ELM BASED SCHEDULING WITH PARTICLE SWARM OPTIMIZATION (MLQS-PSO) OF VMS IN QUEUEING HETEROGENEOUS CLOUD COMPUTING SYSTEMS

[1]S.Rekha, [2]Dr.C.Kalaiselvi

[1]Ph.D.Research Scholar, Department of computer science, Tirupur kumaran college for women, Tiruppur.
Assistant Professor, Department of IT, Dr.N.G.P. Arts and Science college, Coimbatore.
[2] Head and Associate professor, Dept of Computer Applications, Tirupur kumaran college for women, Tiruppur

ABSTRACT

This article investigates in cloud computing systems about problem of delay optimal Virtual Machine (VM) scheduling holds constant resources with full infrastructure like CPU, memory and storage in the resource pool. Cloud computing offers users with VMs as utilities. Cloud consumers randomly demand different VM types over time, and the usual length of the VM hosting differs greatly. A scheduling algorithm for a multi-level queue divides the prepared queue towards lengthy and various queues. System is allocated with single queue in to several longer queues. The systems are allocated to one queue indefinitely, usually on any basis of process property, like memory size, process priority, or process sort. Every queue will have its self-algorithm for scheduling. Likewise, a system that's taking in a less preference queue is so lengthy; a high-priority queue can be transferred. Using Particle Swarm Optimization Algorithm (MQPSO), Multi-level queue scheduling has been done. To evaluate the solutions, it explores both Shortest-Job-First (SJF) buffering and Min-Min Best Fit (MMBF) programming algorithms, i.e., SJF-MMBF. The scheme incorporating the SJF-ELM-specific scheduling algorithms depending SJF buffering and Extreme Learning Machine (ELM) is also being proposed to prevent work hunger in an SJF-MMBF system. Furthermore, the queues must be planned, which is usually used as a preventive fixed priority schedule. The results of the simulation show that the SJF-ELM is ideal inside strong duty as well as maximum is environment dynamically, with an efficient average employment hosting rate.

Keywords: Delay-optimal virtual machine, scheduling algorithm, Shortest-Job-First, Min-Min Best Fit, Multi-level queue scheduling, VM-hosting durations and Particle Swarm Optimization.

1. INTRODUCTION

Cloud computing, a methodology for offering all-round, suitable and ease usage on accessing resources computationally with combined which get
configured easily as well as revealed via effort minimally and interaction from server to service (e.g., networks, servers, storage, application, and services). Cloud computing uses the term "cloud" for a platform that has all kinds of storage computation, etc. [1,2] tools. There are triple services provided by cloud. The first one is an Infrastructure as a Service (IAAS) and extends the infrastructure for cloud users for different resolves such as storage systems and computing services. The second is really the Platform as a Service (PAAS), which produces customers with the platform to make applications for this platform. Second, Software as a Service (SAAS) provides end users along software, meaning users should not have to install the software according to their specific computers and will use the software right from the cloud. [3,4].

Cloud computing is the IT industry's need because of the wide variety of facilities offered by cloud computing. Delivering of the its services done through Internet. Resources that connect its services should also have the ability to access the Internet. Devices have much less memory, a lightweight browser, and the operating system. Cloud Computing provides several benefits: It saves costs because the initial installation of many resources is not needed; it provides scalability and flexibility; the number

of services per requirement may be increased or decreased; Maintenance costs are much lower because the cloud providers control each asset[5,6].

In the cloud computing context, scheduling tasks in accordance with flexible time for the Virtual Machines (VMs), that also requires the correct sequence to be found wherein tasks could be performed in transaction logic constraints. Cloud computing's task scheduling is a challenge. Formulate the VM scheduling as a decision-making process in this queueing cloud computing system, in which the decision variable is the VM configuration vector and objective in optimization would be lagging concert inside the medium total time of the job [7].

An online low-complex scheme combining Shortest-Job-First (SJF) buffering and Min-Min Best Fit (MMBF) scheduling algorithms, i.e. SJF-MMBF, is implemented for identify the results. The plan which blends buffering of SJF with RL-based scheduling algorithms, i.e. SJF-RL, also suggested in diminishing the possible for demand of job in SJF-MMBF. Nonetheless, since continued high purchasing and maintenance costs for cloud infrastructure, over-purchasing cloud infrastructure is inefficient to respond quickly to the resource requirements of all cloud users. And there is no optimal scheduling outcome in that work based on the single level queue. This issue is the focus of this work[8,9].

A multi-level algorithm for scheduling the queue for multiple-level scheduling partitions, designed for several different queues is proposed for this work. The processes are allocated in single queue indefinitely, usually based on some process property, like memory size, process priority, or process sort. There is an algorithm for each queue. Likewise, a system that waits as well long in a lower-priority queue could be relocated to a higher-priority queue. Using the Particle Swarm Optimization (PSO) algorithm, multi-level queue scheduling has been undertaken. It controls both buffering for Shortest-Job-First (SJF) and scheduling algorithms for Min-Min Best Fit (MMBF) [10].

2. LITERATURE REVIEW

In order to minimize the cost of reservations, Karthick et al. [11] has suggested a multi-queue scheduling (MQS) algorithm using a worldwide scheduler. The most important section in cloud computing is scheduling. The vital goal in planner globally maximizes resources distribution. Scientists add significance to the design of a cloud-specific work scheduling algorithms. The scheduling of jobs in the cloud has been one of the key events from the client should pay for services according to the time required. The suggested methodology portrays on job clustering created upon exploding period. While traditional methods like First Come First Serve, EASY, Shortest Job First, Combinational Backfill and Improved Backfill are scheduled, fragmentation is created utilizing balancing spiral method.

A multi-level queue (MLQ) task planning algorithm was suggested by Biswas et al [12] to reduce the range of parallels between subtasks while infringing precursors relations. Our principal objective is to take advantage of algorithms for the scheduling of algorithm tasks in terms of the span, time complexity, the use of resources, system performance and dynamic nature. Via experiments, our contribution is evaluated and calculated.

Jaspreet Singh and Deepali Gupta [13] implemented a Smarter MQS template that essentially divides user roles into two work queues and then places greater emphasis on integrating job trends by combining user tasks from both queue, this technique will allow us to reduce energy use, although, of course, to some extent, the completion time and actual cost will be through. In the cloud computing environment, the suggested technique may attain a high level of job scheduling.

Zhang and Zhou [14] suggested a framework for cloud tasks predicated on a strategy with two-stage. It recreates depending VMs through information scheduling traditionally, thus saving time for tasks waiting for VMs to be produced. This automatically matches tasks in their most appropriate VMs,

thereby saving the expense of their execution. This minimizes the delay for VMs to schedule tasks under the principle of meeting task timelines, thereby reducing the costs to be charged by users using VMs.Compared to those following traditional methods, the widely deployable algorithms are developed and outlined to enhance the planning and execution of cloud tasks.

Sumit Arora and Sami Anand [15] have suggested an effective scheduling algorithm that will truly work to produce better results than conventional scheduling approaches. The implemented algorithm is simulated under different conditions of this Cloud Sim framework and described the best results with decreased wait time and processing time for maximum use of the resource and minimal overhead.

Elmougy, et al [16] developed a new hybrid scheduling depends on task is represented as (SRDQ) integrating Round Robin (RR) and Shortest-Job-First (SJF) schedulers with vibrant unpredictable quantity job. The algorithms suggested relying generally on two fundamental keys the first, to have a complex task of balancing the waiting period between tasks with small sized tasks as well as lengthy tasks Whereas lengthy tasks includes forming two sub-queues from the ready queue, Q1 representing short and Q1 representing lengthy task. Q1 and Q2 are got assigned tasks towards resources. For the intent of the assessment, During the 3.0.3 version simulation of the environment toolkit of Cloud Sim, triple different data sets were used by triple algorithms along various scheduling tasks of SJF, RR and Time Slice Priority Based RR (TSPBRR) Experimental findings and tests showed the superiority of the proposed state-of-the-art algorithm in lowering waiting time, response time and partial starvation of the algorithm.

The issues are get resolved by Zuo, et al [17] by algorithm of improved ant colony. Determination and access input on quality and budget costs, two constraint functions have been used. The primary 2 functions with conditions for producing timeliness, response to feedback, change the quality of the solution to acquire best result. Configuration of experimental validation is done for assessing the performance of this method utilizing 4 parameters: 1) span making; 2) expenditure; 3) limit of violation rate; and 4) utilization of resource. Findings indicate so multi-objective optimization approach is better depending these four metrics than other similar techniques, particularly since in the best case scenario is increased by 56.6%.

A new evolutionary algorithm has been suggested by Navimipour&Mailani [18], calling CSA to plan cloud computing tasks. The CSA algorithm is predicated in conjunction with the behaviour of flight of some birds and fruit flies on the compulsory breed parasite behaviour of some cuckoo species. The outcomes of the simulation presented that the speediness and exposure from algorithm are maximum when the Pa value is low.

S. No	Author name	Method	Merits	Demerits
1.	Karthick et al [2014]	Multi Queue Scheduling.	Achieves the optimum cloud scheduling.	It consumes more time for scheduling .
2.	Biswas, et al [2017]	Heuristic-based task scheduling.	Improves the makespan, complexity and average processor utilization.	Energy consumption is very high.
3.	Jaspreet Singh	Smarter MQS model.	Time taken to	Need to improve merging

	et al [2014]		complete and expenditure are decreased	pattern. Higher energy consumption.
4.	Zhang and Zhou [2017]	Two-stage strategy.	Improves cloud task scheduling.	Time consuming nature.
5.	Sumit Arora and Sami Anand [15]	scheduling on improvement.	Taking less processing time.	High average waiting time.
6.	Elmougy, et al[16]	Combining Shortest-Job-First (SJF) and Round Robin (RR).	Reducing waiting time, response time.	Need to use other task quantum calculation on methodologies.
7.	Zuo, et al [2015]	Improved ant colony algorithm	Improves Resource utilization.	Time consuming nature.
8.	Navimipour and Milani [2015]	CSA	the speediness and exposure from algorithm are maximum	Higher energy consumption.

3. PROPOSED METHODOLOGY

In this part, describes in detail the proposed model based job scheduling is described detail. This model generates multi queuing depending optimization from swarm particle for scheduling jobs between many multi queues, shorter jobs first and least algorithms fit best. To stop starvation for work, SJF-ELM is suggested.

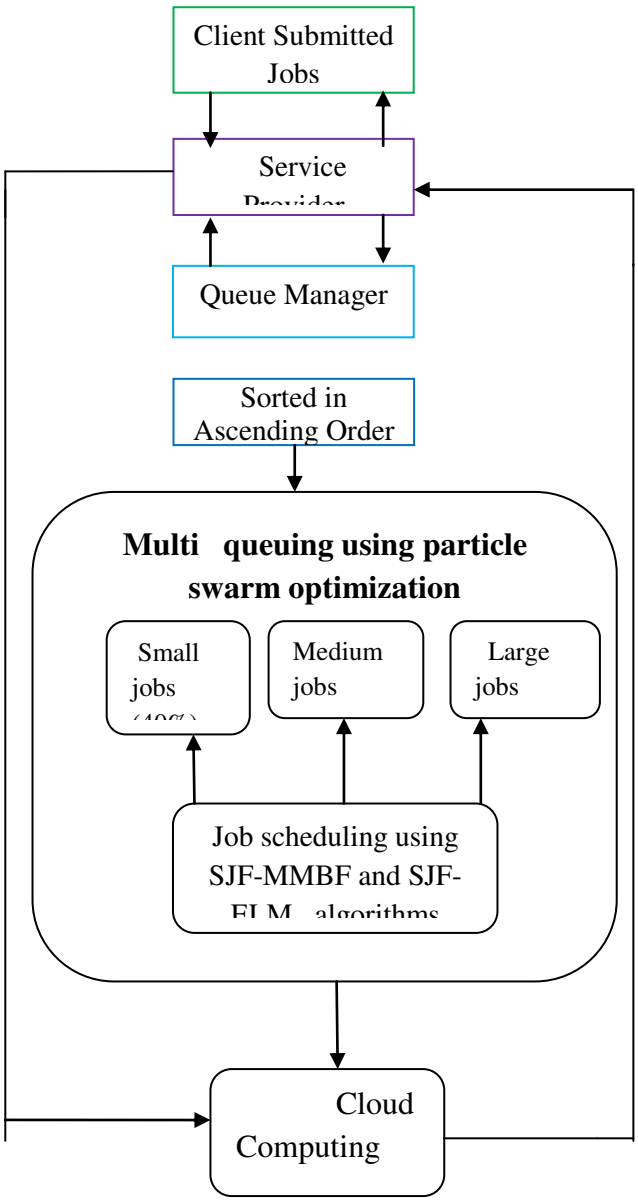

Figure: 1. Overall architecture of the proposed model

The first jobs provided by clients were listed in the order of burst in the given diagram. The works are segregated as multi queues with the optimization of particle swarm due to bursting times such as small, middle and high. SJF- MMBF as well as SJF-ELM algorithms were tested for resource use by the cloud in multi queues preference (scheduling).The queue manager is critical in distributing network resources. It deals with the use of all network resources. Through managing force one of the approached scheduler as well as its disposal, the queue manager searches from time to time who currently runs jobs. It manages the performance of the tasks the queue manager is gathering. Specific queues are based on the results of burst time in ascending order.

Different queues are made in ascending order on the basis of burst time.

1.Storing of 40% of burst time is done in case of small waiting jobs.

2.Storing of further 40% are burst time done in medium waiting jobs.

3.Storing of 20% are burst time in long waiting jobs.

3.1.Multi Queuing Using Particle Swarm Optimization Algorithm

This part tasks are queuing using algorithm of particle swarm optimization.

3.1.1. Particle swarm optimization:

PSO are algorithms for optimization depends bird flocks' social behaviour. Swarms are formed from individual particles in PSO of search process, a population-based. **Particle swarm optimization** is used to differentiate in several queues certain tasks or jobs. In case of optimization problem in PSO, consider particle in the swarm as candidate. Every particle in a PSO system is "flown" via the multidimensional research area, adapting their standing inside space search for individuals experience and the neighbouring particles. (jobs).

A particle thus gets the best position to position to an optimal solution, as well as that of its neighbours.The effect would be that particles "fly" at least thus seeking the best solutions in a wide area. The output of every elements are regulated by a function of fitness from predefined, that wraps features from issues of optimisation.

Each particle (job) S maintains the following information:

X_i The present position of the particle(job);

V_i The current velocity of the particle(job);

$\mathbf{y_i}$ The personal best position of the particle(job)

The velocity and position of the particle(job) *i* are considered by

$$V_{id}^{t+!}=\omega_{id}^{t+1} + C_l * r_{li} * (p_{id} - x_{id}^t) + c_2 * r_{2i} * (p_{gd} - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + V_{id}^{t+1} \quad (2)$$

Where *t* signifies the *t* th iteration in the method and *d* represents the *d*th dimension in the search space. *w* is inertia weight and c_1c$_1$ and c_2c$_2$are acceleration constants. r_{1i} nd r_{2i} are random values uniformly distributed. p_{id} And p_{gd} represent the elements of $pbest$ and $gbest$in the *d*th dimension [19,20].

The position and velocity standards for each job become modernized regularly to look for the correct set of jobs till the stop criterion is reached, which might be a maximum of iterations or a sufficient average completion period of fitness value).The applied PSO algorithm is termed.

PSO Algorithm

Step1 swarm (job initialization) Randomly initialize the position and velocity of each particle.

Step2 particle (job) fitness (average completion time) evaluation*)*

if fitness of $xi > pbesti$

$pbesti=xi$

if fitness of $pbesti > gbesti$

$gbesti=pbesti$

Step3. Update the velocity of particle (job)*i*

$V_{id}^{t+!}=\omega_{id}^{t+1} + C_l * r_{li} * (p_{id} - x_{id}^t) + c_2 * r_{2i} * (p_{gd} - x_{id}^t)$

Update the position of particle (job) *i* $x_{id}^{t+1} = x_{id}^t + V_{id}^{t+1}$.

Step4. If stopping criterion is not met, continue Steps 2 and 3.

Step5. Return $gbest$ and its *fitness values (average completion time).*

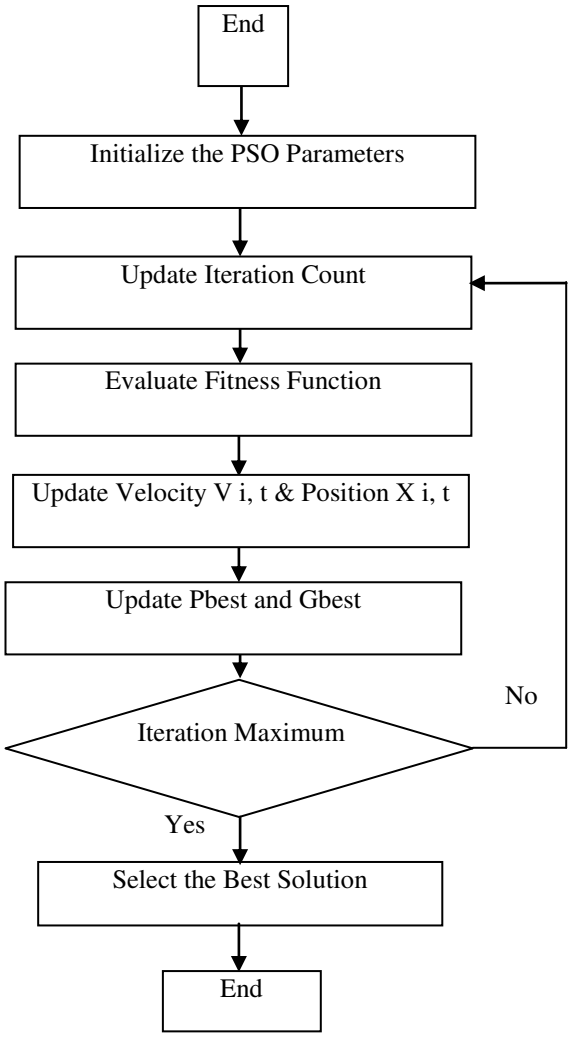

Figure: 2. Flow chart for PSO

Through shortest-job-first buffering and min-min best-fit scheduling algorithms for resource use, jobs are distributed in several queues.

3.2. Scheduling Policy For Min-Min Best Fit and Joint Shortest-Job-First Buffering

Due to strong heterogeneity and dynamism in the task, predict future resource requirements would have been challenging and very expensive to accurately predict traffic features (for example predicted level of arrival as well as medium size of job). [21, 22].

3.2.1. Scheduling Policy for Min-Min Best Fit

Conceptually, when selecting an action that requires the most resources between all the acts which can be chosen in each decision epoch,the median queueing delay would then be reduced to shorten the average completion time of the job.To evaluate a sequential Nai*Originally, the optimal choice was developed for scheduling jobs of single-resource like requirement of recall otherwise st orage, thus, introduce the first algorithm, named MMBF.The core idea would be to locate the lowest resource with multiple free sections, sufficient to meet an application, to eliminate a waste of free resource. In MMBF, instead, the action which minimizes existing multi-resources becomes described as a best choice action.In MMBF, instead, the action which minimizes existing multi-resources becomes described as best choice action. The facts concerning MMBF is represented as [23,24]. In criteria of decision making in scheduling, k be the resource of normalized remaining signified in $\Delta_k(a_t)$.

$N_{at} \sqsubseteq N_{At \times V}, a_t \in \{1, \ldots\ldots, A_t\}$which is derived as

$$\triangle_K (a_t) = \frac{C_k - \sum_{v=1}^{V} N(a_t, V) R_{VK}}{C_K} \quad (3)$$

Let $\triangle (a_t)$ denote the minimum value of $\Delta_K(a_t)$ for k = 1,……K under action at. That is,

$$\Delta(a_t) = \min_{k \in K} \Delta_K(a_t) \quad (4)$$

Then, under the MMBF scheduling policy, the solution a_t^* at time t is the one that satisfies

$$a_t^* = \arg \min_{a_{t=1,\ldots A_t}} \Delta(a_t)$$

(5)

3.2.2. SJF-BASED INTRA- QUEUE BUFFERING

In this part, as MMBF is not ideal for delays, concentrate on the prolonged problem of selecting jobs of the same kind while measuring the number of jobs to obtain the optimal delays.

SJF is an efficient without pre-emptive scheduling scheme for achieving average job completion time optimization in a system consisting of a single resource. In SJF, a system schedules the shortest job first, then the next shortest, and so on. Since jobs requesting the same VM type require the same amount of multi-resources, it is possible to buffer them in the same queue and apply SJF to determine their queueing positions such that their scheduling priorities are determined intra queue to improve the performance in terms of the average job completion time. Therefore, the SJF buffering policy is designed to address the problem of how to select jobs of the same type for scheduling.

Algorithm 1 SJF buffering

While a type-v job f arrives in time interval [t; t + 1), **do**

1. Find a position j' in type-v queue that gratifies

Svj'sfsvj'+1 in type –v queue

(6)

1. Insert job f into type-v queue in a position after j' and let

{QVt+1=Qvt+1Wvt+1= wvt+sf

(7)

End while

where S_f is the new job length f and svj is the jth job length in the type-v queue.

3.2.3. Finally, combine the SJF buffering and MMBF scheduling policies to form the first scheme, called SJF- MMBF.

1) Buffering Algorithm (SJF Buffering): The many type-v jobs that arrived in time intervals [t;t + 1) are buffered in the vth queue as per the buffering policy as well-defined in Algorithm 1, for v V.
2) Scheduling Algorithm (MMBF Scheduling): In result epoch t, do
a) Estimation the resource array N A_tv.
b) Select action at*At such that Nal*NAtV is deterined.
3) Scheduling Process: In a time interval [t; t + 1), Nvp(t) type-v jobs remain to be served, and Nat *,vNvpt type-v jobs are de-queued from the v^{th} queue in an HOL way and start to be aided for v V. The number of jobs to come in the queue and the acquisitive workload requirement are efficient, respectively, as follows.

In existing work Although SJF is efficient in terms of the average job completion time optimization, it has the potential for job starvation under the SJF-MMBF scheme for VM scheduling in a cloud system. This is because the behavior of MMBF, which always selects an action minimizing the remaining resources, will be detrimental to some types of jobs. However it is giving better throughput results.

3.2.4. SJF-RL

In existing work the SJF-RL scheme has achieved its goal of delay-optimal scheduling of VMs by providing low delay of performance in terms average job completion time but lower throughput performance in terms of job hosting rate in a queueing cloud system with workloads range both from light-loaded to heavy-loaded and from slightly dynamic to highly dynamic. Also SJF-RL has a problem of premature convergences. By which it may affect in local search capability to select optimum jobs, so that the algorithm will trap in to local optima. So to avoid all this issues in

this work proposed Extreme Learning Machine instead of RL it will be better in terms of both delay and throughtput.

3.2.5 Extreme Learning Machine algorithm

SJE-ELM is used here to optimize the long-term average job completion time g(∗). SJF-ELM scheme, the SJF discipline is used to buffer arriving jobs and the ELM-based scheduling policy is designed to determine a sequential a_t^* and N a_t^* to minimize the long-term g(∗).

N At ×V array of resources do abstract in capability of resource as well as VM's several requirements from pools of resource. The objectives are to diminish the time taken and expenditure fully for process of training in iteration and performance in generalisation by ELM. As single -hidden-layer feed forward neural networks (SLFNs), the ELM structure includes input layer, hidden layer, and output layer. The best solution is determined easily from existing training parametric of all networks from conventional neural network learning algorithms. the ELM only sets the number of hidden neurons of the network, randomizes the weights between the input layer and the hidden layer as well as the bias of the hidden neurons in the algorithm execution process, calculates the hidden layer output matrix and finally obtains the weight between the hidden layer and the output layer by using the Moore-Penrose pseudo inverse under the criterion of least-squares method[25,26,27].

The structure of ELM is so simple and holds precise parametric in computation it retains its benefits in speedy process. The ELM structure is figured below.

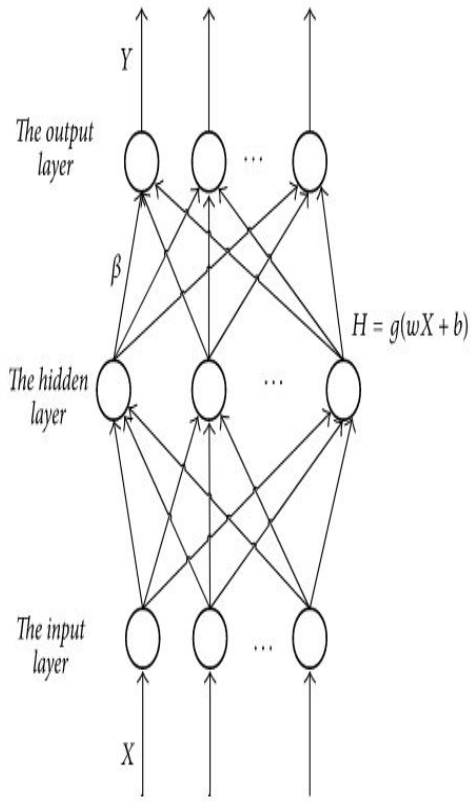

Figure 3 : ELM Structure

Figure 3 is the network structure of the intense learning system that involves layers of neurons as input, hidden and output. First, investigate training sample and there is an input jobs from a different queues and a desired matrix included from sample considered for training, to sate matrix as

For N arbitrary jobs from multiple queue like Type 1, Type 2 and Type 3 jobs $(X_i t_i) R^d \times R^m$ suppose that the SLFNs construct N nodes of hidden and g (x) activation function like function for sigmoid, that stated as

$$i=1Ni\ gwi.xj+bi=tj \qquad (8)$$

wi is the weights vector used to interconnect i^{th} nodes of hidden as well as input with bias bi for i^{th} nodes of hidden as well as input. The vector for i output weights vector used to interconnect i^{th} nodes of hidden as well as output nodes. $w_i.x_j$ signifies $w_i.x_i$ inner product.

$$X=x11x12\ldots\ldots x1Qx21x22\ldots\ldots x2Q...xn1xn2 \ldots\ldots xnQ \qquad \textbf{(9)}$$

Y=y11y12..ymQy21y22.ymQ...ym1ym2..ymQ
(10)

X and Y, the parameters are space matrix of an input action dimensionally and scheduled matrix on output decision. Layers of both input and hidden are weighted by ELM as

=1112..1n2122..2n...l1l2..ln
(11)

Next, Layers of both output and hidden are weighted by ELM as

=1112.1m2122.2m...l1l2lm
(12)

Next in step 4, neurons of hidden layers are set in bias randomly by ELM as

B=b1b2.bn T
(13)

Next in step 5, the function of network activation is selected by ELM. The output matrix job scheduling can be represented as follows:

T=t1,t2,.tQ mxQ
(14)

Each output matrix in column vector are represented by tj=t1jt2j. ...tmj=i=1li1gixj+bi i=1li2gwixj+bi ....i=1li mgixj+bjj=1,2,3..,Q. **(15)**

Next in step 6, (14) and (15) are computed and achieve

Hβ=T' (16)

layer of hidden and T' for output transposition. Using the least square method of measuring the weight matrix values of the minimum error to achieve a unique solution [15, 16].

=H + **T'.**
(17)

To improve network simplification and stabilize the output, add a regularization concept.

In this criteria, neurons of hidden layer are less than training samples and represented as

=I+HTH -1HTT'.
(18)

In this criteria, neurons of hidden layer are greater than training samples and represented as

=HTI+HHT -1T'
(19)

Feature		planes
State s	serve up of huge jobs as , N_v^p	N V
	huge type-1 jobs in queue to serve as Q_v	(N + 1) V
	requirement of Workload as W_v	(N + 1) V
Action a	at action a and configuration of VM is N_a	N V

Table: 1. Job scheduling with corresponding features on State-action

3.2.6. SJF- ELM

1) Algorithm of Buffering (Buffering in SJF): A range of [t; t + 1] are buffering time of sort of v jobs accordingly on queue as well as buffering policy. It is deployed in Algorithm SJF as SJF-ELM, v ∈ V.

2) Scheduling Algorithm (ELM Scheduling): In t_j **of** decision epoch, perform

- $S_t \leftarrow \left(N_v^p(t)\right), Q_v(t), W_v(t)$ State is intellect.

- Huge X actions are calculated feasibly and N $A_s \times$ v as array resource.

3) Scheduling Process:

a) Scheduling: jobs of $N_v^p(t)$ type 1 are being supported in queue along $v \epsilon V$, as well as de-queue $\left(N\left(a_t^*, v\right) - N_v^p(t)\right)$ type-1 jobs from the 1th queue and start serving the function. queue holding plenty of waiting jobs and update the required workload in accumulation are from

$$\begin{cases} Q_v(t+1) = Q_V(T) - \left(N\left(a_t^*, v\right) - N_v^p(t)\right) \\ w_v(t+1) = W_v(t) - N\left(a_t^*, v\right) \end{cases}$$

b) Computation of time for job completion in time-averaged as follows

$$E\left[\tilde{T}(t)\right] = \sum_{v=1}^{V} \alpha E\left(\hat{T}_v(t-1)\right) + (1-\alpha) T_v(t)$$

Where

$\alpha \in (0, 1)$ is a weight parameter.

c) If $E\left[\widetilde{T(t)}\right] T_j > E[T^*]$, $\omega *$,vector of parameter are updated

d) The final number of arrived traffic T gets stored as

$$\{j_v(t - T + 1), \ldots . j_v(t)\}$$ [27].

4. RESULT AND DISCUSSION

This sector discusses the experimental results of suggested model. The model is implemented using JAVA. This model compared with the proposed Multi Queuing SJF-ELM (MQ-SJF-ELM) and the existing single queuing SJF-RL (SQ-SJF- RL), Single queuing SJF-MMBF(SQ- SJF-MMBF) are compared in terms of Throughput, Delay and cost.

Table:1. Performance comparison results

METRICS	METHODS		
	SQ-SJF-RL	SQ-SJF-MMBF	MQ-SJF-ELM
Throughput	0.377	0.677	0.800
Delay (ms)	1170	610	460
cost($)	137	114	52

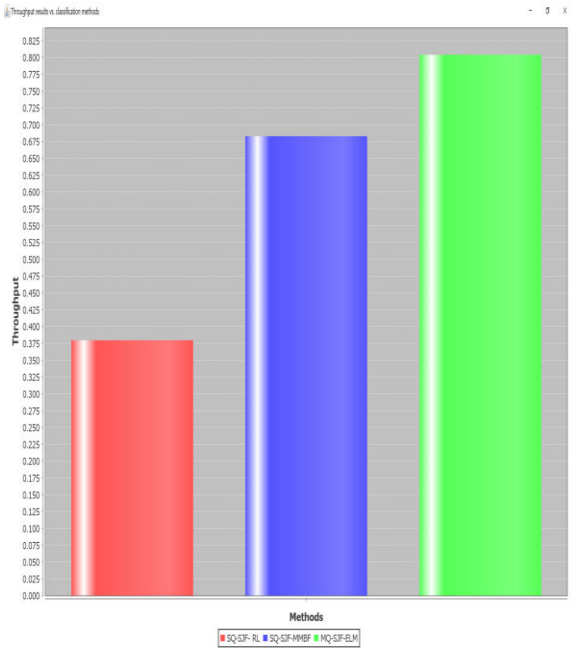

Figure: 4. Classification methods Vs results of Throughput

Figure 4 depicts the throughput performance comparison outcomes of existing SQ-SJF- RL, SQ- SJF-MMBF method and proposed MQ-SJF-ELM method. In the above graph, X Axis with Scheduling methods and Y Axis with throughput values are taken in. From outcome, it confirmed that proposed MQ-SJF-ELM model generated superior throughput results of 0.800 whereas existing SJF- RL, SQ-SJF-MMBF method gives only 0.377 and 0.677correspondingly.

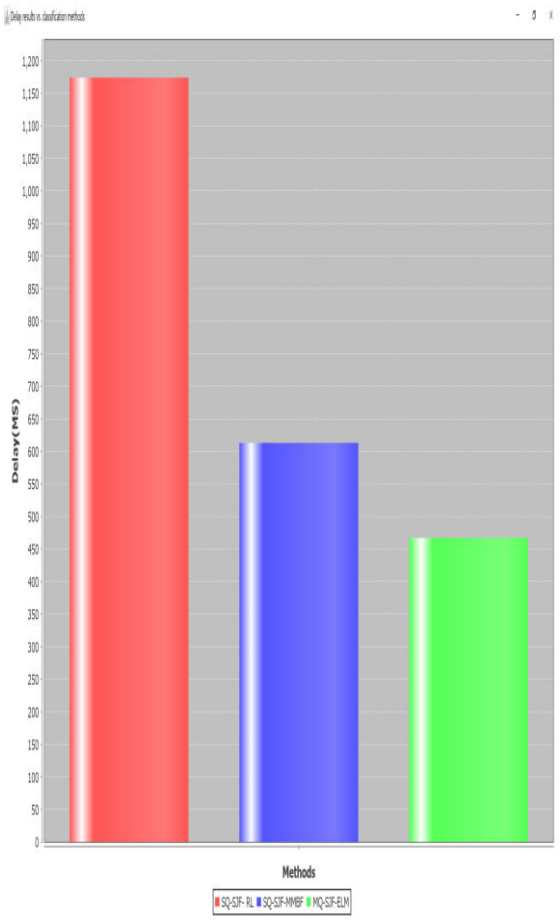

Figure: 5. Delay results vs. Classification methods

Job scheduling result is shown in figure 5 in terms of delay for the existing SQ-SJF- RL, SQ- SJF-MMBF method and proposed MQ-SJF-ELM method. From outcome, it confirmed that proposed MQ-SJF-ELM model generated superior Delay results of 460 (ms)whereas existing SJF- RL, SQ- SJF-MMBF method gives only 1170(ms) and 610(ms) correspondingly.

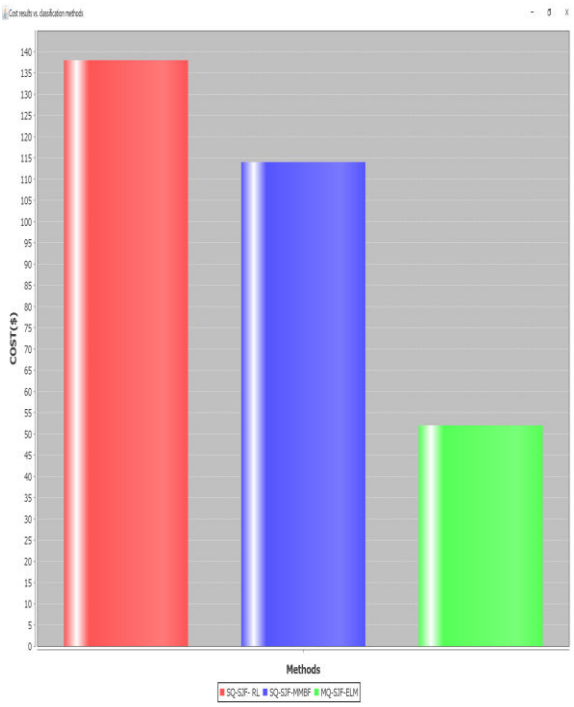

Figure:6. Cost results vs. classification methods

Overall result comparison of the proposed model based job scheduling is shown the above figure for Cost metric with the existing SQ-SJF- RL, SQ- SJF-MMBF method and proposed MQ-SJF-ELM methods and proposed MQ-SJF-ELM method. From outcome, it confirmed that proposed MQ-SJF-ELM model generated lesser cost expensive results of 52$ for scheduling whereas existing SQ-SJF- RL, SQ-SJF-MMBF method gives only 137$ and 114$ correspondingly.

5. CONCLUSIONS AND FUTURE ENHANCEMENT

Algorithm of PSO is being utilised in this proposal that splits the ready queue into certain longer queues as of algorithm in multi-level queue scheduling. Methods are automatically allocated to one queue, based on certain process properties including size of memory, priority in process or process type. It controls both algorithms of SJF as well as Min-Min-Best Fit algorithms for planning.

A further scheme, one that integrates the SJF buffering with the Extreme Learning

Machine (ELM)-based scheduling algorithms, i.e., SJF-ELM, also implemented in diminishing efficient for working demand in SJF-MMBF. Therefore, it resulted from the findings that the current proposal provides better output performance. The scheduling algorithm for future hybrid tasks may also be created. And there will also be called the impact of precedence among tasks and load balancing.

REFERENCES:

1. Shorgin, S., Pechinkin, A., Samouylov, K., Gaidamaka, Y., Sopin, E. and Mokrov, E., 2014, October. Queuing systems with multiple queues and batch arrivals for cloud computing system performance analysis. In *2014 International Science and Technology Conference (Modern Networking Technologies)(MoNeTeC)* (pp. 1-4). IEEE.
2. Eisa, M., Esedimy, E.I. and Rashad, M.Z., 2014. Enhancing cloud computing scheduling based on queuing models. *International Journal of Computer Applications*, *85*(2).
3. Singh, I. and Arora, A., 2015. Fuzzy Based Improved Multi Queue Job Scheduling For Cloud Computing. *International Journal of Advanced Research in Computer Science*, *6*(5).
4. Singh, S. and Chana, I., 2016. A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of grid computing*, *14*(2), pp.217-264.
5. Nan, X., He, Y. and Guan, L., 2014. Queueing model based resource optimization for multimedia cloud. *Journal of Visual Communication and Image Representation*, *25*(5), pp.928-942.
6. Sowjanya, T.S., Praveen, D., Satish, K. and Rahiman, A., 2011. The Queueing Theory in Cloud Computing to Reduce the Waiting Time. *International Journal of Computer Science Engineering & Technology*, *1*(3).
7. Bhoi, U. and Ramanuj, P.N., 2013. Enhanced max-min task scheduling algorithm in cloud computing. *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, *2*(4), pp.259-264.
8. LD, D.B. and Krishna, P.V., 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, *13*(5), pp.2292-2303.
9. Agarwal, D. and Jain, S., 2014. Efficient optimal algorithm of task scheduling in cloud computing environment. *arXiv preprint arXiv:1404.2076*.
10. Salot, P., 2013. A survey of various scheduling algorithm in cloud computing environment. *International Journal of Research in Engineering and Technology*, *2*(2), pp.131-135.
11. Karthick, A.V., Ramaraj, E. and Subramanian, R.G., 2014, February. An efficient multi queue job scheduling for cloud computing. In 2014 World Congress on Computing and Communication Technologies (pp. 164-166). IEEE.
12. Biswas, T., Kuila, P. and Ray, A.K., 2017, January. Multi-level queue for task scheduling in heterogeneous distributed computing system. In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)(pp. 1-6). IEEE.
13. Jaspreet Singh and Deepali Gupta. An Smarter Multi Queue Job Scheduling Policy for Cloud Computing. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 9 (2017) pp. 1929-1934.
14. Zhang, P. and Zhou, M., 2017. Dynamic cloud task scheduling based on a two-stage strategy. IEEE Transactions on Automation Science and Engineering, 15(2), pp.772-783.
15. Sumit Arora and Sami Anand.Improved Task Scheduling Algorithm in Cloud Environment. International Journal of

Computer Applications (0975 – 8887). Volume 96– No.3, June 2014

16. Elmougy, S., Sarhan, S. and Joundy, M., 2017. A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique. Journal of Cloud Computing, 6(1), p.12.
17. Zuo, L., Shu, L., Dong, S., Zhu, C. and Hara, T., 2015. A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. Ieee Access, 3, pp.2687-2699.
18. Navimipour, N.J. and Milani, F.S., 2015. Task scheduling in the cloud computing based on the cuckoo search algorithm. International Journal of Modeling and Optimization, 5(1), p.44.
19. Trelea, I.C., 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information processing letters, *85*(6), pp.317-325.
20. Chander, A., Chatterjee, A. and Siarry, P., 2011. A new social and momentum component adaptive PSO algorithm for image segmentation. *Expert Systems with Applications*, *38*(5), pp.4998-5004.
21. Guo, M., Guan, Q. and Ke, W., 2018. Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload. *IEEE Access*, *6*, pp.15178-15191.
22. Kaur, R. and Kinger, S., 2014. Analysis of job scheduling algorithms in cloud computing. *International Journal of Computer Trends and Technology (IJCTT)*, *9*(7), pp.379-386.
23. Ru, J. and Keung, J., 2013, June. An empirical investigation on the simulation of priority and shortest-job-first scheduling for cloud-based software systems. In *2013 22nd Australian Software Engineering Conference* (pp. 78-87). IEEE.
24. Salot, P., 2013. A survey of various scheduling algorithm in cloud computing environment. *International Journal of Research in Engineering and Technology*, *2*(2), pp.131-135.
25. Huang, G.B., Zhu, Q.Y. and Siew, C.K., 2004. Extreme learning machine: a new learning scheme of feed forward neural networks. *Neural networks*, *2*, pp.985-990.
26. Li, M.B., Huang, G.B., Saratchandran, P. and Sundararajan, N., 2005. Fully complex extreme learning machine. *Neurocomputing*, *68*, pp.306-314.
27. Guo, M., Guan, Q. and Ke, W., 2018. Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload. *IEEE Access*, *6*, pp.15178-15191.