An Efficient Approach For Load Balancing And Scheduling In Cloud Computing

Nivedha[1], Nithya[2], Priyanka[3]
[1]UG Scholar, Department Of Computer Science And Engineering, Velalar College Of Engineering And Technology, Erode, Tamil Nadu
[2]Assistant Professor, Department Of Computer Science And Engineering, Velalar College Of Engineering And Technology, Erode, Tamil Nadu
[3]UG Scholar, Department Of Computer Science And Engineering, Velalar College Of Engineering And Technology, Erode, Tamil Nadu
[1]nivedhaaruchamy@gmail.com, [2]nithi1710@gmail.com, [3]sweetpriyankakm@gmail.com

Abstract: In the cloud environment, the tasks are assigned to the virtual machines based on their requirements without considering their long-term and overall utilization. Also, in many cases, load balancing and scheduling process are computationally cost expensive and also affects the performance of the Virtual Machines (VMs). Task scheduling is a major issue in cloud computing. An efficient load balancing and scheduling algorithm should focus on satisfying all the Quality of Service (QoS) parameters such as task execution time, task waiting time, resource utilization, execution cost, etc. In order to improve the performance of task scheduling and also to meet the above mentioned QoS parameters, a VM scheduling algorithm that takes already running VM resource usage over time by analyzing past VM utilization in order to schedule the tasks among the VMs by optimizing performance by using K-Nearest Neighbor (K-NN) and Naive Bayes (NB) classification technique is proposed. Thus, by using the historical VM utilization, the task waiting time and the execution time have reduced, and also the resource utilization has increased. The count of the physical machine gets reduced by four using K-NN and NB classifier. The task performed by 28 physical machines when using Support Vector Machine (SVM), JAYA is reduced by using K-NN and NB classifier algorithm, and also the error got decreased by 0.025%.

Keywords: Scheduling, Execution time, Machine learning, Optimization, Resource utilization.

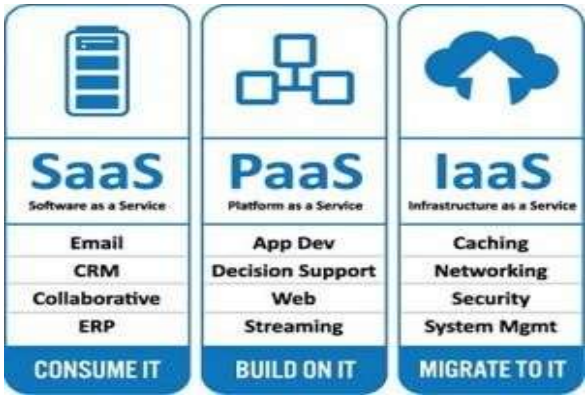
I. INTRODUCTION:

The cloud computing technology uses central remote servers and internet in order to maintain the data and applications. This technology allows the users to use an application without installation and to access their personal data from remote servers only through the internet. Today, the technological world is slowly moving towards developing applications using cloud services by incorporating features such as elasticity, low cost, scalability, virtualization, pay per usage etc. Now-a-days cloud provides various services to users which ease them from several problems such as cost for maintaining data and servers, ensuring availability of data in case of any natural calamities etc. Thus the users no need to stress on these problems because once we move on to cloud services, it's the duty of cloud service provider to maintain the data and applications, no matter what the situation.

The various services provided by cloud are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Therefore, cloud requires scalability, robustness, availability, reliability and high performance features. Cloud has large amount of resources such as Disk Storage, CPU, Bw, RAM etc. But still, this paradigm has a various open key challenges to be resolved such as dynamic scheduling, autonomic load balancing, energy efficiency, fault tolerance, resource allocation and big data analytics etc. Of all autonomic load balancing, resource allocation and fault tolerance are the major issues in cloud computing which has been addressed in this paper. Resource allocation cannot be done manually when there is a large number of VMs in the cloud network. Therefore it is done with a prefixed optimized algorithm in the machine layer, thus balancing the load in the cloud.

1.1 SERVICE MODELS:

Once a cloud is established, how its cloud computing services are deployed in terms of business models can differ depending on requirements. The service models being deployed in cloud environment are commonly known as



Software as a Servive (SaaS) enables the users to obtain an application licence for any user, either as an internet subscription or through on-demand service. In a simple way, it can be rented as pay-per-use way instead the buying the entire software required.

Platform as a Service (PaaS) enables users to quickly and easily create web applications by purchasing access to platforms and maintaining the system's software and infrastructure. Example: Google Engine.

Infrastructure as a Service (IaaS) allows user to use their storage or computational units remotely to access the network. It does it on a demand-based basis whenever the user needs the service.

1.2 TASK SCHEDULING:

The task scheduling is the process that takes place while the virtual machines are using a restricted task based on the operation need to be performed. The scheduler collects the task from the Request Manager and then assigns the task to the respective virtual machine.

The task scheduling is the process that takes place while the virtual machines are using a restricted task based on the operation need to be performed. The scheduler collects the task from the Request Manager and then assigns the task to the respective virtual machine. The main aim of cloud computing is to use the available resources optimally. For the optimization process, the scheduling algorithm play an important role in scheduling the user tasks among the virtual machines. The scheduling algorithms usually have the goal of spreading the load

on available VMs and maximizing the resource utilization while minimizing the total execution time and the waiting time. During scheduling among the virtual machines, 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 capacity. The second one is when a smaller task is assigned to a VM with strong processing capacity. It might take a large task to wait for a long time. This paper presents an optimization algorithm for task scheduling to achieve minimization of overall computation time and maximizing the performance.

II . LITERATURE SURVEY:

[1] examines 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 use of 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. The another aspect considered for the new algorithm was local search.

[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, the Bayes classifier 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 machines at the time of task scheduling. During second stage, the tasks are assigned with various virtual machines dynamically.

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

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

[6] 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 matching mutated children. After receiving a feasible solution, the genetic algorithm focuses on optimizing the execution time.

[7] proposed OLOA, a solution is provided for optimization, which takes 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), thus creating a hybrid Oppositional Lion optimization algorithm (OLOA).

[8] proposed an algorithm named Binary JAYA which mainly focuses on reducing the execution time and improving the resource utilization by assigning the tasks among the VMs dynamically.

III . PROPOSED SYSTEM:

The proposed system aims to increase the performance of scheduling the task and reducing non reasonable allocation in cloud. It also aims to maximize the resource utilization and reduce the task execution time and waiting time. The 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 Knn and Naive bayes technique. These optimizing algorithms classify the tasks among the VMs based on the historical 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 schedule the VM according to the data extracted from the past resource utilization and analyzing it by using two classifier algorithms such as K-NN and Naive Bayes in order to classify the task by optimizing performance. In KNN classification, the classified task are matched to the VMs by calculating the Euclidean distance. These algorithms 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, VMs can be placed by minimizing the penalization of overall performance levels. The optimization schemes involve analytics to the already deployed VMs 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.

K-NEAREST NEIGHBOR CLASSIFIER:

K-nearest neighbor is an algorithm that stores all available cases and classifies new cases based on the similarity measure (e.g., distance

functions). K-NN has been used in pattern recognition and statistical estimation. The k-nearest neighbor algorithm is a non-parametric method used for regression and classifications.

$$\text{Euclidean distance}=\sqrt{\sum_{i=1}^{k}((x_1-x_2)^2+((y_1-y_2)^2+((z_1-z_2)^2)}$$

NAIVE BAYES CLASSIFIER:

The Naive bayes algorithm depends on Bayesian theorem and this algorithm is used when the dimensionality of input is maximum. This Bayesian classifier has the ability to determine the most possible output to the respective input. It is also possible to add raw new data at runtime and have a better result. This algorithm completely believes that the existence of specific attribute of the class is misrelated to the existence of any other features when the class variables are given. For example a fruit might be advise to be an apple if it is red, round. Eventhough these attributes depends on each other or depends on the existence of other features in a class, the naive bayes takes these properties as independent contributions to the chance that the fruit is apple. Algorithm works as follows:

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

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

$$P(C/X) = P(X_1/C)* P(X_2/C)* \ldots\ldots P(X_n/C)*P(C) \quad (1.3)$$

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

P(c) is called the prior probability of class.

P(x/c) is the probability of predictor of given class.

P(x) is the prior probability of the predictor of the class.

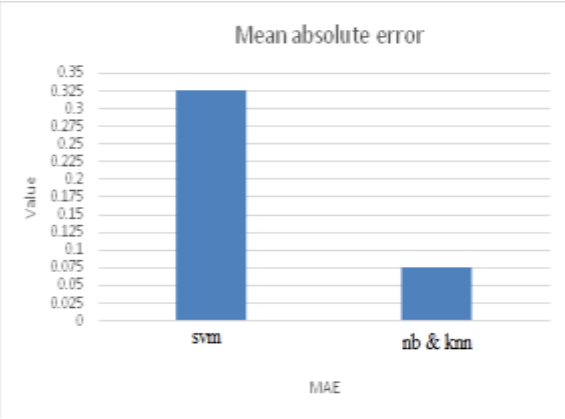
Bayes theorem gives a way of calculating the posterior probability, P(c/x), from P(c), P(x), and P(x/c). The NB classifier considers that the effect of (x) on a given class (c) is independent of the values of other predictors.

ALGORITHM:

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

Output: Types of virtual machines

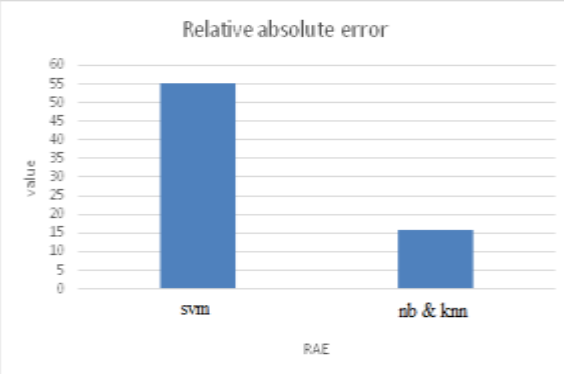
1. Types ← Null (Φ);
2. $\tilde{T}$ ← Processing data of Database;
3. L ← Task types count of $\tilde{T}$;
4. For i = 1 to L
5. Compute P($\tilde{T}_I$);
6. End For



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;

The above algorithm contains following steps:

a) Obtaining the historical scheduling data

b) Proposing a classifier to classify the tasks and storing them in a database, and creating a set of VM types.

c) Creating a proper number of VMs of different types at hosts.

After this the Euclidean distance is calculated for the classified tasks to match the tasks to the VMs.

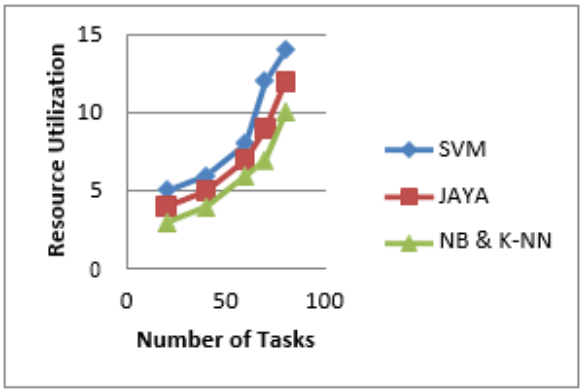
VM SCHEDULING:

The main aim of these optimization schemes is to define the weight of the PM according to the resource usage of the VMs. This will reveal the information about already deployed VMs status, like indications whether workload is running or not. To achieve this we provide two optimization schemes. Here classification of the VM status about its current resource usage is classified using the knn and NB. Initially the virtual machine resource usage dataset is collected and monitored and then the collected data is classified using the machine learning methods like Knn and NB.

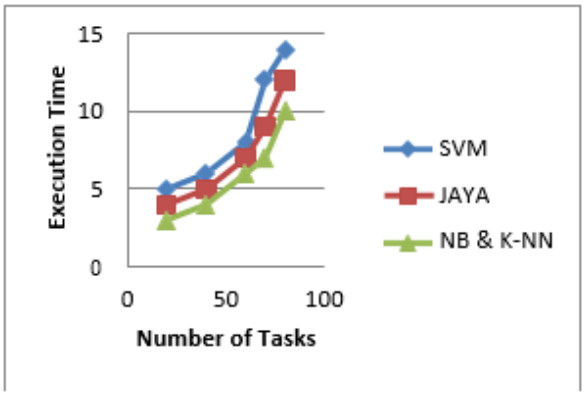
IV. RESULT ANALYSIS:

Mean Absolute Error

Relative Absolute Error



Number of tasks vs Resource utilization (%)



Number of tasks vs Execution time (s)

The algorithm evaluates past resource utilization levels and classifies according to the overall resource usage. At the end the list of candidate hosts is populated and the resources are ranked accordingly. In detail, by using this algorithm PMs are re-ranked according to the selected optimization scheme and based on their VM usage. For example we use as data set resource information from 24 hours monitoring and as training set a seven day resource usage monitoring. The results thus obtained at the end of the process have been shown above in the figure.

V. CONCLUSION:

In this paper, we proposed Naive bayes and K-NN classifier to achieve the desired task classification and creation of virtual machine and improve the QoS such as execution time, waiting time, resource utilization and execution cost of cloud computing. Based on historical scheduling information, certain 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. Therefore, most suitable VMs are chosen from the pre created ones to process tasks. Thus the task performed by 28 physical machine when using SVM and Binary JAYA is reduced by 24 physical machine by using K-NN &NB classifier algorithm also the error rates gets reduced by 0.025%. The future research work may be carried out by considering other QoS parameters to improve the performance.

REFERENCES:

[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] 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.

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

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

68-75.

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

[8] Kaushik Mishra, Jharashree Pati, Santosh Kumar Majhi "A dynamic load scheduling in Iaas Cloud using Binary JAYA Algorithm" Journal of King Saud University-Computer and Information Sciences (2020).

[9] Deepika M, Prabhu S "Cloud Task Scheduling Based on a Two Stage Strategy using KNN Classifier" International Journal of Latest Engineering Science, Vol.02, Issue.6, 2019