January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

A Novel Resource Optimization Model on Realtime Cloud Computing using Bayesian Estimation

K. Siva Rama Krishna [1], Dr. Mohammed Ali Hussain [2]

[1]Research Scholar, Dept. of CSE, ShriVenkateswara University, Uttar Pradesh, India.
[2] Professor, Dept. of Electronics and Computer Engineering, KLEF (Deemed to be University), Guntur Dist.,A.P, India.
[1] sivaramkosuru@gmail.com, [2] alihussain.phd@gmail.com

Article Info
Volume 82
Page Number: 13395 - 13404
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 24 February 2020*

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 bayesian scoring function based PSO model 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.

Keywords:*Bayesian cloud resource score, cloud virtualization.*

I. INTRODUCTION

Data centers host several apps, and information centres use virtualization to multiplex apps and handle funds to decouple apps from the hardware on which they operate. A range of advantages come from virtualization. It allows a versatile distribution of physical assets to virtual apps where virtual mapping and the quantity of funds for every implementation can be dynamically changed in order to adapt to evolving app workloads. Virtualization also allows the multi-tenancy sharing of a physical server by various cases of virtualized apps ("tenants"). Multi-tenancy information centers enable apps to be consolidated, packaged into small servers and operating costs to be reduced. Virtualization will also simplify application replication and scaling.

Cloud problems and temporary access frequently translate into virtual computers. In the mechanism of reduction in general cost and recovery of their big capital expenses, data center providers aim to maximize the use of their servers because inactive computers are reflective of slumping expenses. For example, the virgin computers generally refer to

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

temporary virtual machines as servers provide server abstraction in application modes. However, there are many difficulties to attaining elevated server and data center usage.

The cloud computing utility system encourages consumers to instant cloud services only if required (such as cloud VMs). The information centers are used to host apps with distinct positions and objectives and Cluster leadership software can give priority over data-center resources access. This contributes to a large level of dynamism in the workload, and therefore information center utilization. For example, high-priority user-facing interactive services may preempt lower-priority batch jobs. In the event of resource disputes between apps, the joint approach is to prevent the request with less importance and provide its funds to those with greater priorities[32, 201]. The use of idle assets in a data center could also lead to temporary accessibility for apps operating on old idle assets[237]. As a consequence of planning measures to maximize the use of information centers ' resources, transition occurs at information centres. Grid computing was established in the middle of the 90s. For grid computation facilities, Ian Foster has incorporated distributed, object-oriented programming and Web facilities. The definition explains that the Gird is a group of computers that are loosely linked and networked and are worried about the fact that "the Grid is a kind of parallel and distributed system enabling the sharing, selection and aggregation of geographical autonomous resources, dynamically during operations depending on the availability, capability, performance, costs and quality of service requirements of the users." The enormous work can be split into several subjobs and each runs on large computers.

Software As a Service (SaaS) is a model for shipment of software that provides easy apps such as a Internet browser through a straightforward interface. The consumers do not have the cloud infrastructure that is existing, including networks, servers, OS, storage, platform, etc. This model also eliminates the installation and execution of the implementation on local pcs. The Service Infrastructure (IaaS) offers customers with processing, storage, networks and other basic computer infrastructure. IaaS consumers can install an arbitrary, dynamically-scalable implementation, hardware and operating systems on their infrastructure. 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.

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

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

willing to enhance the process of resource utilization and throughput along with their monetary profits.

Compared to mobile and cloud computation, cloud computing is the most common unique field. The concept is the most appealing characteristic of the Cloud: resource virtualization. In order to execute the method effectively, the word "computation" implies performing multiple duties in several virtual machines. The computing powers of networks also increase every day with the development of data and technology. In order to achieve greater access, we can also handle big amounts of heterogeneous assignments. In the main, pay-as-you-go or pay-as-you-use systems are implemented in the cloud for access to resources. In addition, it has the duty to assign clouds or assignments to virtual machines on a dynamic basis. Dynamic distribution can be readily and effectively achieved by applying the load planning algorithm on the cloud. It is responsible for optimizing performance, reducing running and waiting times, reducing transfer time and lowering computer cost. A developing issue of distributed computation can be classified under the classification NP-Hard / NP-Complete problem with virtual machine planning and use. The above-mentioned questions therefore need to be resolved very much. Since all of these systems are classified as NP hard or NP complete, we should emphasize that virtual machines are properly scheduled. Because of their flexibility and elasticity, the popularity of cloud computation increases day after day. Every day, large amounts of information are produced. Therefore, handling and monitoring these information effectively is very hard.

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.

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. 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]. 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

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

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. Cloud applications provide large data centres, consisting of tens of thousands of servers, storage and networking devices, and the necessary power and cooling infrastructure, with the computing, network and storage resources used. Cloud computing features and flexibility, such as pay-as-you-go pricing, resource elastic scaling, and low cost are possible by carefully managing and assigning data centres. In this section, we examine how the challenges of providing cloud applications with flexibility and low cost are addressed in terms of management of data centers and how these challenges are addressed by operators of data centers.

The consumer of IaaS receives associated programs and information and transfers the results to the laptop of the vendor. The equipment is virtual, versatile, scalable and user-friendly. For example: Amazon EC2, IVPC, IBM Blue Cloud, FlexiScale, Eucalyptus, Joyent, Rackspace, etc. Applications are provided.

Data service concerns customer access from multiple sources to private information in different sizes. These distant information can be run on a local computer just as well. Data software products include Amazon S3, SimpleDB, SQS and Microsoft SQL. The link between cloud users, cloud services and cloud suppliers is shown in Figure 2.3.

Software, platform, storage, and computation assets can be used as pay-as-you-go facilities straight from clients fitted with fundamental phones, Internet, and internet browsers. If an internet protocol connection is established, cloud services can be shared within any one of the service layers. PaaS consumes IaaS products, for instance, and in the meantime provides SaaS platform facilities. At the very bottom of the datacenter is a set of easy guidelines to help find a alternative to a issue: a hardware device and software product like cloud specific operating systems and multi-core processors, networks, drives, etc.

It consists of three components: input, process, output. The entry is an array of parameters.

The method contains descriptive, controllable and repeatable methods for achieving the objective by means of input parameters. The issue is a consequence of the output. The algorithm is a technique by which duties can be accessed, combined or assigned to processors in particular for planning. Generally speaking, there is no ideal algorithm for planning because planning goals can conflict. A nice scheduler makes a appropriate compromise, or combines scheduling algorithms in various apps.

Depending on the algorithm used, a problem can be solved in seconds, hours or even years. An algorithm's effectiveness is assessed by the moment needed to run it.If all feasible alternatives are listed and contrasted one by one for optimization problems, the appropriate solution can be chosen. The exponential time complexity in the worst scenario is the exact listed algorithms. In a weak sense however, some NP-hard problems can be overcome by the pseudopolynomial algorithm, when the number is small in one instance, and the time complexity is limited by a polynomial expression in the input size and maximum number.

Moreover, there is another type of listing, called explicit listing, which assesses all feasible alternatives without explicitly listing them all. Dynamic programming is a viable implicit listing technique for solving issues with combination optimization. It splits a issue into several phases and requires a choice at each point which will affect the choices to be taken at the subsequent phases. The

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

amount and complexity of the programming algorithm is therefore exponential, as the main feature of the leadership of resources, the service planning makes cloud computing distinct from other computing paradigms. The programming feature is a major feature of the leadership of resources. Centralized cluster scheduler seeks to improve general system efficiency while the distributed grid system scheduler seeks to improve the efficiency of particular users. Cloud computing planning is much more complex compared to them. On the one side, a centralized scheduler is required, as each cloud supplier that offers facilities to customers without regard to the host facilities, has a separate datacenter. On the other side, the distributed scheduler is also indispensable since business determines that cloud computing needs to meet the QoS demands of global clients. Md. AbdElaziz, 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]. 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.

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. 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. M. C. Calzarossa, introduced an advanced framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty [9]. 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]. 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. 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. 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

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

scheduling is considered as one of the most vital issues case of heterogeneous environments. T 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.

III.PROPOSED APPROACH

KL-Divergence

$$Dist_{KL} = \sum_{i=1}^{L} p_i \log \frac{p_i}{q_i}$$

where p_i, q_i are the KL divergence parameters.

Rényidivergence(

$$Dist_{KL} = \frac{1}{\alpha - 1} \sum_{i=1}^{L} p_i \log \frac{p_i^{\alpha}}{q_i^{\alpha - 1}}$$

where p_i, q_i are the divergence parameters.

$0 \le \alpha \prec \infty$

Jensen–Shannon divergence :

Measure the similarity between two network traffic distributions. P and Q.

$$JS = 0.5(Dist_{KL}(P, M) + Dist_{KL}(Q, M))$$

Where M=0.5*(P+Q)

$$NaHiD = \frac{|P(i) - Q(i)|}{\|\mu_P - \sigma_P| - P(i)| + \|\mu_Q - \sigma_Q| - Q(i)|}$$

Where P,Q are network traffics.

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

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

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

$\delta_p -> \{\delta_{p_1}, \delta_{p_2}, \delta_{p_3} ... \delta_{p_m}\}$ represents the set of physical machines in cloud environment.

$\chi_C -> \{\chi_{C_1}, \chi_{C_2}, \chi_{C_3} ... \chi_{C_N}\}$ 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.

$\phi_{VM} -> \{\phi_{VM_1}, \phi_{VM_2}, \phi_{VM_3} ... \phi_{VM_K}\}$ 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

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

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

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

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

$K = 1, 2 \ldots$ 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) $$\text{Min} \sum_{i=1}^{N} \log\{PB_i\} \text{ and Max} \sum_{j=1}^{M} \{B_j\}$$

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

$PB_i = 0;$ otherwise ---(1)

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

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

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

S.t

Attribute selection measure for attack detection=A SMAD$(D_i) = \text{Max}\{\rho_1, \rho_2\}$

Conditional Entropy of A_2 on A_1 : $CE(A_2[]/A_1[]) = \sum P(A_1[], A_2[]).\log(\frac{P(A_1[])}{P(A_1[], A_2[])})$

Conditional Entropy of A_1 on A_2 : $CE(A_1[]/A_2[]) = \sum P(A_1[], A_2[]).\log(\frac{P(A_2[])}{P(A_1[], A_2[])})$

$$\rho_1 = \frac{-CE(A_2[]/A_1[])^3}{(\sum A_1[]_i)^3 \text{Chi\_square}(D_i)^3}$$

$$\rho_2 = \frac{-CE(A_2[]/A_1[]).CE(A_1[]/A_2[])}{\sum A_1[]_i)^3 \sqrt{\text{Chi\_square}/(N(m-1)}}$$

N = total observations

m = minimum(#rows, #columns)

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

$$SPU(CPU_i) = \sum_{i=1}^{N} PU(CPU_i)$$

$$\min SPU(CPU_i) = \min \sum_{i=1}^{N} PU(CPU_i)$$

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

$$TP = \alpha P_{max}(CPU_i) + (1-\alpha) SPU(CPU_i)$$

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

$$TP_i = \int_0^T TP(t) \log(TP(t)) dt \quad \text{----(3)}$$

Proposed feature selection fitness measure is given as

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

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.

Published by: The Mattingley Publishing Co., Inc.

TEST Engineering & Management

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

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:

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

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

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

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

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:

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

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.

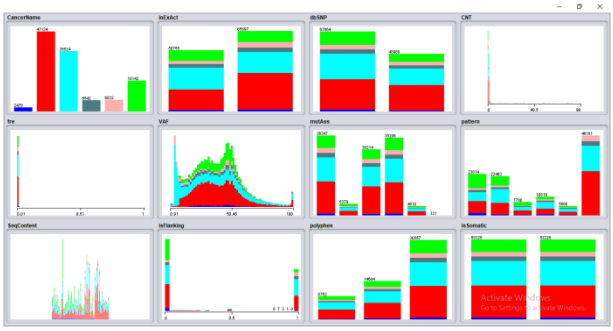

Figure 1: Computational analysis of proposed model based on different datasets

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 Bayesian	8	6346

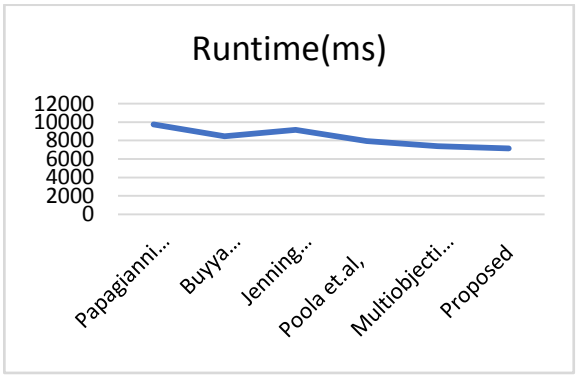

Figure 2 : Comparative analysis of runtime of the proposed model to the existing models.

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

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13395 - 13404

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 bayesian scoring function based PSO model 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.

REFERENCES

1. K. AlmiÕani, Y. C. Lee and B. Mans, On EfÞcient Resource Use for ScientiÞcWorkßows 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. AbdElaziz, 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

Published by: The Mattingley Publishing Co., Inc.