A Novel heuristic technique based load balancing and resource allocation framework in cloud computing environment

Siva Rama Krishna[*], Dr. Mohammed Ali Hussain[#]
[*]Research Scholar, Dept. of CSE, Shri Venkateshwara University, Uttar Pradesh, India.
[#]Professor, Dept. of CSE, Shri Venkateshwara University, Uttar Pradesh, India.

Abstract— As the size of the cloud computing resources and services increases, it is difficult to handle load balancing due to computational cost and time. Since, most of the cloud service providers have their own type, type and price policies for computing resources, including other service features. The load balance between cloud resources ensures an efficient utilization of the physical infrastructure while minimizing runtime. Load balance can improve quality of service (QoS) measurements, including response time, cost, performance and use of resources. In this work, a novel load balancing algorithm is implemented to improve the cloud service load balancing. In order to optimize the delivery of cloud services, the load balance is important between virtual machines at minimum paid costs and overall service delivery time. In order to improve the scheduling process of load-balancing in the cloud environment, many traditional models are used to optimize the load balance. However, the main problem to the cloud service provider's is optimizing cloud service parameters such as reliability, flexibility, time limits and the task refusal rate. A dynamic algorithm is required for the cloud service provider to plan work which will reduce time while increasing the cloud resources use ratio and comply with the user's specific QoS parameters. The proposed framework is based on hybridizing heuristic techniques with metaheuristic algorithm in order to achieve its optimum performance in the load balancing process. Experimental results proved that the present load-balancing model has better performance than the traditional load balancing approaches on various cloud resources.

I. INTRODUCTION

Now-a-days, scheduling of computing resources has become the major concern of different research scientists. It has the basic objective of reducing the task completion time significantly. In case of supercomputers, multi-processor scheduling involves different numbers of parallel processors having equivalent capacity. Apart from this, the data source is required to be centralized and interlinked with the help of a high-speed channel among various processors. In the above scenario, the activities can transfer messages easily and more quickly. There have been extensive amount of research works carried out in the field of scheduling in distributed computer systems. Along with the latest advancements of computer networks, the connecting links in between various computing entities have become faster. Here we can mention that, the latest applications require extended bandwidth, large storage and exchange of huge volumes of data. There are two important applications such as, multimedia and e-Science hose require huge volume of data. It is very much necessary to achieve better performance and quality of service.

Now-a-days, the popularity of customized, high quality products along with quick delivery is forcing different organizations to upgrade their traditional production process. Therefore, implementation of cloud services is very much beneficial. The process of digital transformation or digitization can be defined as a specific kind of process in which the interaction among physical and informal entities exists. Load balancing nodes are represented by Directed-Acyclic Graph. In a workflow diagram, different tasks are interlinked through directed edges in order to represent the data dependency among various tasks. These constraints are known as precedence constraints. Every individual task is executed by considering the starting data inserted through workflow or data inserted through the parent tasks. Basically, tasks of a particular workflow are mostly scheduled in nature. These tasks are usually executed in the distributed form across multiple processing elements without violating precedence constraints. Since last decade, the media streaming services have become more popular on internet. There exist very high demands of dynamic videos from all over the world. As compared to the conventional large server clusters, geo-distributed clouds are more scalable and feasible in nature. Hence, geo-distributed clouds are considered as the perfect solution. The service providers usually set up various data centers at different places. The media streaming system has the responsibility to provide better services near to the actual customers. In case of static contents, content delivery networks are the better and feasible option as compared to cloud. Most of the latest CDN follow the pay-as-you-go pricing schemes. Hence, the service provider is only required to pay for whatever they have downloaded. But in case of dynamic contents, cloud based systems is the best option. If the contents are both static and dynamic, in that case, there is a necessity of hybrid model.

Presently, large numbers of complex data streams are generated by different applications just like multimedia, social media, Internet of Things and social dispersed computing. In order to resolve the issues and support large scale data processing, different methodologies are adopted that supports the concepts of parallelism and big data. Mostly, the cloud customers want to decrease their expenses and delays through combination of privately owned systems with external public infrastructure. Again we can also

state that, the cloud customers are more willing to enhance the process of resource utilization and throughput along with their monetary profits.

Cloud computing is the special domain which is more popular as compared to the distributed and grid computing. It involves the concepts of resource virtualization which is the most attractive feature of cloud. The term "computing" means the execution of different tasks on various virtual machines in order to carry out the process of efficient execution. With the growth of information and technology, the computational power of networks are also enhancing every day.Apart from this, we can manage large numbers of heterogeneous tasks in order to generate better access. Basically, the pay-as-you-go or pay-as-you-use schemes are followed in cloud in order to access resources. Apart from these, it has the responsibility to dynamically allocate cloudlets or tasks to the virtual machines. By implementing the load scheduling algorithm on cloud, dynamic allocation can be obtained easily and efficiently. It has the responsibility to achieve optimized throughput, reduced amount of execution and waiting time, reduced transfer time and decreased computational costs. The process of virtual machine scheduling and utilization is an emerging issue of distributed computing which can be categorized under the category of NP-Hard/NP-Complete problem. Therefore, it is very much necessary to resolve the above mentioned issues. As all of these schemes are categorized under the category of NP-hard or NP-complete, we should give emphasis on the appropriate scheduling of virtual machines.Because of the flexibility and elasticity characteristics, the popularity of cloud computing is growing day by day. Large quantities of data are generated every day. Hence, it is very much difficult to handle and control those data efficiently.

The healthcare domain is getting very much benefitted because of the advancement of cloud computing. This domain needs vast amount of resources. The healthcare industry is required to be automated in order to cope with the latest trends. Purchase and deploy high-end computer system for various jobs are very costly. All of these above mentioned issues can be resolved through the implementation of cloud technology. Presently, cloud environments are used in order to deploy and execute various software applications more specifically FOSS-based applications. The above mentioned applications can include different cloud components in order to use resources. Horizontal and vertical elasticity is considered as the most attractive property of the SaaS level provided by cloud. Virtual elasticity has the responsibility to either increase or decrease the software resources abilities. On the other hand, horizontal elasticity has the responsibility to replicate or eliminate software instances. Most of the cloud resources include two significant characteristics, those are:-

1. Resources are permitted to be scaled on demand.
2. A single resource can be shared by different FOSS applications.

The cloud infrastructure is mostly customizable in nature. Hence, required software can be installed on different virtual machines. According to the number of tasks of a particular scientific application, the demand of cloud resources may also vary. In order to manage this changing demand of resources, it is necessary to manage the resources effectively. An appropriate resource management process may lead to enhanced resource utilization, improved application performance and decreased the usage expenses. In order to manage these resources more efficiently, it is very much required to predict the usage of cloud resources just like CPU and memory. A large scale data center may include large numbers of computing servers installed by ICT organizations. Cloud computing can be defined as a specific model by which different resources can be accessed by cloud customers. Apart from this, several other applications can be accessed easily. Virtualization process is very important in case of cloud infrastructure.

The virtualization process is implemented with the help of virtual machines in order to construct heterogeneous systems. Every individual host is capable to accommodate virtual machines of variable sizes. All of these virtual machines can be turned on or off any time without any modifications of the actual host. Elasticity is considered as the most vital characteristic of cloud computing technology. The elastic cloud is efficient enough to implement resource changes through allocation and de-allocation of resources automatically.

Cloud computing can be defined as the combination of conventional computing schemes for load balancing. Virtual machine is considered as the most important resource in cloud computing. Both the process of resource scheduling and resource allocation influences the quality of service and also influences the profits of the cloud service provider. The process of resource scheduling has become the most attractive field of research. There have been large numbers of resource scheduling approaches developed since last two decades. With the advancement of global manufacturing, cloud manufacturing has attracted the attention of numbers of different researchers. Cloud manufacturing platform is considered as the most important portion of the cloud manufacturing system. Numbers of different distributed resources provided by various manufacturers can be aggregated. During the cloud manufacturing process, various distributed resources are encapsulated within cloud services. In order to offer best services to the customers, the cloud manufacturing platform must apply a centralized planning and management system. According to numbers of different cases, all of the above mentioned tasks are submitted by customers. These tasks are very much complicated in nature and hence, these are required to split into numbers of subtasks. These subtasks are executed with the help of aggregated distributed resources. With the help of precedence constraints, a complicated relationship is mostly present among the above mentioned subtasks. Again, every individual resource is allowed to satisfy all of the task requests. There are two important limitations, those are:-

1. Scheduling of multiple subtasks
2. Scheduling of subtasks by considering precedence constraints and resource eligibility.

II. RELATED WORKS

K. AlmiÕani, Y. C. Lee and B. Mans focused on resource utilization for scientific workflows in clouds [1]. Because of the vast advancements of cloud technology, the cloud-based applications are also increasing day by day. Costs and performance are considered as the two important factors for the wide acceptance of cloud-based applications. As we all know, most of the scientific workflows are complicated and large-scale. Therefore, the complete process of resource management is also very complex and difficult. There are chances of inefficient utilization of resources. In this piece of research work, they have introduced an advanced resource demand aware scheduling scheme in order to improve the overall resource efficiency. They have termed their presented algorithm as RDAS+. The above presented algorithm has the responsibility to maximize the process of resource utilization through assigning least numbers of resources along with delayed completion time. This algorithm is very much efficient for pay-per-use cloud resources and the optimization process makes the algorithm most feasible in terms of expenses. The complete working procedure of the above proposed algorithm can be divided into following two steps, those are:-

1. Partitioning
2. Resource allocation
3. Task scheduling

This algorithm is tested with the help of five different real world scientific workflows. In future, additional research efforts may be performed in order to enhance the above model.

Y. Liu, W. Wei and H. Xu developed a new multi-resource scheduling scheme in case of hybrid cloud-based large-scale media streaming [2]. Therefore, through merging cloud, CDNs and private data centers, we can develop better approach. In order to transfer vast quantity of media content with the help of a geographically distributed hybrid cloud, they presented an advanced This approach is capable enough to decrease the overall expenses around 60%. Apart from this the average delivery distance can also be decreased by 70%.

Á. L. García, E. F. del Castillo and I. C. Plasencia proposed an advanced cloud scheduler design supporting pre-emptible instances [3]. Every cloud provider has the basic objective to improve the resource utilization through the process of resource provisioning. Most of the commercial providers give emphasis in order to increase their revenues. On the other hand, the scientific and non-commercial providers focus to enhance their infrastructure utilization. Batch systems have the responsibility to permit the data centers in order to fill the resources with the help of backfilling and similar approaches. In case of Iaas cloud, the virtual machines usually give services till you pay for them. In the above mentioned scenario, the mentioned policies can't be applied properly and easily. In this research work, they introduced an advanced scheduling approach for IaaS providers. This method is also proposed to consider premeptible instances. The preemptible instances can be easily terminated by higher priority requests. In the above case, no large change takes place in the current cloud schedulers. The proposed scheduler has the responsibility to implement new cloud usage and payment schemes.

Md. Abd Elaziz, S. Xiong, K. P. N. Jayasena and L. Li emphasized on the task scheduling process in cloud computing that depends upon hybrid moth search algorithm and differential evolution [4]. In this piece of research work, an alternative approach was introduced in order to resolve the issues of cloud task scheduling. It has the prime objective to minimize makespan which is essential to schedule different tasks on various virtual machines. The above presented approach depends upon the extended version of Moth Search Algorithm (MSA) with the help of Differential Evolution. The MSA method is based on the general concept of moths flying towards the light source. They usually fly towards light with the help of two general concepts, those are:- the phototaxis and Levy flights. Here, we can mention that, the exploitation capability is required to be improved significantly. Hence, DE is usually implemented as the local search approach. In future, this approach can further be improved.

Y. Hu, F. Zhu, L. Zhang, Y. Lui and Z. Wang focused on scheduling of manufacturers depending upon chaos optimization scheme in cloud manufacturing [5]. In the present era, the domain of cloud manufacturing technology has become more popular. It has become the prime concern of different researchers in order to implement manufacturer scheduling in cloud manufacturing environment. In the domain of cloud manufacturing, all of the manufacturers are virtualized and digitized. After that, these are aggregated in the cloud data format. Most of the manufacturers are presently transforming their service mode from classical decentral to centralized kind. We can mention here that, manufacturers are unable to respond to the requirements of their customers. The scheduling of manufacturers can directly influence the quality of services. In this piece of research work, they have identified the issues of manufacturer scheduling.

F. Abazari, M. Analoui, H. Takabi and S. Fu introduced multi-objective workflow scheduling algorithm in cloud computing with the help of heuristic approach [6]. The implementation of cloud computing technology is very much essential and popular in the domain of workflow scheduling more specifically scientific workflows. The implementation process of data-intensive workflows in the cloud may give rise to various important factors those play vital role during specification and scheduling process. In the absence of intermediate data security, the chances of information leakage and data modifications are very high. Most of the previously existing scheduling approaches never consider the interaction in between different tasks and their influences on application security needs. In order to resolve the above mentioned issue, they have developed an advanced approach that includes both task security requirements and the interaction in between different tasks. To achieve better security and performance, they proposed an advanced heuristic scheme that depends upon task's completion time and security needs.

Furthermore, they have also introduced a new attack response technique in order to decrease several security threats in the cloud environment.

A. R. Arunarani, D. Manjula and V. Sugumaran performed a thorough survey on various existing task scheduling approaches in the domain of cloud computing [7]. Cloud computing involves different numbers of virtualized resources that makes the process of scheduling very difficult and complicated. In cloud, the customers are allowed to use large numbers of virtualized assets for every individual task. As we all know that, the process of manual scheduling is not at all an efficient solution. The prime objective of task scheduling algorithm is to reduce the amount of time loss and enhance the overall performance significantly. Since years, there have been extensive amount of research works carried out in order to develop an efficient and effective task scheduling algorithm. In this piece of research work, they have performed a thorough survey on different task scheduling approaches. They have also considered and analysed different problems associated with the process of task scheduling.

L. F. Bittencourt, presented an advanced scheduling approach for distributed systems [8]. The process of scheduling is considered as the most important decision making process that involves perfect resource sharing in between various activities. It has to determine the proper execution order for numbers of different resources. With the growing popularity of distributed systems, the numbers of challenges are also increasing. It is very much difficult for the new researchers to understand the relationships in between various scheduling issues. These issues may influence the identification process of new research avenues. In this piece of research work, they have presented an advanced classification scheme in order to resolve the issues of scheduling in the distributed systems. They have performed survey on different scheduling schemes. At last, the future research directions are also identified.

M. C. Calzarossa, introduced an advanced framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty [9]. There are two major advantages of cloud environment, those are:- larger storage capacity and dynamic computing resources. Because of the above mentioned advantages, various data parallel applications are deployed in the cloud. It is very much essential to determine feasible solutions in order to satisfy the required service goals. Because of the unpredictable behavior of cloud performance, the estimation of resources becomes more complicated.In this piece of research work, they presented an advanced scheduling framework in order to manage different uncertainties, performance variability and workload. The above presented framework has the responsibility to permit various cloud users in order to carry out the process of estimation before the actual execution process. They have also identified a new optimization issue where the characteristics are influenced by various uncertain phenomena.

I. Casas, J. Taheri, R. Ranjan, L. Wang ande1 0078zx A. Y. Zomaya proposed an improved genetic algorithm in order to carry out the scheduling of scientific workflows in cloud [10]. Now-a-days, cloud computing is considered as the most essential environment in order to run various kinds of scientific experiments. In order to fulfil the growing requirements, the service providers are required to match or schedule applications along with computing resources. A proper scheduling process of scientific workflows depends upon the capability to analyse all the applications much before the execution. An efficient scheduling algorithm has the responsibility to analyse behaviours of available resources. Apart from this, it is also beneficial to provide different scheduling configuration. It also assists the users in order to choose an optimal configuration for executing workflows. There have been large numbers of schedulers introduced previously. Most of them are meant to execute different complicated applications on cloud environments. But, there is no such algorithm that is capable enough to satisfy each and every characteristic. In this research work, they presented a new scheduler known as GA-ETI. It has the responsibility to identified several previously existing issues and resolve them. The above proposed scheme permits to adapt various kinds of scientific workflows. It has also the responsibility to generate schedules in order to consider the relationships among jobs and necessary data. This scheduling algorithm is basically an interface among cloud user and cloud service provider. At first, it receives different applications, then it analyses them. Again, it is responsible for efficient task distribution process.

D. Chaudhary and B. Kumar emphasized on cloudy GSA for load scheduling in cloud computing [11]. The scheduling process of load and data plays vital role during the process of resource utilisation from a particular cloudlet to a completely different cloudlet. The domain of cloud computing is incremental in nature and it also provides the power of distributed computing with the help of a Virtual method. The process of resource allocation is very important during the optimal handling of load scheduling issue with the help of Static and meta-heuristic techniques. The gravitational search algorithm is basically a nature inspired meta- heuristic Optimisation approach. This technique is usually implemented in order to resolve the issues of load scheduling in the cloud environment. It includes the basic concepts of Newton's gravitational law. In this piece of research work, they presented in your optimal load scheduling technique which is also known as cloudy-GSA. It has the responsibility to decrease the transfer time and total expenses during the process of scheduling the cloudlets to the virtual machines. Here we can conclude that, type of proposed algorithm is much better as compared to traditional approaches in terms of performance.

S. G. Domanal and G. R. M. Reddy introduced an advanced cost optimized scheduling for spot instances in heterogeneous cloud environment [12]. In this piece of research work, they presented a new and cost optimised scheduling scheme for a bag of tasks on virtual machines they have implemented artificial neural network in order to predict the future scope of spot instances. On-Demand and Spot are considered as the key instances those are generated by the customers. In this research work, they have considered these instances during the process of cost optimization. The prime objective of the above presented approach is to

utilize the resources efficiently. Apart from this, they have also emphasized on the cost optimization process in a heterogeneous environment. By analysing the outcomes of the above proposed method, we can conclude that, this approach is much better as compared to other pre-existing approaches. In future, additional research works can be performed in order to extend this method.

N. Dordaie and N. J. Navimipour developed an efficient hybrid particle swarm optimization and hill climbing technique in order to carry out the process of task scheduling in the cloud environments [13]. The process of task scheduling is considered as one of the most vital issues case of heterogeneous environments.

K. Dubey, M. Kumar and S. C. Sharma proposed the modified version of HEFT algorithm in order to carry out the process of task scheduling in cloud environment [14]. Heterogeneous earliest finish time e sim capable to tribute the task properly. In this research paper, they have modified and extended the in order to distribute the workload in between different processors effectively. It has the ability to decrease the makespan time of different applications.

P. K. Sahoo and C. K. Dehury developed advanced data and CPU-intensive job scheduling approach for healthcare cloud [15]. Cloud computing environment is usually used to enhance the operational efficiency of different processes. Apart from this, it also has the responsibility to provide various services to the cloud users. Presently, the healthcare domain is transferring the classical business model to the cloud based business model. It can satisfy the resource requirement of various healthcare applications. There are different kinds of jobs starting from basic patient record extraction to complicated biomedical image analysis. Large amount of resources are provided by the cloud in the healthcare domain in order to perform complicated and time-consuming operations. In this research paper, an advanced scheduling framework is developed in order to perform proper distribution of healthcare jobs. They suggested an autonomous meta-director framework in order to find the green data Center's most energy-efficient route through a linear programming approach[24]. It calculates the energy efficiency of the system and evaluates data collected to enable the installation of an efficient virtual machine[25]. Work at different random intervals is provided with different load conditions during server operation. However, potential VMs and task information useful for assigning tasks to the corresponding VM could not be used. Efficiency in finding the necessary VM instance and capacity, depend on the load balancing mechanism and the size of the tasks between the VM.

III. PROPOSED APPROACH

Improved PSO for optimal feature selection

Initializing particles with cloud services, number of iterations, velocity, number of particles etc.

Compute hybrid velocity and position for each particle in 'd' dimensions using the following equations

$$v(d+1,i) = \psi.[\omega(d,i).v(d,i).\theta_{chaos1}(pBest_i - X(d,i)) + \theta_{chaos2}(gBest_i - X(d,i))]$$

$$X(d+1,i) = X(d,i) + v(d+1,i)$$

ψ is the convergence factor computed as

$$\psi = \frac{2*(\theta_{c1}+\theta_2)}{|2-(\theta_{c1}+\theta_{c2}) - \sqrt{(\theta_{c1}+\theta_{c2})^2 - 4(\theta_{c1}+\theta_{c2})}|}$$

where $\theta_{c1}, \theta_{c2} \in$ random numbers

1) $\delta_p -> \{\delta_{p_1}, \delta_{p_2}, \delta_{p_3} ... \delta_{p_m}\}$ represents the set of physical machines in cloud environment.
2) $\chi_C -> \{\chi_{C_1}, \chi_{C_2}, \chi_{C_3} ... \chi_{C_N}\}$ represents the set of data centres with resources in cloud environment.

   $R_i = \{RAM, NBW, CPU, PROTO, ACCPOLI\}$

   Where NBW: Network bandwidth, PROTO: PROTOCOL, ACCPOLI: Access policies.
3) $\phi_{VM} -> \{\phi_{VM_1}, \phi_{VM_2}, \phi_{VM_3} ... \phi_{VM_K}\}$ represents the set of cloud instances in cloud environment. Let $r_i \in R_i$ be the set of resources of all cloud instances.

   $r_i \in R_i \rightarrow \{r_i^{RAM}, r_i^{NBW}, r_i^{CPU}, r_i^{PROTO}, r_i^{ACCPOLI}\}$

For each physical machine PM(δ_p), the Boolean bit vector $B_j = \{B_{j1}, B_{j2}, B_{j3} ..... B_{jr}\}$,

$B_{jr} = 1$ if ϕ_{VMj} assigns δ_{pr}.

$B_{jr} = 0$ otherwise

Similarly the status of physical machine is denoted by { PB_m} where

PB_m=1 $(\exists \phi_{VM_i} \in \phi_{VM} / B_{mi=1})$

In this optimized model, inertia weight is computed as

$$\omega(d,i) = \omega_{max} - (I_{current} / I_{max}).(\omega_{max} - \omega_{min})$$

ω_{max} : max inertia

ω_{min} : min inertia

I_{max} : max iteration

Step 3: Computing fitness value using ortho chaotic gauss randomization measure.

In this proposed PSO model, a random value between 0 to 1 is selected using the following equation as

$$R_i = \frac{1}{\sqrt{2\pi\sigma_X}} e^{-\frac{(X-\mu_X)^2}{\sigma_X{}^2}}$$

$$K = 1, 2 ... \text{iterations}$$

Proposed Objective Functions for Non-linear constraint programming:

The objective function for the proposed resource optimization in cloud computing environment is given by

1) $$\text{Min} \sum_{i=1}^{N} \log\{PB_i\} \text{ and Max } \sum_{j=1}^{M} \{B_j\}$$

$$\text{with } PB_i = 1((\exists \phi_{VM_i} \in \phi_{VM} / B_{ij=1})$$

$$PB_i = 0; \text{otherwise} \quad \text{---(1)}$$

2) $$\text{Max } \sum_{j=1}^{M} \{B_j\}$$

$$B_{jr} = 1 \text{ if } \phi_{VMj} \text{ assigns } \delta_{pr}.$$

$$B_{jr} = 0 \text{ otherwise} \quad \text{-----(2)}$$

S.t

$$\sum_{p=1}^{|V|} r_p^{RAM}.PB_{ip} \le C_i^{RAM}; 1 \le i \le N$$

$$\sum_{p=1}^{|V|} r_p^{NBW}.PB_{ip} \le C_i^{NBW}; 1 \le i \le N$$

$$\sum_{p=1}^{|V|} r_p^{CPU}.PB_{ip} \le C_i^{CPU}; 1 \le i \le N$$

$$\sum_{p=1}^{|V|} r_p^{PROTO}.PB_{ip} \le C_i^{PROTO}; 1 \le i \le N$$

$$\sum_{p=1}^{|V|} r_p^{ACCPOLI}.PB_{ip} \le C_i^{ACCPOLI}; 1 \le i \le N$$

Energy Constraint:

Energy consumption of computing resources such as job computation , server storage, server capacities can be computed using the power model. The linear relationship among the resource utilisation and power consumption is given as:

$$PU(CPU_i) = \alpha P_{max}(CPU_i) + (1-\alpha).P_{max}(CPU_i).PU(CPU_i)$$

$PU(CPU_i)$ is the power utilisation of cloud instance i.

$P_{max}(CPU_i)$ is the maximum power utilisation of ith instance , when the cloud server is fully allocated.

α is the fraction of scaling parameter to the idle server (0-1).

In multi-core cloud environment the total utilization of all cloud instances should be minimized using the constraint programming as

$$\mathrm{SPU}\left(\mathrm{CPU}_i\right)=\sum_{i=1}^{N}\mathrm{PU}\left(\mathrm{CPU}_i\right)$$

$$\min \mathrm{SPU}\left(\mathrm{CPU}_i\right)=\min\sum_{i=1}^{N}\mathrm{PU}\left(\mathrm{CPU}_i\right)$$

Total power utilisation of al lthe cloud instances in the available data centre is given by

$$\mathrm{TP}=\alpha \mathrm{P}_{\max}(\mathrm{CPU}_i)+(1-\alpha)\mathrm{SPU}\left(\mathrm{CPU}_i\right)$$

The energy consumption of all the cloud server over a time period T is given by

$$\mathrm{TP}_i=\int_0^{\mathrm{T}}\mathrm{TP}(t)\log(\mathrm{TP}(t))dt \qquad \text{----(3)}$$

Proposed feature selection fitness measure is given as

$$\mathrm{Fitness}_i=w_1.\mathrm{TP}_i+w_2.(1-\frac{\sum_{i=1}^{|F|}f_i}{N_f})$$

where $w_1, w_2 \in R_i$

f_i is the flag value 1 or 0 . '1' represents selected service , '0' non-selected resource.

N_f represents number of services.

Step 4: For each particle compute its fitness value and compute classification accuracy in the previous step.

Step 5: Update particle velocity, position, global best and particle best according to the fitness value conditions.

Step 6: This process is continuous until max iteration is reached. Otherwise go to step 2.

The main objective of the proposed resource optimization model is to minimize the number of physical machine required to host all the instances.

Algorithm Steps:

1. Connect to cloud environment using credentials with available data centre zones.
2. Initialization of 'k1' number of available data centre zones DC[].
3. Initialization of 'k2' number of physical machines PM[].
4. Initialization of 'k3' number of virtual instances VI[].
5. For each user request of instance VI[i]
6. Do
7. Search PM in the available data centres DC[].
8. Search for instance capacity and its properties in the physical machine PM[].
9. Check the optimization functions for the data centres, Physical machine, virtualmachines and energy computation using (1),(2),(3).
10. Estimating the best servers using the proposed probability estimation formula. The minimum and maximum bound limits are used to decide the workload usage of each instance in the virtual machine as:

$$\text{Lower bound limite}:\mu_{VI[]}-\lambda\sigma_{VI[]}$$

$$\text{Upper bound limite}:\mu_{VI[]}+\lambda\sigma_{VI[]}$$

$$\text{Bounded limit}:\mu_{VI[]}$$

$$\lambda=\frac{1}{\sigma_{VI[]}.\sqrt{2.\pi}}e^{-\frac{(VI[i]-\mu_{VI[]})^2}{2.\sigma_{v[]}^2}}$$

IV. EXPERIMENTAL RESULTS

For experimental results, homogeneous and heterogeneous virtual machines have been used that consist of five instances with specified number of resources and data. To compare the performance of the existing models with the proposed model, three

metrics have been used to evaluate the load balancing, energy consumption and runtime of the virtual instances and available resources. For virtual machine, kernel based VM has been installed in each server node in cloud environment. Different operating systems such as Red hat linux, Centos, Windows etc are used to evaluate the performance of each virtual machine in the cloud environment. For experimental evaluation, Amazon aws cloud environment is used to test the optimial resource allocation and to test the efficiency of the proposed model to the existing models. All experimental results are performed using the Java programming environment with real-time amazon aws third party libraries.

The initialization of the cloud instances and its resources are summarized below:

Instance results :[{ReservationId: r-04c6d023b80074c16,OwnerId: 355850546694,Groups: [],GroupNames: [],Instances: [{InstanceId: i-041824179e09ecdb8,ImageId: ami-d0f506b0,State: {Code: 80,Name: stopped},PrivateDnsName: ip-172-31-4-27.us-west-2.compute.internal,PublicDnsName: ,StateTransitionReason: User initiated (2017-06-01 06:53:35 GMT),KeyName: aws,AmiLaunchIndex: 0,ProductCodes: [],InstanceType: t2.micro,LaunchTime: Thu Jun 01 11:13:49 IST 2017,Placement: {AvailabilityZone: us-west-2c,GroupName: ,Tenancy: default},Monitoring: {State: disabled},SubnetId: subnet-22c5077b,VpcId: vpc-65a71100,PrivateIpAddress: 172.31.4.27,StateReason: {Code: Client.UserInitiatedShutdown,Message: Client.UserInitiatedShutdown: User initiated shutdown},Architecture: x86_64,RootDeviceType: ebs,RootDeviceName: /dev/xvda,BlockDeviceMappings: [],VirtualizationType: hvm,ClientToken: kDyVA1496295829374,Tags: [{Key: Name,Value: PythonOpencv}],SecurityGroups: [{GroupName: ssh_http,GroupId: sg-42e0c139}],SourceDestCheck: true,Hypervisor: xen,NetworkInterfaces: [{NetworkInterfaceId: eni-4b466d44,SubnetId: subnet-22c5077b,VpcId: vpc-65a71100,Description: ,OwnerId: 355850546694,Status: in-use,PrivateIpAddress: 172.31.4.27,PrivateDnsName: ip-172-31-4-27.us-west-2.compute.internal,SourceDestCheck: true,Groups: [{GroupName: ssh_http,GroupId: sg-42e0c139}],Attachment: {AttachmentId: eni-attach-73661910,DeviceIndex: 0,Status: attached,AttachTime: Thu Jun 01 11:13:49 IST 2017,DeleteOnTermination: true},PrivateIpAddresses: [{PrivateIpAddress: 172.31.4.27,PrivateDnsName: ip-172-31-4-27.us-west-2.compute.internal,Primary: true,}]}],EbsOptimized: false}]}, {ReservationId: r-0bef7677d9b7f37ad,OwnerId: 355850546694,Groups: [],GroupNames: [],Instances:

[],Instances: [{InstanceId: i-0b1dc4321d02370d5,ImageId: ami-58998521,State: {Code: 80,Name: stopped},PrivateDnsName: ip-172-31-45-202.us-west-2.compute.internal,PublicDnsName: ,StateTransitionReason: User initiated (2018-01-15 11:51:46 GMT),KeyName: gskpair,AmiLaunchIndex: 0,ProductCodes: [],InstanceType: t2.micro,LaunchTime: Mon Jan 15 17:19:32 IST 2018,Placement: {AvailabilityZone: us-west-2b,GroupName: ,Tenancy: default},Monitoring: {State: disabled},SubnetId: subnet-63e36d06,VpcId: vpc-65a71100,PrivateIpAddress: 172.31.45.202,StateReason: {Code: Client.UserInitiatedShutdown,Message: Client.UserInitiatedShutdown: User initiated shutdown},Architecture: x86_64,RootDeviceType: ebs,RootDeviceName: /dev/sda1,BlockDeviceMappings: [{DeviceName: /dev/sda1,Ebs: {VolumeId: vol-0016c75b7283d6c37,Status: attached,AttachTime: Mon Nov 20 20:11:49 IST 2017,DeleteOnTermination: true}}],VirtualizationType: hvm,ClientToken: ,Tags: [{Key: Name,Value: GSKSPARKJAVA}],SecurityGroups: [{GroupName: launch-wizard-2,GroupId: sg-d48560a8}],SourceDestCheck: true,Hypervisor: xen,NetworkInterfaces: [{NetworkInterfaceId: eni-bfb4c79f,SubnetId: subnet-63e36d06,VpcId: vpc-65a71100,Description: ,OwnerId: 355850546694,Status: in-use,PrivateIpAddress: 172.31.45.202,PrivateDnsName: ip-172-31-45-202.us-west-2.compute.internal,SourceDestCheck: true,Groups: [{GroupName: launch-wizard-2,GroupId: sg-d48560a8}],Attachment: {AttachmentId: eni-attach-c533b525,DeviceIndex: 0,Status: attached,AttachTime: Mon Nov 20 20:11:49 IST 2017,DeleteOnTermination:

true},PrivateIpAddresses: [{PrivateIpAddress: 172.31.45.202,PrivateDnsName: ip-172-31-45-202.us-west-2.compute.internal,Primary: true,}]},EbsOptimized: false}]}, {ReservationId: r-0eeb2df62550d7c0f,OwnerId: 355850546694,Groups: [],GroupNames: [],Instances: [{InstanceId: i-047f013f8b88ef80d,ImageId: ami-82ccade2,State: {Code: 80,Name: stopped},PrivateDnsName: ip-172-31-33-232.us-west-2.compute.internal,PublicDnsName: ,StateTransitionReason: User initiated (2017-06-02 05:33:39 GMT),KeyName: aws,AmiLaunchIndex: 0,ProductCodes: [],InstanceType: t2.micro,LaunchTime: Fri Jun 02 11:03:11 IST 2017,Placement: {AvailabilityZone: us-west-2b,GroupName: ,Tenancy: default},Monitoring: {State: disabled},SubnetId: subnet-63e36d06,VpcId: vpc-65a71100,PrivateIpAddress: 172.31.33.232,StateReason: {Code: Client.UserInitiatedShutdown,Message: Client.UserInitiatedShutdown: User initiated shutdown},Architecture: x86_64,RootDeviceType: ebs,RootDeviceName: /dev/sda1,BlockDeviceMappings: [],VirtualizationType: hvm,ClientToken: poZBQ1496231627480,Tags: [{Key: Name,Value: RStudioWebGSK}],SecurityGroups: [{GroupName: ssh_http,GroupId: sg-42e0c139}],SourceDestCheck: true,Hypervisor: xen,NetworkInterfaces: [{NetworkInterfaceId: eni-0787452d,SubnetId: subnet-63e36d06,VpcId: vpc-65a71100,Description: ,OwnerId: 355850546694,Status: in-use,PrivateIpAddress: 172.31.33.232,PrivateDnsName: ip-172-31-33-232.us-west-2.compute.internal,SourceDestCheck: true,Groups: [{GroupName: ssh_http,GroupId: sg-42e0c139}],Attachment: {AttachmentId: eni-attach-e2215c0b,DeviceIndex: 0,Status: attached,AttachTime: Wed May 31 17:23:48 IST 2017,DeleteOnTermination: true},PrivateIpAddresses: [{PrivateIpAddress: 172.31.33.232,PrivateDnsName: ip-172-31-33-232.us-west-2.compute.internal,Primary: true,}]},EbsOptimized: false}]}]You have 4 Amazon EC2 instance(s) running.

The complexity of proposed model to the existing models depends on the number of physical machines and virtual machines. In the experiments, different number of physical machines and virtual machines are used to measure the improved of the proposed model to the existing models. The maximum and minimum bound limits of the physical machines and virtual machines computed in the proposed model are listed in table 1 and table 2.

The below graph represents the different cancer dataset processing in each cloud instance server.

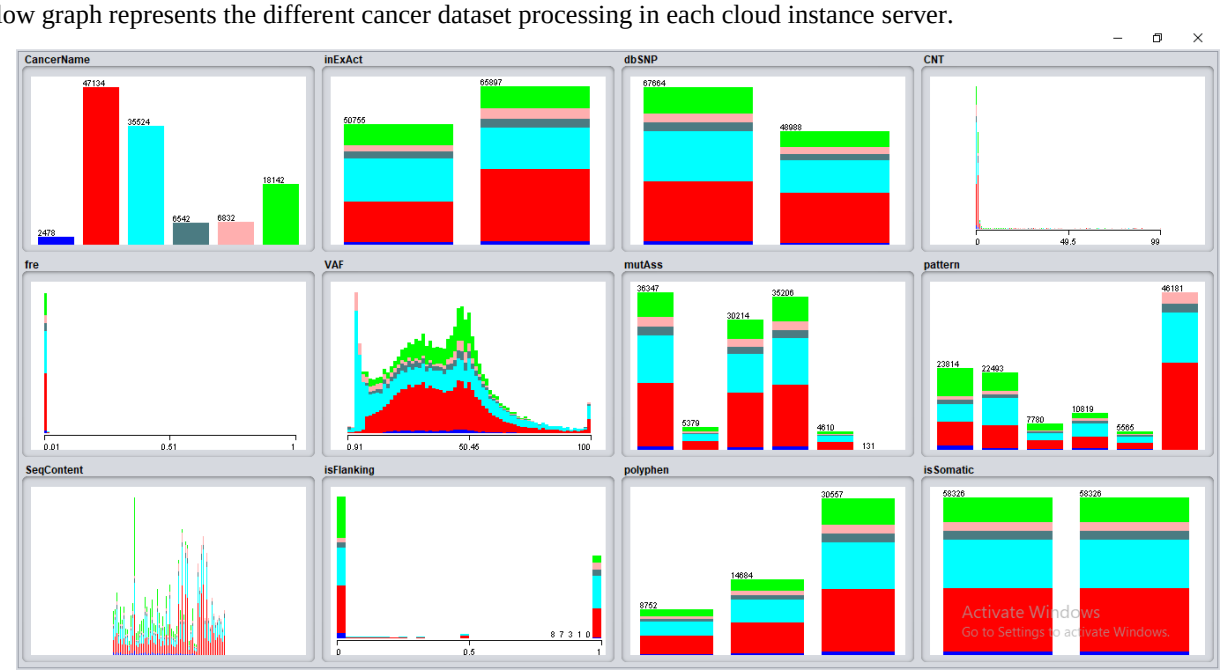

Figure 1: Computational analysis of proposed model based on different datasets

Bounds	CPU(Hz)	RAM(MB)	BANDWIDTH(Kbps)
Lower bound	1500	1000	1500
Upper bound	9500	9000	9000

Table 1: Physical machine bound limits

Bounds	CPU(Hz)	RAM(MB)	BANDWIDTH(Kbps)
Lower bound	350	400	200
Upper bound	3500	3500	800

Table 2: Virtual machine bound limits

Table 3: **Comparative analysis of resource allocation and runtime of the proposed model to the existing models.**

Model	Avg Allocated Resources	Runtime(ms)
Papagianni et al, 2013	15	9743
Buyya et.al, 2016	12	8475
Jenning et.al, 2015	13	9164
Poola et.al,	16	7935
Multiobjective	9	7395
Proposed	9	7143

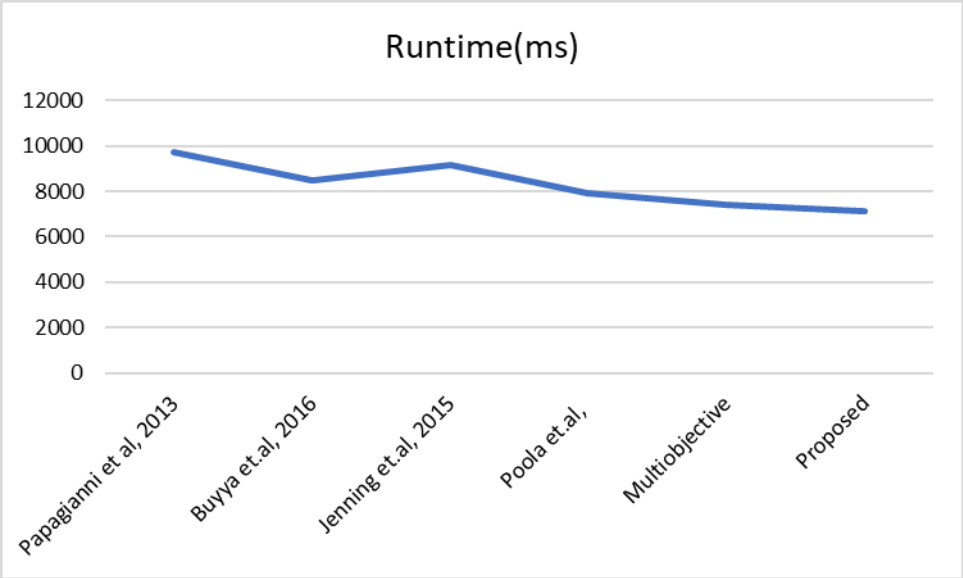

Figure 2 : Comparative analysis of runtime of the proposed model to the existing models.

V. CONCLUSION

In this paper, different load balancing functions are integrated by using cloud optimization functions. These models are designed and implemented to test the resource allocation using the available physical machines and virtual instances. Load balance can improve quality of service (QoS) measurements, including response time, cost, performance and use of resources. In this work, a novel load balancing algorithm is implemented to improve the cloud service load balancing. In order to optimize the delivery of cloud services, the load balance is important between virtual machines at minimum paid costs and overall service delivery time. In order to improve the scheduling process of load-balancing in the cloud environment, many traditional models are used to optimize the load balance. However, the main problem to the cloud service provider's is optimizing cloud service parameters such as reliability, flexibility, time limits and the task refusal rate. A dynamic algorithm is required for the cloud service provider to plan work which will reduce time while increasing the cloud resources use ratio and comply with the user's specific QoS

parameters. The proposed framework is based on hybridizing heuristic techniques with metaheuristic algorithm in order to achieve its optimum performance in the load balancing process. Experimental results proved that the present load-balancing model has better performance than the traditional load balancing approaches on various cloud resources.

VI. REFERENCES

[1] K. AlmiÕani, Y. C. Lee and B. Mans, On EfÞcient Resource Use for ScientiÞc Workßows in Clouds, computer networks.

[2] Y. Liu, W. Wei and H. Xu, Efficient multi-resource scheduling algorithm for hybrid cloud-based large-scale media streaming, Computers and Electrical Engineering 75 (2019) 123–134.

[3] Á. L. García, E. F. del Castillo and I. C. Plasencia , Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution, Knowledge-Based Systems.

[4] Md. Abd Elaziz, S. Xiong, K. P. N. Jayasena and L. Li , Y. Hu, F. Zhu, L. Zhang, Y. Lui and Z. Wang, Scheduling of manufacturers based on chaos optimization algorithm in cloud manufacturing, Robotics and Computer Integrated Manufacturing 58 (2019) 13–20.

[5] Y. Hu, F. Zhu, L. Zhang, Y. Lui and Z. Wang, Scheduling of manufacturers based on chaos optimization algorithm in cloud manufacturing, Robotics and Computer Integrated Manufacturing 58 (2019) 13–20

[6] F. Abazari, M. Analoui, H. Takabi and S. Fu, MOWS: Multi-Objective Workßow Scheduling in Cloud Computing based on Heuristic Algorithm, simulation modeling practice and theory.

[7] A. R. Arunarani, D. Manjula and V. Sugumaran, Task scheduling techniques in cloud computing: A literature survey, Future Generation Computer Systems 91 (2019) 407–415.

[8] L. F. Bittencourt, A. Goldman, E. R. M. Madeira, N. L. S. da Fonseca and R. Sakellariou, Scheduling in distributed systems: A cloud computing perspective, Computer Science Review 30 (2018) 31–54

[9] M. C. Calzarossa, M. L. Della Vedova and D. Tessera, A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty, Future Generation Computer Systems 93 (2019) 212–223.

[10] I. Casas, J. Taheri, R. Ranjan, L. Wang andelzx A. Y. Zomaya, GA-ETI: An Enhanced Genetic Algorithm for the Scheduling of Scientific Workflows in Cloud Environments,

[11] D. Chaudhary and B. Kumar, Cloudy GSA for load scheduling in cloud computing, applied soft computing.

[12] S. G. Domanal and G. R. M. Reddy, An efficient cost optimized scheduling for spot instances in heterogeneous cloud environment, future generation computer systems.

[13] N. Dordaie and N. J. Navimipour, A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments, ICT express

[14] K. Dubey, M. Kumar and S. C. Sharma, Modified HEFT algorithm for task scheduling in cloud environment.

[15] P. K. Sahoo and C. K. Dehury, Efficient data and CPU-intensive job scheduling algorithms for healthcare cloud, Computers and Electrical Engineering 68 (2018) 119–139