Cloud Task Scheduling Based on a Two Stage Strategy using KNN Classifier

M.Deepika[1], Mr.S.Prabhu[2]

[1]PG Scholar, [2]Assistant Professor, Department of Computer Science and Engineering, Nandha Engineering College, Erode,Tamil Nadu

ABSTRACT - *In Cloud systems, Virtual Machines (VMs) are assigned to hosts respective to their host resource usage without consider their long term and overall utilization. As well as in many cases, the scheduling and placement processes are computationally cost expansive and affects the performance of the deployed VMs.Cloud task scheduling is one of the major problems in cloud computing, especially when deadline of the task and cost are considered. As an important actuator, virtual machines (VMs) play a essential role for cloud task scheduling. To meet task deadlines, we need to save the time of creating VMs, task waiting time, and executing time. To minimize the task execution cost, we need to schedule tasks onto their most suitable VMs for execution.To increase the task scheduling performance and reduce non reasonable task allocation in cloud, this paper proposes a a Cloud VM scheduling algorithm that takes into account already running VM resource usage over time by analyzing historical VM utilization levels in order to schedule VMs by optimizing performance by using Neive bayes technique. A job classifier motivated by Neive bayes classifier design principle is utilized to classify tasks based on historical scheduling data. A certain number of virtual machines are accordingly pre-created. This saves the time of creating virtual machine during task scheduling. Under the premise of meeting task deadlines, itreduceses the waiting time of VMs to schedule tasks, thus minimizing the cost to be paid by users who utilize VMs.*

Keywords: *Task scheduling, Load Balancing, Cloud Computing, Makespan, Cloud Scheduling*

I. INTRODUCTION

Maintaining rapid development of applications is an important aspect in the information technology sector and minimizing the time and effort spent on software deployment [1]. It is an upcoming trend widely used for the purpose of storage, memory sharing, computational capacity sharing, and sharing of hardware resources over a network like the internet. Provides resources for both individuals and organizations as a service that can be used at any time or place of the user's request and convenience [6]. This leads to time and cost saving for users because they don't necessarily need to have the resources they need and can use the service at their own wil[4]l. Cloud computing major advantages are that it addresses important and necessary aspects such as scalability, reliability, energy consumption, load balancing, time efficiency and cost efficiency. Of these tasks, resource allocation is an important task for the network to carry out[10]. This cannot be done manually when there is a large number of virtual machine in the network and is therefore done with a prefixed optimized algorithm by the machine layer[10].The cloud computing services are categorized into three ways named as Platform as a service (PaaS),Software as a service (SaaS), and Infrastructure as a service (IaaS). SaaS applications are deployed over the internet for the clients in a single instance multi-tenant model and are accessed by different devices that have internet capacity through the program interface web browser or It is one of the fastest growing services in cloud.

1.1. CLOUD MODELS:

There are three models present in cloud computing environment which are given as follows:

Public Cloud Model: The public cloud model is defined as a cloud infrastructure which is managed by an organization providing third-party service[6]. This is available as a service over the internet for both individual users and software companies/ organizations. This model's main advantage is that it is very large in scale. With limited configurations and security protection, the users in this model share the same infrastructure pool as provided by the service provider[6].

Private Cloud Model: The private cloud model is defined as a cloud computing infrastructure exclusively developed by a given company for each project or software[8]. This requires a policy of permission to host cloud applications to enforce system security and control.[8]In addition to being creating for each specific project, an external supplier or party also provide the cloud service.

Hybrid Cloud Model: The hybrid cloud model is defined as a cloud computing infrastructure that combines both public and private cloud models' advantageous factors[9]. This is done using separate algorithms used to switch between the two infrastructures.

1.1 CLOUD COMPUTING MODELS:

Infrastructure as a service (IaaS) allows users to use their storage or computational units remotely to access the given network. It does so on a demand-based basis whenever the service is required by the usere[6].

Platform as a Service (PaaS) enables users to quickly and easily create web applications with permissions to provide a substitute for the purchase and maintenance of the system's software and infrastructure. E.g: Google Engine[6].

Software as a service (SaaS) enables users to obtain an application license for any user, either as an on-demand service or through Internet subscription. In a simple way, it can be rented for use in a pay-as-you-go way instead of buying the required software. Example: Cisco WebEx, Sales force[6].

1.3. CLOUD COMPUTING TOOLS:

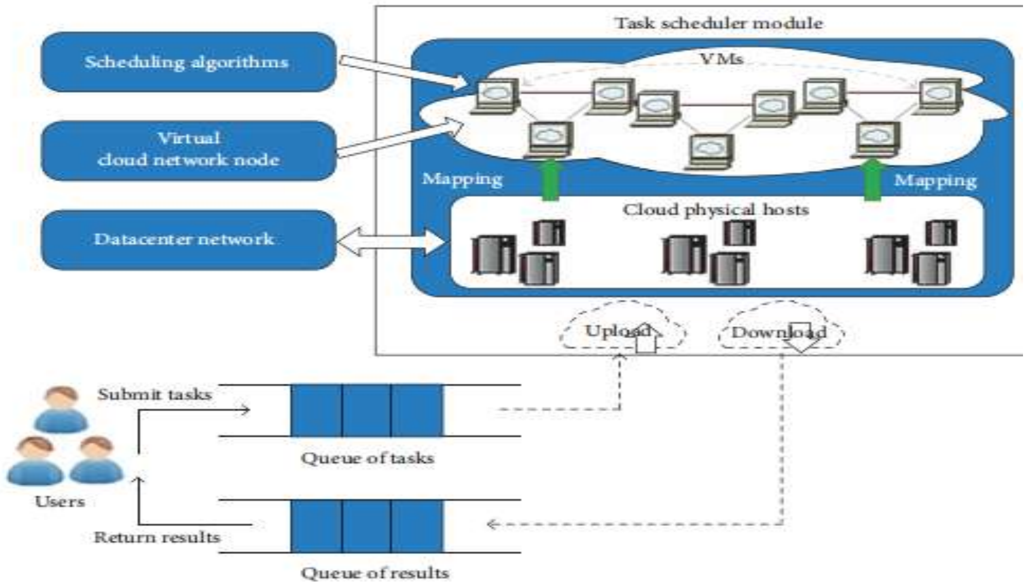
Cloud services across a network are used as efficient, organizational-based business solutions [7]. Various cloud computing tools, such as Eucalyptus, Open Nebula, Nimbus, Open stack, ete., are available where they all have various different deployment strategies[9].

Cloud computing load balancing is characterized as the process of distributing task workload and computing resources within a networked cloud environment[10]. It facilitates an organization to control applications or workload demands on a task-by-task basis, by assigning resources on the networks between the different computers or through servers[10].

1.4. TASK SCHEDULING:

This is a process that takes place while the virtual machines are using a restricted task based on the operation to be performed. [2]The scheduler collects the data from the Request Manager or Server and Resource and then calculates it to make a decision that assigns each task to their respective virtual machine.

Another important challenge that cloud computing faces is load balancing, which represents under-loaded nodes and overloaded nodes in cloud networking. Since load control and management can achieve fairness for network and better service efficient algorithms are needed for this issue [10]. In static Load balancing algorithms, the load is calculated at the compilation time and once the load is given, modifications cannot be done.[8] In dynamic load balancing algorithms, the determined load is calculated at the runtime with no previous information required. And because of huge number of requests, dynamic load balancing algorithms are suitable for many cloud computing environments because it focuses on response time minimization and throughput of the overall system, while the static load balancing algorithms focuses on minimizing the response time without focusing on overall system throughput [10].



In clouds, virtual machines (VMs) are very essential service resources for task scheduling. During scheduling, we face two types of problems. The first one is when a relatively largest task is assigned to a VM with low or weak processing ability, its processing time was relatively longer and, sometimes, the task might notbe completed before its deadline. This may be interrupting the whole task sequence. The second one is when a smaller task is assigned to a VM with strong processing ability; it mightmake a large task stay in a waiting state for the extra time[3].

II.LITERATURE SURVEY

[1] Examins the task scheduling in cloud computing environment and analyzing the program model structure of cloud computing, and proposes a hybrid scheduling algorithm based on genetic algorithm and ant colony algorithm. This algorithm makes full use of the rapid random global search ability of the genetic algorithm , It also overcomes the problem of initial pheromone lacking in an ant colony algorithm resulting in slow solution.

[2]presented the technique whose objective is to produce a new task-scheduling algorithm using simulated annealing and firefly algorithms. This new algorithm takes benefits of both firefly annealing algorithm and simulated annealing algorithms. In addition, attempt have been made to change the primary solution or primary population for the firefly algorithm. This algorithm uses a better primary solution, local search was another aspect considered for the new algorithm.

[3]proposes a technique based on a two-stage strategy to reduce the non reasonable task allocation and increase the task scheduling performance in clouds. At the first stage, a job classifier motivated by a Bayes classifier’s design principle is utilized to classify the tasks based on historical scheduling data. A definite amount of virtual machines of the various types are accordingly created. It avoids the time of creating virtual machinesat the time of task scheduling. During second stage, the tasks are assigned with various virtual machines dynamically.

[4]proposes a heuristic approach that combines the longest expected processing time pre emption (LEPT), modified analytic hierarchy process (MAHP), bandwidth aware divisible scheduling (BATS) + BAR optimization and divide-and-conquer methods to perform task scheduling and resource allocation. In this approach, each and every task is prepared before than its real allocation time to the cloud computing resources

using a MAHP approach. The combined BATS and BAR optimization method is used to allocate the resources, which considers the bandwidth and load of the cloud resources as constraints. In addition, proposed system pre-empts resource intensive tasks using LEPT pre emption. Divide and conquer technique is used to improve the performance of the proposed system.

[5] proposes the task scheduling technique based on hybrid algorithm, which merges the essential characteristics of two most commonly used biologically inspired heuristic techniques, genetic algorithms and bacteria foraging algorithm in the cloud computing. The major contribution of this study is dual fold. First to minimizes the make span and second to reduces the energy consumption, both ecological and economic perspectives.

[6] Combined two algorithms namely cuckoo search algorithm and oppositional based learningalgorithm and produces a new hybrid algorithm named oppositional cuckoo search algorithm (OCSA).The proposed algorithm shows noticeable improvement over the other task scheduling algorithms.

[7]proposed a new PEFT genetic algorithm approach to further decrease the execution time on PEFT algorithm. This strategy is developed to let genetic algorithm focuses on the optimize chromosomes objective to get best matchingmutated children. After obtaining a feasible solution, the genetic algorithm focuses on optimizee the execution time.

[8]proposes a new algorithm named Expanded Max-Min (Expa-Max-Min) algorithm to effeciently give equal importance to both cloudlets with high execution time and low execution time to be scheduled to reduce time and cost. This algorithm assigns maximum execution time cloudlet with high completion time resources as well as minimum execution time cloudlet with low computing time resources.

[9]proposed OLOA, a solution is provided for optimization, taking the make span and cost as a major constraints.This is accomplished using the two algorithms, Opposition Based Learning (OBL) algorithm and Lion optimization algorithm (LOA); and create a hybrid Oppositional Lion optimization algorithm (OLOA).

[10]proposed“MEMA Technique” with static variables techniques. In the proposed algorithm fewer steps are added to the weighted round robin (WRR). This algorithm is divided into two parts, the first one is determining the priority messages where the original balancer is divided into priority request balancer and normal request balancer. On the other hand, distribute the requests among the servers in which maximum weight obtains maximum number of request.

III. PROBLEM STATEMENT

Scheduling virtual machines for user requests in cloud computing is a NP-complete problem. This problem is usually solved by using heuristic methods in order to reduce to polynomial complexity. In this process, the cloud computing scheduler acquire the tasks from the users and assigns them to available resources taking into consideration tasks’ attributes, and requirements such as length, deadline, waiting time etc., and the resource parameters and properties. So, to address this challenge, there is a need to design priority-based task scheduling approach in cloud environment that aim to achieve an advantage performance so as to reduce the make span and execution time and gives full consideration to the characteristics of tasks. In cloud computing environment task scheduling means distributing the most suitable resources for the user task to be processed with the many parameters.

IV. PROPOSED SYSTEM

Traditionally Virtual Machines are assigned to hosts respective to their task resource usage without consider their overall and long-term utilization. Aa well as in many cases, the scheduling and placement processes are computationally cost expensive and affect performance of already deployed VMs. Thus the traditional VM placement algorithm does not consider past VM resource utilization levels.Thus the proposed work aims to increase the task scheduling performance and reduce non reasonable task allocation in cloud, this paper suggest a

Cloud VM scheduling algorithm that takes into account of already processing VM resource usage over time by analyzing past VM utilization levels in order to schedule VMs by optimizing performance by using Neive bayes technique. A job classifier motivated by neive bayes classifier design principle is utilized to classify tasks based on historical scheduling data. A definite amount of virtual machines are accordingly created. This can save the time of creating virtual machine during task scheduling.In general, the proposed work aims to prioritize the task list based on multiple criteria into dynamic queue and assign an appropriate resource to the task.

The objective is to propose the concept of VM scheduling according to resource monitoring data extracted from past resource utilizations and analyze the past VM utilization levels by using two classification technique such as Neive bayes and K-NN in order to classify tasks by optimizing performance. The algorithm enhances the VM selection phase by using real time monitoring data collections and analysis of virtual and physical resources. Our aim is to increase strength of VM scheduling .In order to incorporate criteria related to the actual VM utilization levels, so VMs can be placed by minimizing the penalization of overall performance levels. The optimization schemes involve analytics to the already deployed VMs to include

(a) Maximization of utilization levels
(b) Minimization of the performance drops.

A monitoring engine collects or gathers online resource usage monitoring data collection from VMs. The engine is capable of collecting system data based on interval and stores it to an online cloud service that makes it available for data processing. Data is collected each and every small interval of time and is stored temporarily in a local file.

Neive Bayes classifier:

The Naive Bayes Classifier algorithm is depends on Bayesian theorem and it is used specifically when the dimensionality of the input is maximum. The Bayesian Classifier has the capability of determining the most possible output respective to the input. It is also possible to add raw new data at runtime and have a better probabilistic classifier. A naive Bayes classifier completely beleives that the existence of a specific attribute of a class is misrelated to the existence of any other features when the class variablesare given. For example, a fruit might be advised to be an mango if it is yellow, round. Even though if these attributes are depends on each other or depends upon the existence of other features of a class, a naive Bayes classifier takes these properties as independent contributions to the chance that the fruit is a mango. Algorithm works as follows,

$$P\text{(label /features)} = \frac{P\text{(label)} * P\text{(features /label)}}{P\text{(features)}} \quad (1.1)$$

$$P(C/X) = \frac{P(X/C)*P©}{P(X)} \quad (1.2)$$

$$P(C/X)=P(X1/C)*P(X2/C)*\ldots\ldots P(Xn/C)*P© \quad (1.3)$$

In equation 1.3 P(c/x) is the posterior probability of the target class is given by the predictor attribute of class.

P(c) ← the prior probability of class.
P(x|c)← the probability of predictor of given class.
P(x) ← the probability of prior predictor class.

Bayes theorem gives you a way to determine the posterior probability P(c/x) from P(x), P(c)and P(x/c). Naive Bayes classifier taken into account the effect of the value of a predictor (x) on a given class (c) is free from the values of many other predictors already available in it.

Classification Framework:

Let V is a set of virtual machine, V={1,2,…..,N} & V_i , i belongs to{1,2,…..,N}, denotes the i^{th} virtual machine. V_i has four types of features indicated asv^{α},αbelongs to{1,2,3,4}.They denotes CPU resources (clock speed of CPU), network bandwidth, memory resources and hard disk storage, respectively. Hence we have $V_i = <V_i^1, V_i^2, V_i^3, V_i^4>$
Let T={1,2,…..M} is the task set given by the users & t_j , j belongs to {1,2,…..M} denotes the j^{th} task.

$t_j = <t_j^{id}, t_j^r, t_j^f, x_j>$ where

1. t_j^{id} is the unik ID of task j.

2. t_j^r represents the requirement of task j.$t_j^r = <t_j^1, t_j^2, t_j^3, t_j^4>$ specifies t_j's requirements for CPU, memory, network bandwidth and hard disk storage.

3. t_j^f indicates t_j's deadline. When t_j's deadline is not satisfied, then task scheduling fails.

4. x_j specifies the importance of the user task j.If t_j is a rush or high-paying users job, it is of high priority and x_j =1; otherwise, if it is a regular job, x_j =0.

Algorithm:

Input: Database. /*Database contais historical scheduling data*/

Output: Types of virtual machines

1. Types ← Null (Φ);

2. $\tilde{T}$ ← Processing data of Database;

3. L ← Task types count of $\tilde{T}$;

4. For i = 1 to L

5. Compute P($\tilde{T}_i$);

6. End For

7. k← TopK(P($\tilde{T}_i$));

8. For i=1 to k

9. Types← (VM types are created according to the value of P($\tilde{T}_i$));

10. v_i← create virtual machine of type i;

11. End of For

12. Output: Types;

As shown in the algorithm, we have the following steps.

a) We obtain historical data based on the historical task scheduling information, .
b) We propose a task classifier to classify tasks and store them in a database , and create a set of VM types.
c) We create a proper number of VMs of different types athosts.

The aim of this optimization schemes is to define the weight of the PM according to the resource usage of the VMs. This may reveal the information about the already deployed VMs status, like indications that the workload is running or not. To achieve this we are provided with the optimization schemes. Here classification of the VM status about its current resource usage is classified using the Neive bayes technique. Initially the virtual machine resource usage dataset is collected and monitored and then the collected data is classified using the Naive Bayes machine learning method.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a Neive bayes calssifier framework to achieve desired task classification and virtual machine creation and improve service quality of the clouds. Based on historical task scheduling information, a proper number of VMs with different resource attributes are pre created. It can save much time to create VMs and decrease the failure rate of task scheduling. According to task complexity, most suitable VMs are chosen from the pre created ones to process tasks.In future the classified tasks are matched to the virtual machines by using the euclidean distance calculated by the KNN classification algorithm and enable the concept of task migration when the tasks are matched wrongly.

REFERENCE

[1] Jun Nie "Research on Task Scheduling Strategy Based on Cloud Computing Environment" Journal of Applied Science and Engineering Innovation, Vol.5 No.1, 2018, pp. 9-12 ISSN (Print): 2331-9062 ISSN (Online): 2331-9070

[2] Fakhrosadat Fanian,Vahid Khatibi Bardsiri,Mohammad Shokouhifar "A New Task Scheduling Algorithm using Firefly and Simulated Annealing Algorithms in Cloud Computing" International Journal of Advanced Computer Science and Applications(IJACSA),Vol. 9, No. 2, 2018

[3] PeiYun Zhang "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy" IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 15, NO. 2, April- 2018.

[4] Mahendra Bhatu Gawali, Subhash K. Shinde"Task scheduling and resource allocation in cloud computing using a heuristic approach" SPRINGER: Gawali and Shinde Journal of Cloud Computing: Advances, Systems and Applications (2018) 7:4

[5] Sobhanayak Srichandan ,Turuk Ashok Kumar, Sahoo Bibhudatta "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm" Future Computing and Informatics Journal 3 (2018) 210-230.

[6] Pradeep Krishnadoss,Prem Jacob "OCSA: Task Scheduling Algorithm in Cloud Computing Environment" International Journal of Intelligent Engineering and Systems, Vol.11, No.3, 2018.

[7] Deepak K. Gupta, Raj M. Singh,and Rohit Nagar "Time Effective Workflow Scheduling using the Genetic Algorithm in Cloud Computing" I.J. Information Technology and Computer Science, 2018, 1, 68-75.

[8] James Kok Konjaang, Fahrul Hakim Ayob,Abdullah Muhammed "Cost Effective Expa-Max-Min Scientific Workflow Allocation and Load Balancing Strategy in Cloud Computing" Journal of Computer Science 2018, 14 (5): 623.638.

[9] Pradeep Krishnadoss,Prem Jacob "OLOA: Based Task Scheduling in Heterogeneous Clouds" International Journal of Intelligent Engineering and Systems, Vol.12, No.1, 2019.

[10] Saher Manaseer, Metib Alzghoul, Mazen Mohmad "An Advanced Algorithm for Load Balancing in Cloud Computing using MEMA Technique" IJITEE ISSN: 2278-3075, Volume-8 Issue-3, January 2019.