International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 3| ISSN: 2456-3307

Task Scheduling and Resource Allocation Using a Heuristic Approach In Cloud Computing

K. Durailingam[1], Dr.V.S. Prakash[1]
[1]Indian Arts and Science College, Kondam, Tiruvannamalai, Tamil Nadu, India

ABSTRACT

Cloud computing is needed by trendy technology. Task planning and resource allocation are vital aspects of cloud computing. This paper proposes a heuristic approach that mixes the changed analytic hierarchy method (MAHP), bandwidth aware divisible scheduling (BATS) + BAR optimization, longest expected processing time preemption (LEPT), and divide-and-conquer strategies to perform task planning and resource allocation. During this approach, every task is processed before its actual allocation to cloud resources using a MAHP process. The resources are allocated victimization the combined haywire + BAR optimization methodology, that considers the information measure and cargo of the cloud resources as constraints. Additionally, the planned system preempts resource intensive tasks exploitation LEPT preemption. The divide-and-conquer approach improves the planned system, as is established by experimentation through comparison with the existing bats and improved differential evolution algorithmic rule (IDEA) frameworks once turnaround and time interval square measure used as performance metrics.

Keywords: Cloud computing, Task planning, Heuristic, Resource management, Analytic hierarchy system, BATS, BAR

I. INTRODUCTION

Cloud computing is an accelerating technology within the field of distributed computing. Cloud computing will be utilized in applications that embody storing information, information analytics and IoT applications [1]. Cloud computing may be a technology that has modified ancient ways in which during which services square measure deployed by enterprises or people. It provides differing kinds of services to registered users as net services so the users don't need to invest in computing infrastructure. Cloud computing provides services like IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) [2]. In every type of service, the users square measure expected to submit the requests to the service supplier through the medium of the internet. The service supplier is responsible for managing the resources to fulfill the requests generated by users. Service suppliers use programing algorithms to schedule the incoming request (tasks) and to manage their computing resources with efficiency. Task programing and resource management allow providers to maximize revenue and the utilization of resources up to their limits. In follow, in terms of the performance of cloud computing resources, the programing and allocation of resources are vital hurdles. For this reason, researchers are interested in studies of task programing in cloud computing. Task programing is that the method of arrangement incoming requests (tasks) in a very bound manner so the obtainable resources are properly utilized. as a result of cloud computing is that the technology that delivers services through the medium of the web, service users should submit their requests on-line. because every service features a variety of users, variety of requests (tasks) is also generated at a time. Systems that do not use scheduling might feature longer waiting periods for tasks what is more, some short-run tasks might terminate, as a result of the waiting amount. At the time of programing, the hardware must think about variety of constraints, as well as the character of the task, the dimensions of the task, the task execution time, the avail- ability of resources, the task queue, and the load on the resources. Task programing is one amongst the core problems in cloud computing. correct task programing might end in the economical utilization of resources. the key advantage of cloud computing is that it promotes correct utilization of resources [3]. Thus, task programing and resource allocation square measure 2 sides of a single coin. every affects the opposite. Currently, net users will access content anyplace and anytime, with no need to contemplate the hosting infrastructure. Such hosting infrastructure consists of various machines with numerous capabilities that square measure maintained and managed by the service supplier. Cloud computing enhances the capabilities of such infrastructure, which may access the web. Cloud service suppliers earn profits by providing services to cloud service users.

The cloud service user will use the complete stack of computing services, that ranges from hardware to applications. Services in cloud computing use a pay-as-you-go basis. The cloud service user will scale back or increase the obtainable resources, per the stress of the applications. this is often one the key benefits of cloud computing, however service users are also accountable for paying extra prices for this advantage. The cloud service user will rent the re- sources at any time and unharness them with no issue. The cloud service user has the liberty to use any service supported application would like. The liberty of service alternative for users has semiconductor diode to problems; that's future user request cannot be dead expected. Thus, task programing and re- supply allocation square measure mandatory components of cloud computing analysis. The potency of resource uses depends on the programing and cargo leveling methodologies, rather than the random allocation of resources. Cloud computing is wide used for finding complicated tasks (user requests). In finding complicated task problems, the employment of programing formula is suggested. Such programing algorithms leverage the resources. The projected system employs options of the Cybershake scientific workflow and the Epigenomics scientific progress, that are represented in Section computer file. The major contributions of this paper are summarized as follows.

1. The analytic hierarchy method is changed to rank scientific tasks.
2. To manage the resources given information measure constraints and therefore the load on the virtual machine, the projected system incorporates a version of the prevailing round the bend formula that has been changed by introducing BAR system optimization.
3. Bipartite graphs square measure utilized to map tasks to appropriate virtual machines once the condition is glad.
4. A pre-emption methodology offers US the standing of the virtual machine, and a changed divide-and-conquer methodology has been projected to mixture the results when tasks pre-emption.
5. The projected resolution is by experimentation-investigated victimization the Cloud Sim machine.

The remainder of the paper is organized as follows. Section "Introduction" provides Associate in Nursing introduction to cloud computing and its outstanding problems, particularly task programing and resource allocation. Section "Related work" focuses on connected studies that investigate task programing and resource allocation. Section "Input data" describes the computer file provided to the Cybershake scientific progress and the Epigenomics scientific workflow. Section "Proposed system" addresses the design of the projected system. Section "Proposed Methodology," explains the projected methodology. Section "Evaluation of the projected heuristic approach" focuses on evaluating the projected heuristic approach. Section "Results and discussion" describes the results and discusses the projected system compared with the prevailing round the bend and plan algorithms. Finally, concluding remarks and future directions are conferred in Section "Conclusion".

II. RELATED WORK

This section provides a short review of task programing and resource allocation ways. Several researchers have projected solutions to beat the matter of programing and resource allocation. However, additional improvements will still be created. Tsai et al. [4] projected a multiobject approach that employs the improved differential evolution algorithmic rule. This existing technique provides a value and time model for cloud computing. However, variations within the tasks aren't thought-about during this approach. Magukuri et al. [5] projected a load balancing and programing algorithmic rule that doesn't contemplate job sizes. The authors considered the refresh times of the server in fulfilling requests. Cheng et al. [6] introduced the programing of tasks supported a vacation queuing model. This technique doesn't show the correct utilization of resources. Lin et al. [7] projected the programing of tasks whereas considering information measure as a resource. A nonlinear programming model has been shaped to portion resources to tasks. Ergu et al. [8] proposed AHP ranking-based task programing. Zhu et al. [9] introduced rolling-horizon programing architecture to schedule real-time tasks. Authors have illustrated the link between task programing and energy conservation by resource allocation. Lin et al. [10] projected programing for parallel workloads. Authors have used the FCFS approach to order jobs once resources are avail in a position. The projected system doesn't specialize in aborting the roles and starvation. Ghanbari et al. [11] projected a priority-based job programing algorithmic rule to be used in cloud computing. Multi criteria selections and multiple attributes square measure thought-about. Polverini et al. [12] introduced the optimized value of energy and queuing delay constraints. Alejandra et al. [13] projected the employment of meta-heuristic optimization and particle swarm optimization to reduce execution prices through programing. Keshk et al. [14] projected the employment of changed ant colony optimization in load equalization. This technique improves the makespan of a job. This technique doesn't contemplate the supply of resources or the load of tasks. Shamsollah et al. [15] projected a system supported a multi-criteria algorithmic rule for programing server load. Shamsollah et al. [16] pro- posed a system supported priority for acting separable load programing that employs analytical hierarchy method. Gougarzi et al. [17] projected a resource allocation drawback that aims to reduce the full energy value of cloud computing systems while meeting the required client-level slas in a very probabilistic sense. Here, authors have applied a reverse approach that applies a penalty if the client doesn't meet the SLA agreements. Some authors have enforced a heuristic algorithmic rule to solve task programing and resource allocation drawback de- scribed higher than. Radojevic et al. [18] introduced central load equalization call model to be used in cloud environments; this model automates the programing method and reduces the role of human administrators. However, this model is deficient in decisive the capabilities of nodes and, configuration details, and therefore the complete sys- tem has no backup, therefore leading to one purpose of failure. Additionally, Ghanbari et al. [12] and Goswami et al. [14] specialise in programing tasks whereas considering various constraints. This state-of the art motivates the authors of this study to conduct additional analysis on task programing and resource allocation.

III. INPUT WORK

Cybershake scientific workflow Cloud computing is that the service supplier paradigm within which users submit requests for execution. Thus, the responsibility of the cloud service supplier is to schedule various requests and manage resources with efficiency. To the most effective of the authors' data, most existing work involves scheduling tasks once they enter a task queue. However, the particular procedure of planning tasks and resource management begins with however the service supplier addresses incoming tasks. The pro- posed system uses Cybershake scientific advancement knowledge as in place tasks [12]. Fig. 1 shows a visualization of the Cybershake scientific advancement, that is employed by the Southern CA Earthquake Center (SCEC) to characterize earthquake haz- wet lung using the Probabilistic Unstable Hazard Analysis (PSHA) technique. It additionally generates inexperienced strain tensors (GSTs). Table one shows the Cybershake seismogram synthesis tasks with their sizes and execution times. The Cybershake could be a collection of assorted node knowledge that area unit out there for study [14]. The Cybershake scientific work- flow sample tasks area unit out there with task size 30,50,100 and 1000. From a machine purpose of read, the seismogram synthesis tasks area unit quite stringent. The Cyber- shake spends lots of time on seismogram synthesis throughout its execution. These kinds of tasks additionally need large amount of computational resources, like central processing unit time, and memory.

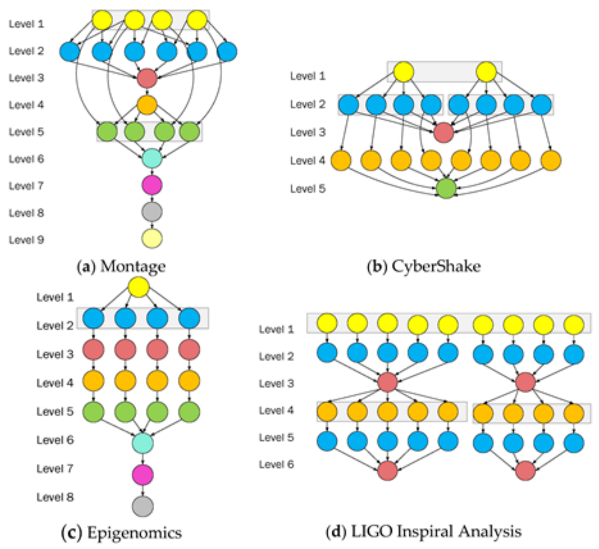


Figure 1. Cybershake scientific workflow

Cybershake scientific workflow has been divided into 5 steps.

1. Extract GST - This step of the workflow extracts the GST (Green strain tensor) data for processing.
2. Seismogram synthesis – These tasks are the most computationally intensive. Most of the time spent in running the Cybershake algorithm is employed on this step.
3. ZipSeis–This step aggregates the processed data.
4. PeakValCalcOkaya – The highest-strength values of each seismogram are calculated in this step.
5. ZipPSA-This step aggregates the processed data.

Table 1 Cybershake seismogram synthesis tasks

Tasks	Size	of
tasks	Time	
Task 3	62,69,51,663	39.06
Task 5	69,47,76,323	38.49
Task 7	58,57,63,637	36.27
Task 9	53,68,97,326	32.29
Task 11	67,05,35,542	62.25
Task 14	40,67,28,38,798	96.91
Task 16	45,23,96,996	45.60
Task 18	50,27,64,231	28.67
Task 20	62,41,88,532	24.56
Task 22	42,65,77,006	31.05
Task 24	51,58,32,878	54.87
Task 26	68,14,99,417	23.99
Task 28	44,14,51,516	26.46

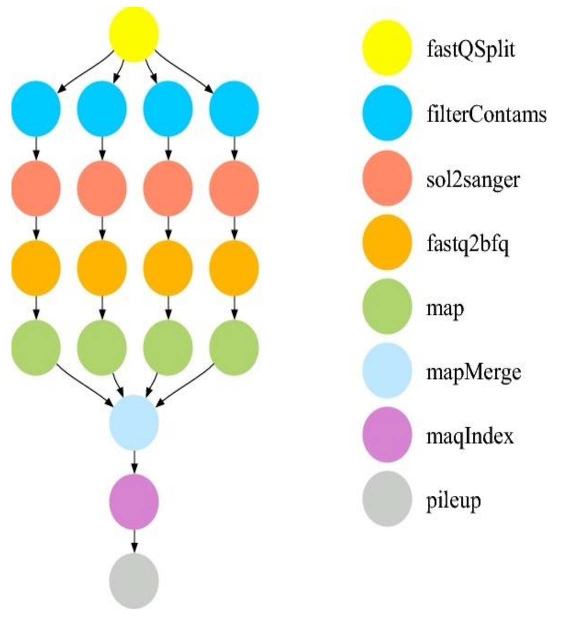


Figure 2. Epigenomics Scientific Workflow

IV. EPIGENOMICS SCIENTIFIC WORK FLOW

Figure a pair of shows the Epigenomics scientific workflow [18] that is used to change the method of ordering sequencing. This operation is related to resource- intensive tasks. The generated information area unit born-again into files and forwarded to magazine system. This method additionally involves several operations, andthese operations area unit time over whelming.

V. PROPOSED SYSTEM

Figure three shows the design of the planned system. In apply, varied varieties and sizes of tasks gain the cloud information centers for execution. The planned system takes the $64000 tasks as Associate in Nursing input, as delineate in Section 3. In general, scientific tasks represent collections of various types and sizes. To manage the tasks that get a cloud information center, the planned system uses the analytic hierarchy method (AHP). The first aim of this planned system is to manage incoming tasks. Therefore, the planned system uses the AHP method-field of study to assign a rank to every task supported its length and run time. The procedure for ranking the tasks for scientific workflows is delineate in section 5.1. As before long because the tasks area unit assigned individual rankings, they are collected and organized into task queues. The tasks within the task queue area unit strictly organized following the AHP ranking. Thus, the primary stage of the planned system is completed. Next, within the second stage, the planned system additionally addresses the computing resources of cloud information centers, like central processing unit, memory and information measure mistreatment the planned BATS+BAR optimized allocation methodology. This technique works as follows. It takes the task to be dead from the task' queue. The assignment of resources and tasks follows the allocation combining weight. 4. A detailed clarification is givenin section five.2. This stage is that the second a part of the procedure during which the allocations of resources are distributed using BATS+BAR. Within the next half, the planned system uses a preemption methodology, i.e., the preemption technique. LEPT incessantly checks the load of the Virtual machine. If it's exceeded theproposed system then uses a virtual machine standing table to confirm this standing of alternative Virtual machines (vms). During this regard, if this virtual machine is overladen et al. Area unit idle, then such vmsarea unit set. Once this identification, the proposed system uses a divide-and-conquer methodology that breaks up the task and distributes it to alternative virtual machines, as described intimately in section five.3. During this means, the propose system has overcome the restrictions of buggy in terms of the allocation of resources based on CPU, memory and information measure. If anyone resource (CPU, memory, bandwidth) is not available in sufficient amounts, then the tasks must wait. In addition, existing systems do not consider preemption, and the inputs to existing systems are tasks of the same size. Fig.4 presents a flow chart that represents the proposed heuristic approach.

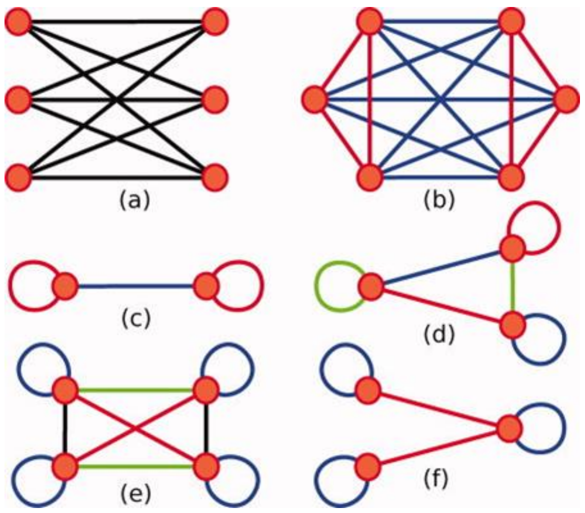


Figure 4. Complete Bipartite Graph

VI. PROPOSED METHODOLOGY

Here, we offer a detailed explanation of the proposed system to beat the programing challenge.

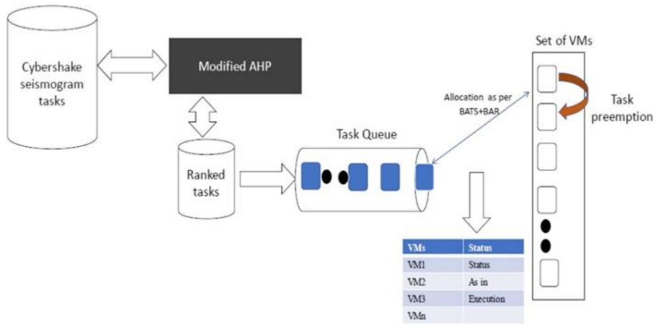


Figure 3. Proposed System Architecture

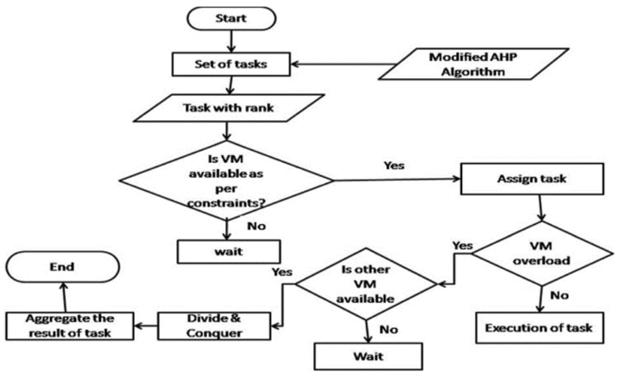
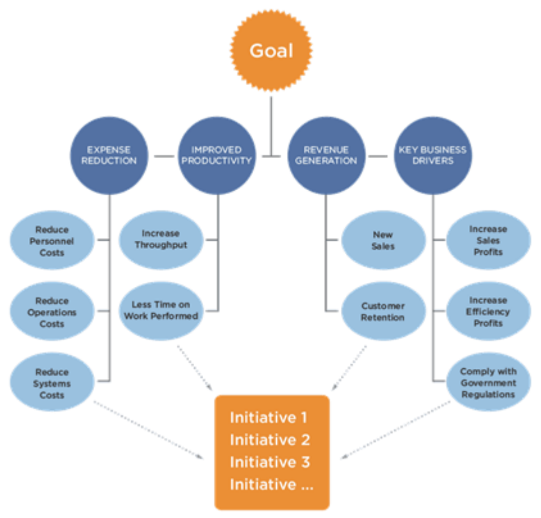


Figure 4. Proposed System Flowchart

VII. ANALYTIC HIERARCHY PROCESS

The analytic hierarchy process [18] is intended to solve complicated issues with multiple criteria. The proposed system uses this procedure in cloud computing environments to rank the incoming tasks in a sure manner. The projected system uses scientific work flow tasks, like those of Cybershake and Epigenomics, for experiments because such need long execution times. Initially, the advancement is Split into 5 stages, which are introduced in the input information section. Before continuing with the planned system, the AHP methodology is applied for the Overall Cybershake advancement.The Cybershake workflow is management flow Dependent; so, the second stage can execute only once the execution of the primary stage.

To evaluate preferences, the projected system uses the Saaty preference table, that is given in Table two with its numerical ratings. To push understanding whereas accounting for house limitations, the projected system divides every calculation table into 2 elements. The primary half extends from Task three to Task , whereas the opposite half shows the calculations from Task twenty to Task . Here, the projected system considers two important criteria that are concerned in scientific tasks; task length and task run time. The comparison numerical ratings are given in Table 2, that is thought because the Saaty preference table. Before the actual calculation is begun, the projected system assigns preference values to the tasks. Here, the preferences associated with the tasks are supported their lengths and also the execution times of the various tasks. The projected system slightly modifies the Saaty table preferences as a result of, as tasks with totally different ranks are on a server, the ranks of resulting tasks amendment, and new rankings should be calculated.



THE ANALYTIC HIERARCHY PROCESS (AHP) MODEL

The pro- display system calculates such rankings of tasks. Tables 3 and 4 show the assignment of Saaty preferences in line with examination the sizes and runtimes of tasks. Within thebottom row, the add of every column is noted. Tables 5 and 6 show the multiplication of the Saaty preference values by the results organized within the bottom rows of Tables 3 and 4 and so gift the results of adding every column at rock bottom. Tables 2 and 4 show the normalized values of Tables 5 and 6, which appear earlier in the manuscript. These tables include average at the bottom. The results show that the summation of each column is equal to 1.

Table 2 Numerical saaty preferences

Numerical rating	Judgment preference
9	Extremely preferred
8	Very strongly to extremely preferred
7	Very strongly preferred to preferred
6	Strongly to very strongly
5	Strongly preferred
4	Moderately to strongly preferred
3	Moderately preferred
2	Equally to moderately preferred
1	Equally preferred

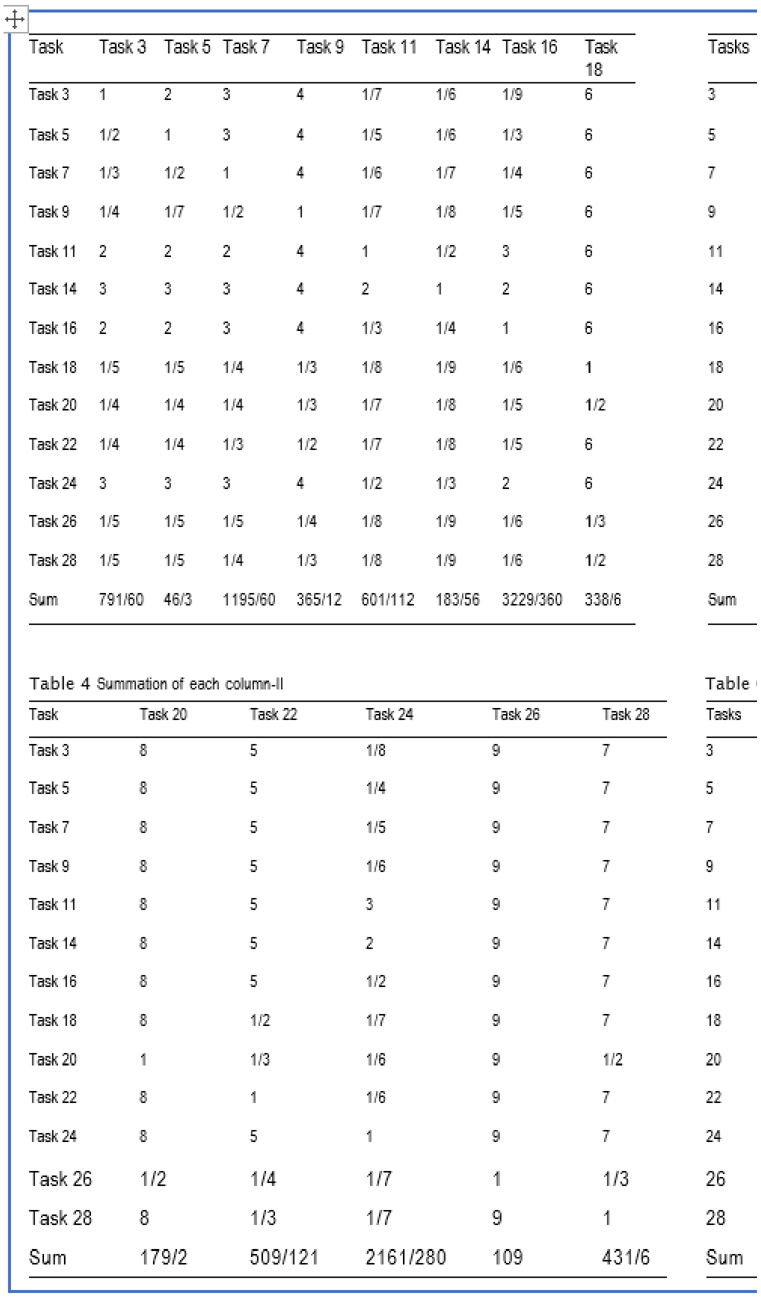
VIII. BATS+ BAR SYSTEM

The planned system has two aspects that involve planning tasks and managing resources. Here, we improve upon the round the bend algorithmic rule, that was originally planned by Weiwei Lin [7]. freelance tasks of equal size ar considered within the design of this technique. How- ever, in allocating resources, the system doesn't consider the load on virtual machines as a result of the waiting amount for the tasks is long. In alternative cases, one virtual machine is busy whereas it executes a task, whereas others ar occupied and waiting for jobs. The bar systems (BSs) algorithmic rule was planned by Acebo and Rosa (2008) [14]. The social behavior of bartenders is that the basis of BS systems. Swarm intelligence has else an optimisation side to BS. In a bar, bartenders should act in a very extremely dynamic, asynchronous and time-critical setting, and no obvious greedy strategy (such as serving the most effective client initial, serving the closest client initial or serving the first-arriving client first) offers smart results. Thus, multi-agent systems offer a decent framework at intervals that to handle the challenge of developing a brand new category of adaptational and sturdy systems. In general, the crucial step within the SB algorithmic rule is that the selection of the task that the agent should execute within the next time step. In BSs, agents acting as bartenders, operate at the same time in an environment during a exceedingly in a very synchronous manner; that's, they execute tasks by deciding that drinks to pour. once Associate in Nursing initial section, the "bartenders" build their selections according to totally different problem-dependent properties (e.g. weight, speed, location, time interval, most load, etc.), instead of constructing selections willy-nilly. Over time, if an agent is unable to adapt the environment to the preconditions of the task (such because the value for the agent to execute the task within the current state of the environment) or if it's unable to hold the task out by itself, it'll be eliminated. To over- return this behavior, we propose modifying nuts by adding a BAR system.

The procedure is as follows:

- Aggregate all of the task information that is ordered by rank.

- Virtual machine (server) information is collected. This information includes the initial load on the virtual machine, its bandwidth and the time required to process the tasks on the server.
- A bipartite graph is generated with the number of tasks. The ranking prioritiescan be used to construct a graph, by which each task is allocated to a virtual machine.



Task	Task 3	Task 5	Task 7	Task 9	Task 11	Task 14	Task 16	Task 18	Tasks
Task 3	1	2	3	4	1/7	1/6	1/9	6	3
Task 5	1/2	1	3	4	1/5	1/6	1/3	6	5
Task 7	1/3	1/2	1	4	1/6	1/7	1/4	6	7
Task 9	1/4	1/7	1/2	1	1/7	1/8	1/5	6	9
Task 11	2	2	2	4	1	1/2	3	6	11
Task 14	3	3	3	4	2	1	2	6	14
Task 16	2	2	3	4	1/3	1/4	1	6	16
Task 18	1/5	1/5	1/4	1/3	1/8	1/9	1/6	1	18
Task 20	1/4	1/4	1/4	1/3	1/7	1/8	1/5	1/2	20
Task 22	1/4	1/4	1/3	1/2	1/7	1/8	1/5	6	22
Task 24	3	3	3	4	1/2	1/3	2	6	24
Task 26	1/5	1/5	1/5	1/4	1/8	1/9	1/6	1/3	26
Task 28	1/5	1/5	1/4	1/3	1/8	1/9	1/6	1/2	28
Sum	791/60	46/3	1195/60	365/12	601/112	183/56	3229/360	338/6	Sum

Table 4 Summation of each column-II

Task	Task 20	Task 22	Task 24	Task 26	Task 28	Tasks
Task 3	8	5	1/8	9	7	3
Task 5	8	5	1/4	9	7	5
Task 7	8	5	1/5	9	7	7
Task 9	8	5	1/6	9	7	9
Task 11	8	5	3	9	7	11
Task 14	8	5	2	9	7	14
Task 16	8	5	1/2	9	7	16
Task 18	8	1/2	1/7	9	7	18
Task 20	1	1/3	1/6	9	1/2	20
Task 22	8	1	1/6	9	7	22
Task 24	8	5	1	9	7	24
Task 26	1/2	1/4	1/7	1	1/3	26
Task 28	8	1/3	1/7	9	1	28
Sum	179/2	509/121	2161/280	109	431/6	Sum

The Load on the virtual machine(S) is calculated as,

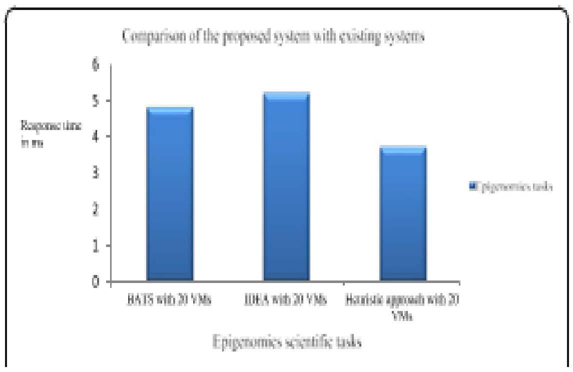
L^{ini} ¼ L_s^{ini} js⊂S 1

The bandwidth is calculated as,

DB_w¼b'i<¼bi 2

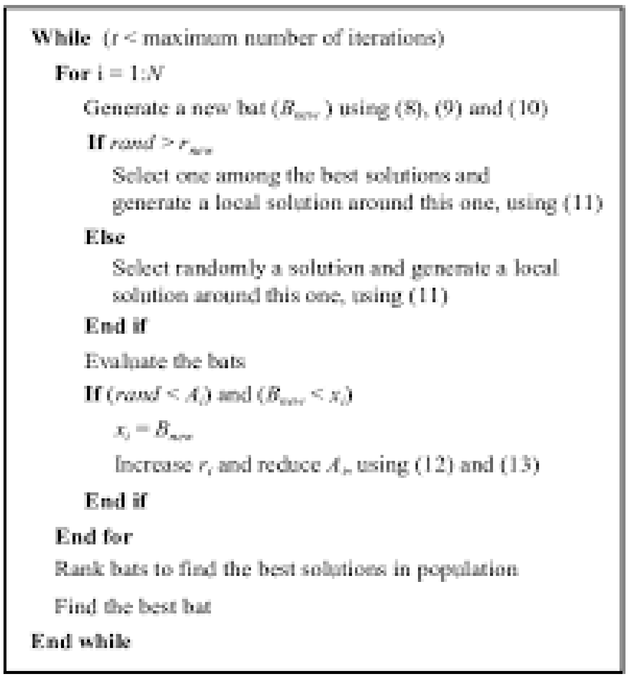
Thetotaltimetakentoprocessthetasksiscalculatedas,

L_s^{fin}ðαÞ¼L_sðαÞ 3



CPU utilizing Bar Diagram

Where, (α) = any task.



```
While  ($r$ < maximum number of iterations)
  For i = 1:$N$
    Generate a new bat ($B_{new}$ ) using (8), (9) and (10)
    If $rand > r_{new}$
      Select one among the best solutions and
      generate a local solution around this one, using (11)
    Else
      Select randomly a solution and generate a local
      solution around this one, using (11)
    End if
    Evaluate the bats
    If ($rand < A_i$) and ($B_{new} < x_i$)
      $x_i = B_{new}$
      Increase $r_i$ and reduce $A_i$, using (12) and (13)
    End if
  End for
  Rank bats to find the best solutions in population
  Find the best bat
End while
```

Bipartite graph

A bipartite graph is produced based on the following conditions:

1. A bipartite graph is constructed as-, G = (T_nU S, E) in which 'T_n' represents the number of tasks, 'S'represents the servers, and 'E ⊆TX S' that is, the set of edges that are present between the task and the server. An edge represents the tasks 'Ti ⊂Tn', which are present on virtual machine'rs⊆S'. 2.Agraph isconstructed using bipartite graph with the number of tasks.
2. Balancetheconstructedgraphwithconstrain ts includingthelocalcost,theinitialloadandthe bandwidth.
3. Based on the local cost and the initial load we computethetotalloadonthevirtualmachine.
4. Next,weapplytheconditionrepresentedbyE q.If thisconditionissatisfied,thenweallocatethet asks tothatparticularvirtualmachine.Ifthiscondit ionis not satisfied by that virtual machine, then we move on the next server and check thiscondition.
5. After allocating the tasks, the constructed bipartite is updated if any task remain to be processed. It is the bipartite graph of the set of virtual machines and set ofresources.

IX. EVALUATION OF TIME INTERVAL

As a second performance metric, we take into account the time interval of the algorithm to incoming tasks. The time interval is actually the time during that the request is actually considered. In different words, we are able to say that the time interval is directly addicted to the supply of resources. The availability of resources depends abreast of the programming of tasks. If the programming of tasks is performedproperly, then the resources can naturally be free early or earlier of deadlines, the response times can be less in such cases.

By, scrutiny the response times obtained for our proposed heuristic approach with those obtained using the existing bats and idea frameworks, we are able to see that our system's response time is almost 500th less. The latent period comparisons for Cybershake and Epigenomics are conferred in Figs. and nine respectively. The comparison is additionally shown in tabular we have a tendency to take into account two parameters the latent period and turnaround compare the proposed heuristic approach with the prevailing nutty and plan frameworks. Because we have a tendency to be evaluating these frame- works in a very cloud computing surroundings, the response time is generally less effective.

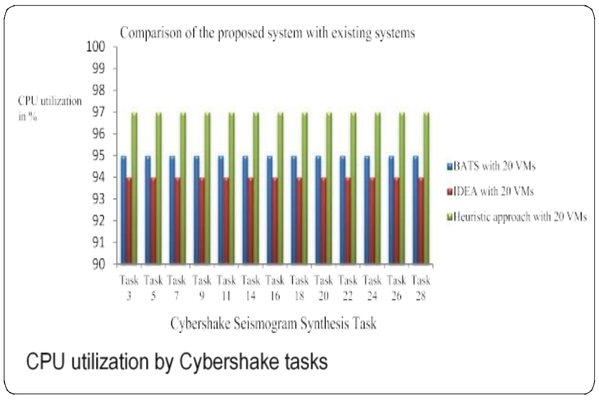
On the other hand, we have a tendency to additionally evaluate our planned heuristic approach to determine its resource performance compare it to those of the prevailing nutty and plan frameworks.

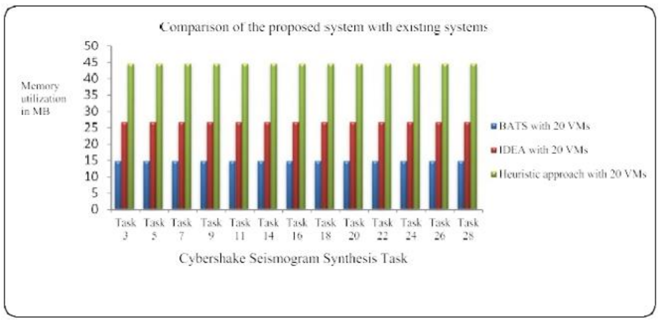
X. EVALUATION OF CPU UTILIZATION

Key comparison of resource utilization between the planned heuristic approach and existing nutty and plan frameworks. The proper utilization of resources produces profits for cloud computing service suppliers. The experimental results shows that the planned heuristic approach utilised the CPU resource more efficiently than the prevailing nutty framework.

XI. EVALUATION OF MEMORY UTILIZATION

The second key comparison of resource utilization between the planned heuristic approach and therefore the existing nutty and idea frameworks. The experimental results shows that the proposed heuristic approach utilizes memory re- sources more efficiently than the prevailing idea and bats frameworks.





XII. CONCLUSION

In this study, we proposed heuristic formula that performs task scheduling and allocates resources efficiently in cloud computing environments. We use real Cybershake and Epigenomics scientific workflows as input tasks for the system. After we compare our projected heuristic approach with the existing fruity and plan frameworks with respect to turnaround time and interval, we find that our approach provides improved results. On the opposite hand, from the point of view of resupply utilization, the projected heuristic approach efficiently allocates resources with high utility. We tend to obtain the utmost utilization result for computing resources like computer hardware, memory and bandwidth. Most existing systems consider only 2 resources, computer hardware and memory, in evaluating their performance the projected system adds bandwidth as a resource. Future work can specialize in simpler scheduling algorithms in which turn- around time and response time will be improved.

XIII. REFERENCES

[1] Mezmaz M, Melab N, Kessaci Y, Lee YC, Talbi E-G, Zomaya AY, Tuyttens D (2011) A parallel bi-objective hybrid meta heuristic for energy-aware scheduling for cloud computing systems. J Parallel Distributed Computing 71(11):1497–1508

[2] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I et al (2010) A view of cloud computing. Commun ACM 53(4):50–58

[3] Tsai J-T, Fang J-C, Chou J-H (2013) Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. ComputOper Res 40(12):3045–3055

[4] Cheng C, Li J, Wang Y (2015) An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. Tsinghua SciTechnol 20(1):28–39

[5] Ergu D, Kou G, Peng Y, Shi Y, Shi Y (2013) The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. The Journal of Supercomputing. 64(3):835-848

[6] Zhu X, Yang LT, Chen H, Wang J, Yin S, Liu X (2014) Real-time tasks oriented energy-aware scheduling in virtualized clouds. IEEE Transactions on Cloud Computing 2(2):168–180

[7] Handfield R, Walton SV, Sroufe R, Melnyk SA (2002) Applying environmental criteria to supplier assessment: a study in the application of the analytical hierarchy process. Eur J Oper Res 141(1):70–87

[8] Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience

[9] Maguluri ST, Srikant R (2014) Scheduling jobs with unknown duration in clouds. IEEE/ACM Trans Netw (TON) 22(6):1938–

[10] Lin W, Liang C, Wang JZ, Buyya R (2014) Bandwidth-aware divisible task scheduling for cloud computing. Software: Practice and Experience 44(2):163–174

[11] Shamsollah G, Othman M (2012) Priority based job scheduling algorithm in cloud computing. Procedia Engineering 50:778–785

[12] Polverini M, Cianfrani A, Ren S, Vasilakos AV (2014) Thermal aware scheduling of batch jobs in geographically distributed data centers. IEEE Transactions on Cloud Computing 2(1):71–84

[13] Keshk AE, El-Sisi AB, Tawfeek MA (2014) Cloud task scheduling for load balancing based on intelligent strategy. Int J IntellSystAppl 6(5):25

[14] Rodriguez MA, Buyya R (2014) Deadline based resource provisioningand scheduling algorithm for scientific workows on clouds. IEEE Transactions on Cloud Computing 2(2):222–235

[15] Liu X, Zha Y, Yin Q, Peng Y, Qin L (2015) Scheduling parallel jobs with tentative runs and consolidation in the cloud. J SystSoftw 104:141–151