteration in population do;
3. Generate new „I iterations by values of LRVM And LRt ;
4. Endfor
5. Calculation the fitness for all iterations
6. Set maximum fitness as a Goal point;
7. If
8. Goal point <Globalpoint;
9. Global Point = Goalpoint;
10. Endif

*B. Immigration to Optimal Iteration*

In this stage we have selected a point by consideration of execution the tasks with higher utilization percentage of resources like CPU, memory and bandwidth. Equ. 12 is the optimal habitat having the virtual machine and task base on,





Max(VMCPU,Mem,Band(VMi)) where VMi€ VM (12)

Once, we set the optimal point then other iterations immigrate towards it. We checked here the optimum global point which is closed to goal point. As shown in Fig. 4 „A is the goal point where „B and „C are the habitat having their own specification with the values of

resources (CPU, memory, and bandwidth). The Bs and Cs ultimate aim to reach up to goal point.

vertopal_f470afb54b004d538d800c66b7504665/media/image4.png

Figure 4. Optimal immigration habitat.

*C. Modified Stop Criteria*

In this proposed SDMCOA system after set the goal point we tested the algorithms to reach towards goal point. For this proposed system we have set the stop

criteria, if proposed systems algorithms results with steady values for up to 5 times of execution.

**Algorithm 3:** Modified Immigration Algorithm
**Input:** The laying population, CPUmax, memorymax, bandwidthmax
**Output:** The Migrated Population, Finish Time (FT), Respone Time (RT).

1.For
2.Each iteration in population do;
3.Generate CPUmax, memorymax, bandwidthmax
4.Each sets of Global point do;
5.For
6.Compare the iteration values and Goalpoint values of CPU, memory, bandwidth
7.Compare FT and RT of iteration values with Goalpoint FT and RT;
8.Endfor
9.Calculate the Fitness for all iterations
10.Set minimum fitness for FT and RT;
11.Set maximum fitness VMCPU,Memory,bandwidth
12.If
13.Goalpoint< Global point then
14.Globalpoint= Goalpoint
15.Endif

16.Calculate I, LRVM, LRt for this iterations
17.Endfor.

**Algorithm 4:** Modified Cuckoo Optimization Algorithm
**Input:** ACybershake
Scientific Workflow, Tasksrunsize, populationsize, VMnumber, Nmax, NImin, NImax, and F;

**Output:** Taskschedule

1.Run MCOAHabitat

2.For

3.Run MCOALayingstage

4.Run OptimalNmax

5.Run MCOAimmigration

6.Upto

7.Iterationsize.

TABLEII.DATACENTER INFORMATION

| Sr. No. | Information | Contains |
|---|---|---|
| 1 | Number of Datacenter | 1 |
| 2 | Number of Host | 1 |
| 3 | Number of Processing Units | 4 |
| 4 | Processing capacity (MIPS) | 9600 |
| 5 | Storage Capacity | 11 TB |
| 6 | Total Amount of RAM | 40 GB |

TABLEIII.DATACENTER CONFIGURATION DETAILS

| Sr. No. | Information | Contains |
|---|---|---|
| 1 | Allocation Policy | SDMCOA |
| 2 | Architecture | X86 |
| 3 | Operating system | Linux |

| Sr. No. | Information | Contains |
|---|---|---|
| 4 | Hypervisor | Xen |
| 5 | Upper threshold | 0.8 |
| 6 | Lower threshold | 0.2 |
| 7 | VM Migration | Enabled |
| 8 | Monitoring Interval | 180 |

TABLEIV.HOST CONFIGURATION DETAILS

| Sr. No. | Information | Contains |
|---|---|---|
| 1 | RAM | 40 GB |
| 2 | Bandwidth | 10,00,000 |
| 3 | Operating System | Linux |
| 4 | Hypervisor | Xen |

TABLEV.CUSTOMER CONFIGURATION DETAILS

| Sr. No. | Information | Contains |
|---|---|---|
| 1 | Users | 1 |
| 2 | Cloudlets sent per minutes | 50 |
| 3 | Avg. Length of Cloudlet | 50,000 |
| 4 | Avg. Cloudlet file Size | 500 Bytes |
| 5 | Avg. Cloudlet output size | 500 Bytes |

TABLEVI.CUSTOMER CONFIGURATION DETAILS

| Sr. No. | Information | Contains |
|---|---|---|
| 1 | Number of VMs | 1 |
| 2 | Avg. Image Size | 1000 Bytes |
| 3 | Avg. RAM | 512 MB |

| Sr. No. | Information | Contains |
|---|---|---|
| 4 | Avg. Bandwidth | 1,00,000 Mbps |
| 5 | Procedure Element | 1 |
| 6 | Priority | 1 |
| 7 | Hypervisor | Xen |
| 8 | Scheduling Priority | Dynamic Workload |

## VII. EVALUATION OF PROPOSED SDMCOA APPROACH

*A.Experimental Setup*

The proposed SDMCOA approach work is

experimented on Cloud Simulator [42], which gives the

real-time environment scenario of Cloud Computing.

Datacenter Information has been listed in Table II, Table

III consist of configuration for Datacenter which includes



allocation policy, architecture, OS, hyper visor, scheduling and monitoring interval, threshold value etc. Host in the Datacenter used to show the amount of provisional RAM, bandwidth, storage capacity, power, processing element etc. of given task which process by datacenter. Table IV explains the host configuration details. Configuration details of customized simulation setup are given in Table V and it consist of general information of Datacenters like number of Datacenters, number of host, number of processing units, capacity etc. Every Datacenter component instantiates a generalized application provisioning component that implement a set of policies for allocating bandwidth, memory and storage devices to hosts and virtual machines. Table VI holds information related to storage area network capacity, latency and bandwidth.

## VIII. RESULT AND DISCUSSION

This section will brief about the performance of proposed novel SDMCOA approach.

*A. Evaluations*

Let, we evaluate our proposed SDMCOA approach with existing COA, BATS [43] on the given Cybershake Seismogram Synthesis tasks.

TABLEVII.COMPARISON OF PROPOSED SDMCOA WITH MABBLDC, COA,BATS ON FT IN MS WITH 20VMS

| Tasks | SDMCOA | MABBLDC | COA | BATS |
|---|---|---|---|---|
| Task 3 | 2613.79 | 2832.94 | 2913.79 | 3599.29 |
| Task 5 | 2613.07 | 2914.42 | 2913.07 | 3599.29 |
| Task 7 | 2611.02 | 2913.87 | 2911.02 | 3599.29 |
| Task 9 | 2606.93 | 2911.75 | 2906.93 | 3599.29 |
| Task 11 | 2636.78 | 2907.67 | 2936.78 | 3599.29 |
| Task 14 | 2556.36 | 2772.11 | 2856.36 | 3599.29 |
| Task 16 | 2554.44 | 2857.89 | 2854.44 | 3599.29 |
| Task 18 | 2537.48 | 2855.97 | 2837.48 | 3599.29 |
| Task 20 | 2533.36 | 2833.36 | 2833.36 | 3599.29 |
| Task 22 | 2540.06 | 2834.72 | 2840.06 | 3599.29 |
| Task 24 | 2563.70 | 2841.49 | 2863.70 | 3599.29 |
| Task 26 | 2532.86 | 2832.86 | 2832.86 | 3599.29 |
| Task 28 | 2535.37 | 2833.96 | 2835.87 | 3599.29 |

vertopal_f470afb54b004d538d800c66b7504665/media/image5.png

Figure 5. Finish time comparison.

*B. Evaluation of Finish Time*

In order to check the performance of our proposed SDMCOA algorithm, we have first applied the algorithm to the Cybershake Seismogram Synthesis Tasks. The performance of proposed SMDCOA evaluated, by the

finish time which is span of time from submission of task to complete the respective task. Fig. 5 and Table VII. shows the comparison between existing BATS, COA with the proposed SMDCOA approach. We have calculated the finish time which is span of time from submission of task to complete the respective task. We checked the performance over the finish time and our proposed SMDCOA has given better result as compared to existing BATS and COA.

TABLEVIII.COMPARISON OF PROPOSED SDMCOA WITH ON RT IN MS WITH 20VMS

| Tasks | SDMCOA | COA |
|---|---|---|
| Task 3 | 2832.44 | 2832.44 |
| Task 5 | 2832.44 | 2832.94 |
| Task 7 | 2832.44 | 2832.94 |
| Task 9 | 2832.44 | 2832.94 |
| Task 11 | 2832.44 | 2832.94 |
| Task 14 | 2771.61 | 2772.11 |
| Task 16 | 2771.61 | 2772.11 |
| Task 18 | 2771.61 | 2772.11 |
| Task 20 | 2771.61 | 2772.11 |
| Task 22 | 2771.61 | 2772.11 |
| Task 24 | 2771.61 | 2772.11 |
| Task 26 | 2771.61 | 2772.11 |
| Task 28 | 2771.61 | 2772.11 |

vertopal_f470afb54b004d538d800c66b7504665/media/image6.png

Figure 6. Response time comparison.

*C. Evaluation of Response Time*

The second performance parameter we have taken response time of the algorithm to the incoming tasks. The response time is basically the time when the request is actually entertained. In other words, we can say that the response time is directly dependent on the availability of the resources. The availability of the resources is dependent upon the scheduling of tasks. Because if the scheduling of task is done properly then naturally the resources will be free early or within the deadline so the response time will be less. While, comparing the response time as a second performance parameter of our proposed SDMCOA with existing BATS [43], we can



*Journal of Advances in Information Technology Vol. 8, No. 4, November 2017*

see our systems response time is almost less. The comparison has represented in Fig. 6. The comparison again shows in a tabular form in Table VIII We have considered two parameters such as response time and finish time for analysis of proposed SDMCOA with BATS, COA. As we are evaluating these results on Cloud platform so by generally the response time must be less.

## IX. CONCLUSION &FUTURE WORK

This paper describes a proposed Standard Deviation based Modified Cuckoo Optimization Algorithm for task scheduling to efficiently managed the resources in Cloud Computing. Calculation of sample initial population based on mathematical terms has given better results as compared to randomly selection of population. The results from various simulations using Cybershake Scientific Seismogram tasks as an input shows that the SDMCOA approach performs better than BATS, COA. More accurate population calculation methods may increase the performance of the Cloud Computing from scheduling point of view will be addressed in upcoming work.

REFERENCES

[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *Proc. 10th IEEE International Conference on High Performance Computing and Communications*, Dalian, China, 2008, pp. 5-13.

[2] J. E. Smith and R. Nair, "The architecture of virtual machines," *Computer*, vol. 38, no. 5, pp. 32-38, 2005.

[3] [Online]. Available: https://www.1and1.com/dynamic- cloudserverperformance

[4] Amazon. (August 2017). [Online]. Available: http://docs.aws.amazon.com/AmazonECS/latest/developer guide/task-placement-strategies.html
[5] Microsoft Azure. (August 2017).

https://opbuildstorageprod.blob.core.Windows.net/outputpdf- files/en-us/Azure.azuredocuments /live/scheduler.pdf
Century Link Cloud. (August 2017). [Online]. Available: [6]
https://www.ctl.io/legal/centurylinkcloud/
[7] Rackspace. (August 2017). [Online]. Available: https://support.rackspace.com/how-to/cloud-loadbalancer-scheduling-algorithms/
[8] Workflow Generator. (August 2017). [Online]. Available: https://confluence.pegasus.isi.edu/display/pegasus/ rator
[9] W. J. Liu, M. H. Zhang, and W. Y. Guo, "Cloud computing resource schedule strategy based on MPSO algorithm," *Computer Engineering*, vol. 37, no. 11, pp. 43-42, 2011.

[10]K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in *Proc. China grid Conference* , 2011, pp. 3-9.

[11]B. L. D. Dhinesh and P. V. Krishna, "Honey bee behaviour inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing*, vol. 13, no. 5, pp. 2292- 2303, 2013.

[12]R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing*, vol. 11, no. 8, pp. 5508-5518, 2011.

[13]L. Guo, S. Zhao, S. Shen, and C. Jiang, "Task scheduling optimization in cloud computing based on heuristic algorithm," *JNW*, vol. 7, no. 3, pp. 547-553, 2012.

[14]J. T. Tsai, J. C. Fang, and J. H. Chou, "Optimized task scheduling and resource allocation on cloud computing environment using improved differential

evolution algorithm," *Computers Operations Research*, vol. 40, no. 12, pp. 3045-3055, 2013.

[15]M. Mezmaz, N. Melab, *et al.*, "A parallel bi objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497-1508, 2011.

[16]L. F. Bittencourt and E. R. M. Madeira, "HCOC: A cost optimization algorithm for workflow scheduling in hybrid clouds," *Journal of Internet Services and Applications*, vol. 2, no. 3, pp. 207-227, 2011.

[17]B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Advances in Engineering Software*, vol. 42, no. 7, pp. 419-425, 2011.

[18]H. Sun, S. P. Chen, C. Jin, and K. Guo, "Research and simulation of task scheduling algorithm in cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 11, pp. 6664-6672, 2013.

[19]W. Shu, W. Wang, and Y. Wang, "A novel energy efficient resource allocation algorithm based on immune clonal optimization for green cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 64, 2014.

[20]L. Y. Zuo and L. F. Zuo, "Cloud computing scheduling optimization algorithm based on reservation category," *Computer Engineering and Design*, vol. 33, no. 4, pp. 1357-1361, 2012.

[21]F. Zhang, J. Cao, K. Li, S. U. Khan, and K. Hwang, "Multi-objective scheduling of many tasks in cloud platforms," *Future Generation Computer Systems*, vol. 37, pp. 309-320, 2014.

[22]Z. Liu, W. Qu, W. Liu, Z. Li, and Y. Xu, "Resource pre processing and optimal task scheduling in cloud computing environments," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 13, pp. 3461-3482, 2015.

[23]T. Mizan, S. M. R. A. Masud, and R. Latip, "Modified bees life algorithm for job scheduling in hybrid cloud," 2012.

[24]X. Wang, C. S. Yeo, R. Buyya, and J. Su, "Optimizing the make span and reliability for workflow applications with reputation and a look-ahead genetic algorithm," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1124-1134, 2011.

[25]X. Wang, Y. Wang, and H. Zhu, "Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm," *Mathematical Problems in Engineering*, vol. 2012, 2012.

[26]G. Shen and Y. Q. Zhang, “A shadow price guided genetic algorithm for energy aware task scheduling on cloud computers,” *Advances in Swarm Intelligence*, pp. 522-529, 2011.

[27]Y. Huang, N. Bessis, P. Norrington, P. Kuonen, and B.

Hirsbrunner, “Exploring decentralized dynamic scheduling for grids and clouds using the community aware scheduling algorithm,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 402-415, 2013.

[28]W. Wang, B. Li, and B. Liang, “Dominant resource fairness in cloud computing systems with heterogeneous servers,” in *Proc. INFOCOM*, 2014, pp. 583-591.

[29]S. G. Domanal and G. R. M. Reddy, “Load balancing in cloud computing using modified throttled algorithm,” in *Proc.*

*International Conference on Cloud Computing in Emerging Markets*, 2013, pp. 1-5.

[30]S. G. Domanal and G. R. M. Reddy, “Optimal load balancing in cloud computing by efficient utilization of virtual machines,” in *Proc. Sixth International Conference on Communication Systems and Networks*, 2014, pp. 1-4.

[31]X. L. Zheng and L. Wang, “A Pareto based fruit fly optimization algorithm for task scheduling and resource allocation in cloud computing environment,” in *IEEE Congress on Evolutionary Computation*, 2016, pp. 3393-3400.

[32]G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K.

Vahi, “Characterizing and profiling scientific workflows,” *Future Generation Computer Systems*, vol. 29, no. 3, pp. 682-692, 2013.

Available: [33][Online].

http://www.sjsu.edu/faculty/gerstman/StatPrimer /table. pdf/
[34]S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M.

Abdulhamid, “Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunity,” *Journal of Network and Computer Applications*, vol. 68, pp. 173-200, 2016.

[35]S. M. Abdulhamid, S. M. A. Latiff, and M. B. Bashir, “Scheduling techniques in on-demand grid as a service cloud: A review,” *Journal of Theoretical & Applied Information Technology*, vol. 63, no. 1, pp. 10-19, 2014.

*Journal of Advances in Information Technology Vol. 8, No. 4, November 2017*

[36]S. H. H. Madni, M. S. A. Latiff, and Y. Coulibaly, "An appraisal of meta-heuristic resource allocation techniques for IaaS cloud," *Indian Journal of Science and Technology*, vol. 9, no. 4, 2016.

[37]S. M. Abdulhamid and S. M. A. Latiff, "League championship algorithm based job scheduling scheme for infrastructure as a service cloud," in *Proc. 5th International Graduate Conference on Engineering, Science and Humanities*, Malaysia, 2014, 381-382.

[38]S. M. Abdulhamid, S. M. A. Latiff, and I. Ismaila, Tasks scheduling technique using league championship algorithm for makespan minimization in IaaS cloud," *ARPN Journal of Engineering and Applied Science*, vol.9, no. 12, pp. 2528-2533, 2014.

[39]S. M. Abdulhamid, M. S. A. Latiff, S. H. H. Madni, and O.

Oluwafemi, "A survey league championship algorithm: Prospects and challenegs, *Indian Journal of Science and Technology*, vol. 8, no. s3, pp. 101-110, 2015.

[40]S. M. Abdulhamid and M. S. A. Latiff, "A checkpointed league championship algorithm-based cloud scheduling schedme with secure fault tolerance responsiveness," *Applied Soft Computing*, vol. 61, pp. 670-680, 2017.

[41]S. H. H. Madni, M. S. A. Latiff, M. Abdullahi, S. I. M.

Abdulhamid, and M. J. Usman, "Performance comparison of heuristic algorithm for task scheduling in IaaS cloud computing environment," *Plos One*, vol. 12, no. 5, p. e0176321, 2017.

[42]R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: A toolkit for modelling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software Practice And Experience*, vol. 41, pp. 23–50, 2010.

[43]M. B. Gawali, and S. K. Shinde, "Implementation of IDEA, BATS, ARIMA and queuing model for task scheduling in cloud computing," in *Proc. Fifth International Conference on Eco-friendly Computing and Communication Systems*, 2016.

**Mahendra Bhatu Gawali** received his BE degree in 2008 and M.E. degree in 2013 from North Maharashtra University, Jalgaon, MS, India. Currently he is pursuing his Ph.D. at Thadomal Shahani Engineering College, Bandra (W), University of Mumbai, Mumbai, India. He focuses on Task Scheduling and Resource Allocation in Cloud Computing.

vertopal_f470afb54b004d538d800c66b7504665/media/image7.png

**Subhash K. Shinde** is working as a Professor in the Department of Computer Engineering at Lokmanya Tilak College of Engineering, Navi Mumbai, India. He received his Ph.D. from Swami Ramanand Teertha Marathwada University, India in 2012. He has published more than 40 research papers in the field of Web Mining, Frequent Pattern Discovery and Integration of domain knowledge in web personalized recommendations in the reputed journals and conferences.

vertopal_f470afb54b004d538d800c66b7504665/media/image8.png