TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

Resource Scheduling Policy using Population based Metaheuristic Search along with Firefly Caching Policy in Cloud Environment

Mamatha C M, Research Scholar, Dept. of CSE, NCET, Bengaluru, Karnataka, India
GururajMurthugudde, Prof. and Head, Dept. of ISE, NCET, Bengaluru, Karnataka, India

Article Info
Volume 82
Page Number: 2871 - 2881
Publication Issue:
January-February 2020

Abstract:
Rapid development in the field of network technology, the cloud computing receives great attention among small scale industries to large industries because of its decentralized environment. The cloud environment comprised of distributed resources in a dynamic fashion, so this necessitates the need for developing optimal scheduling in cloud environment with the satisfaction of QoS necessitated by the cloud consumer with the maximum profit to cloud providers. But the presence of impreciseness while scheduling cloud resources is the challenging issue of traditional scheduling policies. The main objective of this paper is to treat the vagueness in scheduling of cloud resources by designing intuitionistic fuzzy with population-basedmetaheuristic search which works based on the inspiration of teaching and learning process. This proposed work represents the parameters involved in resource scheduling by the means of intuitionistic fuzzy representation with the aim of reducing the response and execution time with maximize throughput which favors the profit of cloud service providers. Once the job is passed to the local schedulers the optimization is also fine grained at the local host level by utilizing cache segmentation with the knowledge of firefly algorithm. The simulation results showed that this developed model provides potential resource scheduling in cloud computing by treating hesitancy in adversarial situation.

Article History
Article Received*: 14 March 2019*
Revised*: 27 May 2019*
Accepted*: 16 October 2019*
Publication*: 18 January 2020*

Keywords:*Cloud computing, resource scheduling, impreciseness, uncertainty, intuitionistic fuzzy, population-based metaheuristic search, teaching learning process, firefly, cache*

I. INTRODUCTION

In recent years one of the popular and best technology which enables both start up or small-scale organization that use the resources by cloud computing with the concept of pay as you use. Due to the exponential advancement in field of cloud paradigm there is an increasing demand for numerous services from varied user community. Cloud service providers assists in management and allocation of different resources connected in cloud depending on the customers need[1]. Management of resources in cloud is a roof activity which covers various phases of resources and job loads form job submission to job execution. There are two phases in cloud they are provisioning of resource and scheduling resources.

The phase which involves in discovering passable resources for a given job load depending on the quality of service requirements defined by cloud clients is known as resource provisioning. Whereas, scheduling resources involves in mapping and execution of cloud clients job request rely on chosen resources by means of provision resources as depicted in the figure 1.

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

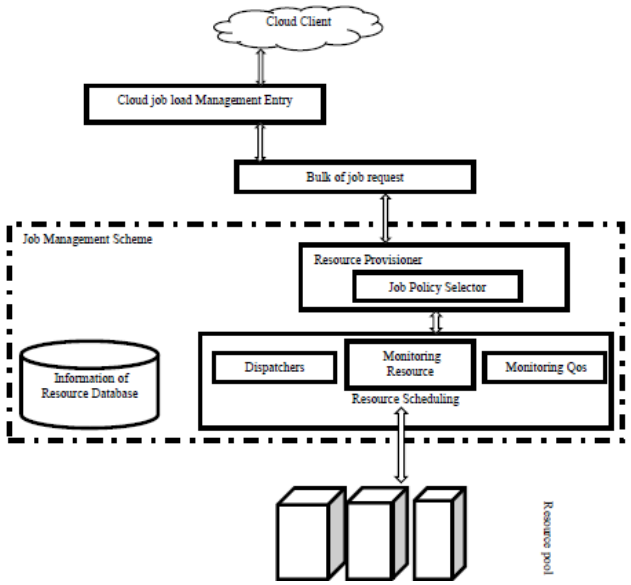


Figure 1 Basic Flow of Resource Scheduling in Cloud Paradigm

Clients initiate the process by submitting request for job request execution by providing details of the job load. Depending on these details resource provisioner discovers suitable resources for a specified workload and identifies the availability of resource provisioning with Q0S necessities [2]. After gathering information, the provisioner passes the request to the resource scheduler that is involved in scheduling resources subsequently to positive resources provisioning. Additional responsibilities of resource provisioner is extra resources are released to resource pool, it maintains information about provisioned resources, performance monitoring is done by adding or removing resources

Following resource provisioning, scheduling of resource is performed. The provisioned resources are kept in resource queue and other are placed in pool of resource [3]. Assigned jobs are process in job load queue. During this stage, agents involved in scheduling maps resources provisioned to given job loads, job loads are executed and after job load execution is completed the acquired resources are released to the pool. But the major challenge arises depending on the quality of service requirements to accomplish the task of resource scheduling for appropriate job or work load is a toughest challenge faced in cloud computing paradigm.

An important and most focusing area of research in cloud is scheduling of resources with the ability to full fill the requirements of QoS, this is because of more resource cost and execution time. There are different categories of resource scheduling algorithms are in existence depending on scheduling criteria and parameters involved in scheduling process. This paper also focuses on the caching policies to extend the optimization in host level to consume the cost of processing the job request with better utilization of resources in cloud.

Problem Definition

This paper work focuses on effective resource scheduling which reducing the cost of execution, time, consumption of energy and security. To maintain the quality of service the resource providers reject the resource requests in unpredictable or uncertain environment. Most of the cloud-based resource scheduling are not focusing on impreciseness and heterogeneity of resources and they cannot be treated effectively using traditional resource scheduling algorithms. To overwhelm these constraints in case of uncertainty for solving resource scheduling the present research work focused on devising uncertainty handling population based metaheuristic searching model for optimized resource scheduling with less tuning parameters in cloud paradigm. This work also incorporates the optimization of cache replacement strategy because these policies plays a vital role in refining Caching algorithms which also improves the resource consumption in cloud computing.

II. RELATED WORKS

Pradeep and Jacob [4] reported a comparison analysis of quality of service-based scheduling scheme with different types of task scheduling schemes with their advantages and advantages. The optimal

Time algorithm is developed by Raju et al [5] which aims to increase the productivity in task performance. It uses the concept of map reduce

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

based task scheduling scheme to minimize the makespan of workloads.

Ge et al [6] developed a genetic algorithm based task scheduling scheme in cloud environment. They used the whole group of tasks in the job queue and scheduling of resources is done by considering the reduction of make span to achieve balanced load among virtual machine.

Jang et al [7] devised a genetic algorithm for task scheduling which focuses on merits of QoS and the cost benefits to providers in cloud computing

QiangGuo et al [8] developed an ant colony algorithm for scheduling the task in cloud computing. The algorithm utilized the parameters such as pheromone collection and updating of the fitness function to discover the best scheduling scheme. Their work aims to produce better makespan, less the cost and maintain balanced load in cloud environment.

Zuoet al.[9] anticipated an multi-objective scheduling scheme using ant colony model. They used the budget constraint to analyze the previous feedback on the quality of service. This characteristic avoids the pitfall of local optima problem in ACO and then from the negative feedback.

Hongbo Liu et al [10] introduced a particle swarm optimization-based scheduling job. The performance of PSO is compared with simulated annealing and genetic algorithm/

Srinivasarao and RaveendranBabu [11] designed an evolutionary algorithm-based resource scheduling using genetic algorithm they proved that the performance of this model greatly suits than the batch queuing heuristic with the control parameters such as mutation and crossover rate that influence the impact of the effective solution.

Juan et al.[12] designed a PSO based task scheduling scheme to overwhelm the issues of cloud computing by using cost vector model. It measures the scheduling schemes by means of cost and developed the model based on the input task and their needed QoS parameters. Though it is effective it leads to more complexity.

Krishnasamy [13] introduced a hybrid PSO based job scheduling algorithm whose aim is to reduce the average operation time with limited resource utilization.

Alkayalet al.[3] in their work also used PSO with multi objective task assignment with ranking strategy. The tasks are assigned to the virtual machine depending on the rank. Its performance resulted in less waiting time and system performance is high.

Rao et al [15] devised Teaching-Learning-Based Optimization (TLBO) algorithm which works under two phases with the phenomena of teaching – learning environment the resources are scheduled to produce better result

Problem Definition

Thought there are many works in existence for performing resource scheduling in cloud environment,the ultimate goal of scheduling the resources in the uncertain conditions are not greatly focused by many works. While working in Cloud Computing based resource scheduling the presence of impreciseness about the availability of resources are the toughest challenge and the main objective of this research work is to discover the optimal resources for scheduling the appropriate workloads on time in order to minimize the response and completion time to satisfy the QoS of Cloud consumers and maximize the profit of the cloud consumer. To accomplish the issue of hesitancy in mapping job work load to optimal resource scheduling. Using grade of membership and non-membership the vagueness of resource scheduling is handled. Using population basedmetaheuristic search model the process of teaching and learning is developed in this work to achieve the optimized resource scheduling in cloud computing.

The teaching–learning-based optimization algorithm

One of the interesting population-based metaheuristic methods which involves in optimization of engineering issues is teaching learning-based optimization (TLBO) [16]. This

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

algorithm was recently devised by the inspiration of real life behaviour of teacher and learner. This algorithm narrates how the influence of a teacher will reflect on the learner in a class. This model works in two different methods of learning they are teacher mode and learner mode. In teacher mode, through the teacher the learning process takes place, whereas in learner mode, by interacting with other learners learning process is achieved.

In TLBO, the population of learners are chosen and parameters to discover the objective function are considered as subject to be learned. In this work, the results gained by the learners are termed as fitness value for the optimized resource scheduling problem. The learner who gained the highest fitness value is considered to have better knowledge and designed as teacher. This process is repeated until it met the end condition such as optimization of the scheduling is achieved or number of iterations reach it maximum.

In teacher phase, the learner with overall best result is considered as smart learner and identified as the teacher ($BL_{teacher}$) for that iteration. The conventional TLBO influences to enhance other learners (L_i) by shifting mean value of overall leaners (L_{mean}) towards the $BL_{teacher}$as expressed in the following equation

$$L_{i.new} = L_{iold} + rnd^* (BL_{teacher} - TC_F * L_{mean}) \quad (1)$$

Where, rnd ranges among 0 and 1 and TC_F is a teaching factor,whose value is either 1 or 2.

TC_F is not a parameter it is randomly decided with equal probability as formulated

$$TCF = round\,[1+rnd(0,1)\{2-1\}]$$

During Learner phase, a learner$L_i$tries to rise the acquaintance by communicating among themselves and they communicate with other learners in a random manner denoted as L_j. IF the Learner L_j is more knowlegeable than L_i ,L_i is shifted towards $L_j$as depicted in calculation [2]

$$L_{i\,new} = L_i + rnd \times (L_j - L_i) \quad (2)$$

Or else, it is moved away from $L_j$as shown in the following calculation

$$L_{i\,new} = L_i + rnd \times (L_i - L_j) \quad (3)$$

The learners in both teacher and learner phases are updated if the newly ($L_{i.new}$) obtained solution gains more knowledge than the previous learner (L_i).

III. About Intuitionistic Fuzzy Logic

The grade of membership μ(y), y∈Y, alone is considered in fuzzy set which lacks to handle the issue of uncertainty in discovering optimal resources scheduling, where the optimization results in increasing the throughput of the resource which satisfies the QoS demands, decrease the response and execution time of the job incloud environment. Hence this paper realizes the importance of processing uncertainty in adversarial environment specifically in scheduling of resources in presence of vagueness in determining appropriate resources to be provisioned. Hence, this work used Intuitionistic fuzzy set (IFS) devised by Atanassov[14, 15], in which each parameter involved in resource scheduling are denoted in terms of

Membership grade μ(y), and non-membership grade ν(y). An intuitionistic fuzzy set D in Y, is inscribed as:

$$D=\{y, \mu_D(y), \nu_D(y) | \; y \in Y\}$$

Where both $\mu_D(y)$, $\nu_D(y)$ value ranges from 1 to 1 with the condition

$$0 \le \mu_D(y) + \nu_D(y) \le 1$$

Intuitionistic fuzzy introduces a new grade which handles the impreciseness of each element known as grade of hesitation signified as $\pi_D(y)$ which is considered as an important factor when there is a lack of knowledge in defining the hesitancy of an element and it is formulated as follows:

$$\pi_D(y) = 1 - \mu_D(y) - \nu_D(y); \; 0 \le \pi_D(y) \le 1$$

Owing to hesitation degree, the value lies between $[0 \le \mu_A(x), \mu_A(x) + \pi_A(x) \le 1]$. Generally, intuitionistic fuzzy interpretation includes three different process such as Intuitionistic fuzzification, Intuitionistic inference model and DeIntuitionisticfuzzification

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

which generates IFS rules are normal if-then rules which is signifiedas :

If T is low and U is high, then V is medium

Where, Tand U are input variables, Vis a resultant output.

IV. Proposed Methodology

Optimized Hesitancy Handling Resource Scheduling Policy using Population based Metaheuristic Search in Cloud Environment

In real time problem space like resource scheduling, teaching approach plays a serious role in the effectiveness of transmitting knowledge among teacher and Learner in the class. In some cases, it will be more appropriate to use the concept of teacher centralized approach, here the teacher plays the dominant role who give instructions directly to the leaners. Still, there are few cases were leaners are actively cooperative and engaged in discussion and perform learning among themselves. At the same time, unsuitable way of teach will leads to negative impact on the learning process.

In standard TLBO method, it forces all the learners to enter both teacher and learner phase. As a result, TLBO needs totally two function evaluations to compute the fitness of each learner solution generated in each iteration. This research work modifies the standard TBLO by introducing intuitionistic fuzziness to handle the real time resource scheduling which consist of vague and inconsistent parameters of resources.

Important terminologies

- Priority: Depending on the memory and processing time needed to complete a job the priority value is assigned to the concern job. The highest priority job is chosen and assigned to the respective virtual machine which fulfills QoS requirement is used for execution
- Submission time(sbt): It is termed as the time at which the job enters inside the ready queue. Job which needs less execution time will be assigned to the corresponding virtual machine.
- Time of burst (BST): the amount of time needed to execute the job on corresponding virtual machine is known as burst time. It is calculated as length of the job divided by the speed of the processor or virtual machine is known as burst time.
- Start Time (SttT) : Job arrived to the machine and ready for execution by virtual machine
- Completion time (CmpTt): Job operation completes execution in its assigned virtual machine is referred as completion time.
- Execution Time (ExT): Amount of time taken by an individual job for executing completely on respective virtual machine. Difference among job completion time and the start time of the job is formulated as

  ExT (i,j) = CmpTt(i,j) - SttT(i,j)

  Where, i represent number of virtual machines (VM_1,VM_2,VM_3,..,VM_m) and j represents number of jobs (job_1, job_2, job_3,...job_n)

  Turnaround time (Trnt): total amount of time send by a job in a virtual machine is known as turnaround time

  $$Trn(i,j)=CmpTt\ (i,j)-Sbt\ (i,j)$$

  Waiting Time(WtT): Waiting time can be obtained from subtracting burst time from turnaround time

  $$WtT\ (i,)=Trnt\ (i,j)-BST(i,j)$$
- Response Time(RsT): Time gap between job submission and first response received by the task is known as response time

  $$RsT(i,j)=sbt(i,j)-WtT(i,j)$$
- Throughput c(Thpt): Total number of jobs executed successful Sucs(N) with respective to the total amount of executing time (T)

  $$Thpt(i,j)=Sucs(N)\ /T$$

This proposed model considers the resource scheduling as multi objective constraint in aspect of utilizing the resource in an optimizer manner.

1. Decreasing the response time

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

Objf(F) $\propto \frac{1}{\mathbf{RsT(i,j)}}$

2. To decrease the execution time
   Objf(F) $\propto \frac{1}{\mathbf{ExT(i,j)}}$
3. To increase the throughput of the resources
4. Objf(F) $\propto \frac{1}{\mathbf{Thpt(i,j)}}$

Thus, final fitness equation is represented as

$$Fitness = W1\left(\sum \frac{1}{\mathbf{Rst(i,j)}}\right) + W2\left(\sum \frac{1}{\mathbf{ExT(i,j)}}\right) + W3\left(\sum \frac{1}{\mathbf{Thpt(i,j)}}\right)$$

Resource Scheduling using hesitancy handling resource scheduling policy using Population based Metaheuristic Search (HRSP-PBMS)

This proposed work devised a hesitancy handling resource scheduling policy using Population based Metaheuristic Search which integrates intuitionistic fuzzy with modified Teacher learning algorithm for the resource scheduling in cloud environment.

The procedure for implementing this developed model is as follows:

Steps

1. The parameters for resource scheduling are initialized such as number of jobs to be executed, task priority assignment, total number of virtual machines available
2. Determining design variables like number of jobs in priority queue and number of virtual machines allotted for job completion and maximum generation
3. Generation of Population
   - Population of size N is generated in a random manner. Here, number of learners is denoted as population size. Each learner L_{ijk}refers to membership grade and non membership grade of the jth resources with kth jobs andIntuitionistic fuzzy Matrix (IFM) as follows:

$$IFM(L_i) = \begin{bmatrix} U_{i,1,1} & U_{i,1,2} & \dots & U_{i,1,n} \\ U_{i,2,1} & U_{i,2,2} & \dots & U_{i,2,n} \\ \vdots & \vdots & \vdots & \vdots \\ U_{i,m,1} & U_{i,m,2} & \dots & U_{i,m,n} \end{bmatrix}$$ Each element in the IFM(L_i) corresponds to $U_{i,j,k}$ = <$\mu_R(R_j,J_k)$,$\nu_R(R_j,J_k)$>, j = (1..m), k = (1..n). μ_R refers to grade of membership and ν_R grade of non-membership that the cloud resource Rj would process the job J_k in the possible schedule result. $U_{ijk} \in (0,1)$,$j \in 1,2, \dots, m$, $k \in 1,2, \dots, n$

4. Determine the makespan values for each learner in the population.
   - Mean of makespan is calculate
   - Any one of them acts as Mean and the best solution is treated as teacher (with minimum makespan
   - M = the solution of the makespan which is very nearer to the mean (M_D)
5. By adding the difference of teacher and mean of the class to each learner in the present generation a new set of improved learners will be generated

$$L_{i.new} = L_{iold} + rnd^* (BL_{teacher} - TC_F * L_{mean})$$

where, T_F is the teaching factor with the value 1 or 2 and rndis the random number range from 0 to 1.

$$TC_F = round\ [1+rnd(0,1)\{2-1\}]$$

6. Update knowledge ofeach learner with the help of other learners (Learner Phase)

   ○ Choose two leanersL$_i$andL$_j$and the performance of the leaner L_i can be increased as shown

   if fitness(L_i)< fitness(L_j) then $L_{i\ new} = L_i + rnd \times (L_j - L_i)$

   elseif fitness(L_j)< fitness(L_i) then $L_{i\ new} = L_i + rnd \times (L_i - L_j)$

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

$$Fitness(Li) = W1\left(\sum \frac{1}{\mathbf{Rst(i,j)}}\right) + W2\left(\sum \frac{1}{\mathbf{ExT(i,j)}}\right) + W3\left(\sum \frac{1}{\mathbf{Thpt(i,j)}}\right)$$

7. Repeat from step 3 to step 6 till a difference among fittest individual best learner whose fitness value in any successive generates is less than 0.001 and the best learner is considered as best solution

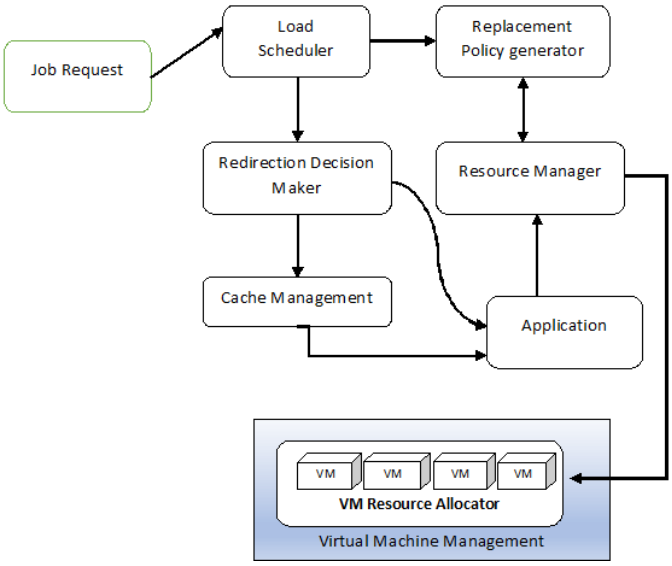
This algorithm consists of two phases teacher and learner. The first phase makes the learners, trained with the assistance of the teacher and with the gained knowledge of intuitionistic fuzzy the resource scheduling in cloud environment greatly overcomes the issue of uncertainty to execute the job request of cloud consumers.

Optimizing the performance of proxy servers using machine learning based cache algorithm

A general method in computer system design in caching technique. There are two popular network caching applications are used they are memory cache and content delivery network. The memory cache is deployed in dataset, which offloads traffic from backend database server and decreases application server demand latency. Whereas, Content delivery network influences edge server which be located near to the end users to offer internet access with low latency.

In cloud computing, web server shares its information to the local hosts within the cluster and performs distributed computing to complete their task quickly. Local servers highly depend on the server to accomplish a particular task. Frequent request from local host to server may increase the computation complex when a task is assigned to a specific host. Hence, this work highly focuses on effective utilization of cache memory within the local host by formulating cache evicting procedure as a reinforcement learning issue. The objects in cache are represented using cache state along with their request timestamps. During eviction process, the algorithm needs to choose one object or item depending on state of cache and evict it from the cache. Without proper knowledge if cache management is organized then it increases the risk of high miss rate. Thus, this paper includes the importance of cache eviction process along with optimized resource scheduling to overcome the hurdles faced by local host during replacement strategy of cloud cache or any other item.

Developing a cache replacement strategy in cloud paradigm aims to present a novel cloud cache replacement policy which optimizes the scalability, cloud data-out charge, overall responsiveness of data loading of cloud infrastructure.



Once a job request is received by the load scheduler, it evidently seizes the incoming traffic and accomplishes clustering of the requests based on their type of request, size of content and cost of processing. The load scheduler maintains the response time and hit rate of the complete job requests. With the gained information both replacement policy generator and redirection decision maker regulate the optimal selective caching.

Developing an optimized Replacement Policy Generator (RPG) is the core of this resource

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

scheduling scheme in cloud paradigm. The proposed RPG recognizes the set of requests that benefit most from caching. This generator uses request clusters as input and generates request redirection map as output to the Redirection Decision Maker (RDM). Depending on the requests selected for caching, RPG identifies the least size of the cache to fit these requests.

The request clustering, clusters the incoming job requests into two groups namely static and dynamic content, which are distinguished based on their Uniform Resource Indicator (URI). While deciding the caching policy, this model treats static and dynamic content differently using the size of the requested content.

Caching Policy generator using Firefly Algorithm

While the number of job request increases, the time required for computing the optimal caching policy also grows rapidly. It is very expensive to enumerate whole set of request mappings, specifically when the policy needs to be calculated online. Thus, this proposed work introducedfirefly based caching policy which accelerate the optimality and searching strategy.

In this firefly-based caching policy the algorithm arbitrarily produces N number of fireflies to form an initial population. The fireflies are assigned to each job request and the content size is considered to identify the appropriate cache space to accommodate the specific job request. The fireflies with high light intensity which is represented as fitness value discovers most optimal cache size which better suits for the request to be accomplished with minimum requirement such as processing cost. The fireflies with highest light intensity are considered to have higher chance to be selected.

The optimized firefly-based caching policy is the mapping of requests to the cache or to the multitier applications that incurs the least total processing cost for all request. If there are n job request clusters, each of which can be either processed in cache or out of the cache with the possibility of 2^n request mappings. The cost of request mapping is formulated as

$$Z = \sum_{k=1}^{p} [pc_k n_k h_k + apc_k n_k (1 - h_k) + \sum_{l=p+1}^{n} apc_l n_l]$$

Where pc_k is the processing costs of job request k on cache, apc_k is the processing cost on application, n_k is the number of request received in a specific Cloud server cluster k, h_k is the hit rate of cluster k.

Only those firelfies that have lower cost than all the fireflies is considered for caching in and the remaining requests are send to the applications directly. By repeating this metaheuristic search, a near optimal solution can be obtained for caching policy issue.

Every cahced request inhabits one cache block and the size of the block is set to the largest content. Hence, the cache size is identified by the content sizes and the number of requests stored in cache. Based on the request mapping, this proposed method determines the number of requests going to the cache server. The model recieves the requested content sizes of different job requests from the load scheduler. Next, the cache size is computed as

$$\text{Size}_{ch} = \text{maximum}(\text{Size}_i) \sum_{j=1}^{p} NRC_j$$

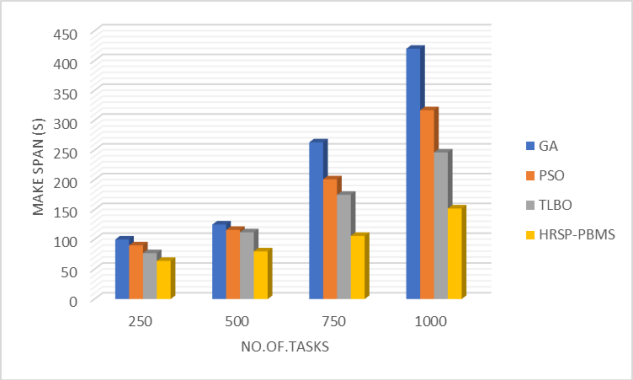
Where NRC_j is the number of requested contents and $Size_i$ is the requested content size in cluster i correspondingly. Cache size is determined by contents maximum size.

V. RESULTS AND DISCUSSION

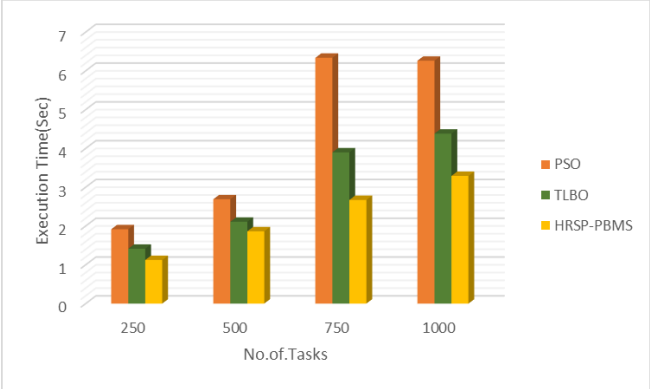
To validate the performance of the proposed methodology hesitancy handling resource scheduling policy using Population based Metaheuristic Search (HRSP-PBMS) in cloud environment is developed and simulated using Matlab 2018b software. The system with i5 processor, 8GB RAM and 1TB hard disk capacity. The cloud data used for resource scheduling process is depicted in the table 1. The

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

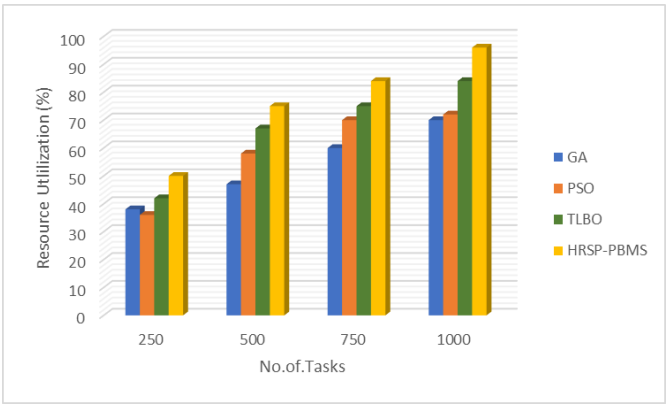
number of jobs varies from 10 to 1000 are considered to be in priority queue which has to be executed in the virtual machine's selection from 100 to 200. Weights of three objectives response time, execution time and throughput are set to 1. The results of proposed HRSP-PBMS is compared with genetic algorithm (GA), Participle Swarm Optimization (PSO), traditional teaching learningbased optimization (TLBO) and their performance outcome are explained in the following section.



The figure depicts the performance of four different resource scheduling schemes in cloud environment based on the makespan of four different size of number of tasks with 100 virtual machines. The makespan is referred as total execution time needed to execute entire jobs. From the result it is observed that theHRSP-PBMS is better compared to GA, PSO and traditional TLBO in terms of makespan because the problem of uncertainty in scheduling resources to the jobs in cloud environment are greatly handled by using intuitionistic fuzzy logic to search for more appropriate virtual machine to complete the selected job. This merit of HRSP-PBMS motivates to decrease the makespan of entire resource scheduling process in cloud environment.

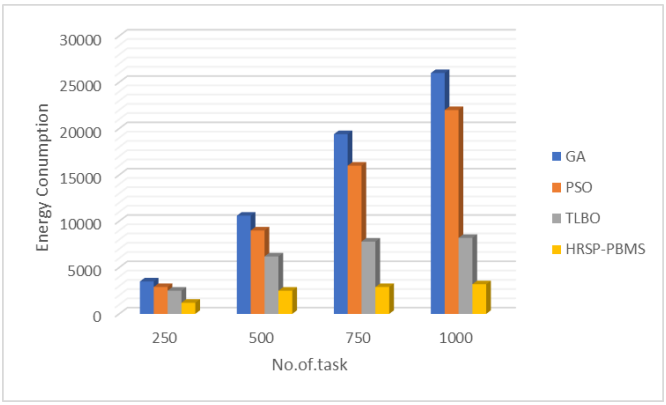


Based on the results illustrated in the figure, the execution time of GA, PSO TLBO and proposed HRSP-PBMS when using 10 virtual machines with varying size of tasks in cloud environment. The outcome of the performance exposed that HRSP-PBMS outperforms the reaming scheduling policies because using the grade of membership and non-membership of parameters needed by each job to be assigned for the appropriate virtual machine the impreciseness in selection process is greatly achieved and with the help of best fitness value the resources are allotted.



From the figure it is discovered that the resource utilization is highly influenced by intuitionistic fuzzy based hesitancy handling with population based meta heuristic search to acquire maximum benefits from the cloud service provider by gaining restricted resources to the cloud consumers with reduced make span. This proposed HRSP-PBMSpolicy selects the optimal virtual machines which complete the task of assigned job with uniform distribution of load of work and thus overcomes the overfitting problem and overload which was failed by another scheduling policies GA, PSO and traditional TLBO.

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881



While accomplishing reduced response and execution time and increase throughout which is based on the CPU and resources utilization in cloud are the main influences of energy consumption by a job. Due to imprecise knowledge heavy load and lack in selection of appropriate virtual machines to execute the job in a successful manner is very tough and results in low performance by GA. PSO and TLBO as shown in the figure. Whereas, HRSP-PBMSutilize less energy for the reason that it uses low makespan and choice of optimal virtual machines allocation greatly enriched the performance of it.

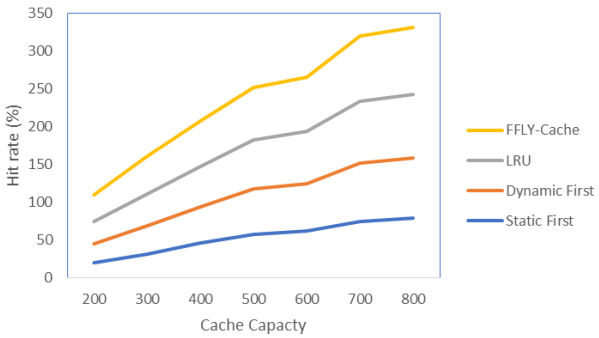


Figure: Performance Comparison of Four different caching policies in cloud paradigm

From the figure the outcome of the results revealed that the performance of firefly intelligent influenced caching policy performs better and controls the cost computation in an optimized manner in the adversarial environment very effectively compared to the traditional models Static first, Dynamic first and LRU. The hit rate of the developed FFLU-Cache achieves higher even in the lower cache capacity while comparing other three policies, and as the cache capacity increases the performance of all the four models increases their hit rate gradually.

VI. CONCLUSION

This work mainly focused on handling the uncertainty on resource scheduling in cloud environment by treating the impreciseness in job assignment to appropriate virtual machines are done by devising intuitionistic fuzzy population-based metaheuristic searching policy. The population metaheuristic is developed by the inspiration of teaching-learning process with optimization of multi objective aiming by maximizing the throughput of the resource scheduling by reducing the response and completion time of jobs with required QoSstandard. Once the job is submitted to the load scheduler the optimization of caching policy is also taken as a primary factor for optimization of resource utilization which fine grains the problem to the host level. The metaheuristic searching model firefly algorithm is used for caching policy to handle both dynamic and static content in a precise manner. The simulation result proved the performance of the proposed model greatly treats the vagueness in assigned of jobs to the appropriate virtual machines using an optimized policy in cloud environment.

VII. REFERENCES

1. G Tilak, S., &Patil, D. (2012). A survey of various scheduling algorithms in cloud environment. International Journal of Engineering Inventions, 1(2), 36-39.
2. Buyya, R., Vecchiola, C., &Selvi, S. T. (2013). Mastering cloud computing: foundations and applications programming. Newnes.
3. Alkayal, E. S., Jennings, N. R., &Abulkhair, M. F. (2016, November). Efficient task scheduling multi-objective particle swarm optimization in cloud computing. In 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops) (pp. 17-24). IEEE
4. K. Pradeep and T. P. Jacob, “Comparative analysis

January - February 2020
ISSN: 0193 - 4120 Page No. 2871 - 2881

of scheduling and load balancing algorithms in cloud environment", In: Proc. of International Conf. on Control, Instrumentation, Communication and Computational Technologies, pp. 526-531, 2016.

5. R. Raju, J. Amudhavel, M. Pavithra, S. Anuja, and B. Abinaya, "A heuristic fault tolerant Map Reduce framework for minimizing makespan in Hybrid Cloud Environment", In: Proc. of International Conf. on Green Computing Communication and Electrical Engineering, pp. 1-4 , 2014.

6. Ge, Y., & Wei, G. (2010, October). GA-based task scheduler for the cloud computing systems. In 2010 International Conference on Web Information Systems and Mining (Vol. 2, pp. 181-186). IEEE.

7. Jang, S. H., Kim, T. Y., Kim, J. K., & Lee, J. S. (2012). The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation, 5(4), 157-162.

8. Guo Q, Task scheduling based on ant colony optimization in cloud environment. In AIP Conference Proceedings (Vol. 1834, No. 1, p. 040039), 2017.

9. Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing, 2687-2699, 2015.

10. H.Liu, A.Abraham, A.E.Hassanien, Scheduling Jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems (2009).

11. Ch.SrinivasaRao, B.RaveendraBabu, DE Based Job Scheduling in Grid Environments, Journal of Computer Networks, 2013 Vol. 1, No. 2, 28-31.

12. Juan, W., Fei, L., &Aidong, C. (2012). An Improved PSO based Task Scheduling Algorithm for Cloud Storage System. Advances in Information Sciences and Service Sciences, 4(18), 465-471.

13. Krishnasamy, K. (2013). Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization In Cloud Computing Environment. Journal of Theoretical & Applied Information Technology, 55(1)

14. Krassimir T. Atanassov, Fuzzy Sets and Systems, North-Holland, Volume 20, pages 87-96, ISSN 0165-0114, 1986.

15. Intuitionistic Fuzzy Sets, Krassimir T. Atanassov, Series Studies in Fuzziness and Soft Computing, Volume 35, Springer Physica-Verlag, 1999, ISBN 3-7908-1228-5

16. Rao, R. V., Savsani, V. J., Vakharia, D. P, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303-315, (2011).

17. I. S. Altingovde, R. Ozcan, O. Ulusoy, A cost-aware strategyfor query result caching in web search engines. In Proc. of the31th European Conference on IR Research on Advances in InformationRetrieval (ECIR), 2009.

18. K. S. Candan, W.-S. Li, Q. Luo, W.-P. Hsiung, D. Agrawal, Enablingdynamic content caching for database-driven web sites. In Proc. ACMSIGMOD, 2001

19. NurFarahlinaJohari, AzlanMohd Zain, NoorfaMustaffa, AmirmudinUdin, Firefly Algorithm for Optimization Problem, Applied Mechanics and Materials ,421, 2013

20. Xin-She Yang, Firefly Algorithms for Multimodal Optimization, Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications, pp 1-11, 2010.