2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. Sep 3-4, 2021

2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) | 978-1-6654-1703-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRITO51393.2021.9596289

Load Balancing in Cloud Computing Using Support Vector Machine and Optimized Dynamic Task Scheduling

D.Radhika
Assistant Professor
Department of computer Science and Engineering
Vivekanandha College of Engineering for women
Tirunchengodu, Namakkal
radhikadeva6@gmail.com

Dr.M.Duraipandian
Professor
Department of computer Science and Engineering
Vivekanandha College of Engineering for women
Tirunchengodu, Namakkal
durainithi@gmail.com

***Abstract*: Cloud computing enables the internet based computing to everywhere. In cloud computing the task scheduling and load balancing is the important issue that must be considered in data centers. In cloud platform service providers allocate the resources based on the available resources this will increase the QoS. The process of distributing workloads is a load balancing Technique. In this paper focusing the load balancing between the virtual machine. Load balancing are different types that is network load, memory capacity, storage capacity and CPU load. Load balancing supports failover, enabling scalability and decreases costs and maximizes the resource availability. This paper presents the detailed review of load balancing with machine learning technique such as naïve bayes, random forest, Decision Tree and Support Vector Machine. By these machine learning model, SVM algorithm significantly to classify the incoming request to reduce the processing time (makespan). Also provide the moth flame optimization technique for dynamic task scheduling to make better scheduling under different load condition. We have experimented and evaluated this algorithm in a cloud simulation environment; we obtain better outcome than some other well-known algorithms. The Simulation results demonstrate that the proposed technique has a better development in terms of reducing the makespan time and load balancing.**

Keywords: load balancing, Cloud computing, processing time, makespan in cloud and Machine learning classifier.

I. INTRODUCTION

Big data is a wide range of data collected, processed, structured and analyzed. Big Data is information, while cloud computing is an infrastructure, which combines both stable and efficient technology. Often big data is structured or unstructured. These information is also so comprehensive that conventional database and software methods cannot be used for processing. In cloud computing it is the shortest possible time, instead of the local computer disk drive, to save and access data, files, and programs over the internet. The cloud can serve as a network metaphor.

Cloud Storage offers the ability to store substantial amounts of information with the aid of scalability, faults and accessibility through hardware virtualization. The virtualization of hardware makes big data accessible in a cloud computing environment that can be scalable and defect-tolerant. In our modern world, big data and cloud computing act an enormous role. Both of these links allow people with excessive ideas, but partial resources are a chance for business success. The task and load balance for servers is needed in limited resources. Cloud computing tasks provide the best way for computing resources to be allocated within the data center according to customer request[1]. Tasks are assigned in schedule to different virtual machine types with regard to the user and service provider restrictions. While a number of computer environments are programmed with multiple scheduling methods, they should be adapted to a cloud situation if they are not explicitly configured for it. For instance, there are a few cluster independent and homogeneous computers that work together as a single device, which use local resources, and the cloud surroundings contain a collection of heterogeneous resources that are distributed and users can use all of them; therefore a scheduling algorithm that can be good for a cluster might not be suitable for a cloud environment [2]. The problem space should be transformed into the elements of the algorithm so that the algorithm can be implemented in the cloud. The growth of heterogeneous VMs and the variation of task sizes lead to a large variety of tasks [3-4].

Users are allowed to access the servers, memory, network, etc., Depends on the user needs, resource requirements can be increased or reduced. VMware and Amazon EC2 are some of the familiar examples for IaaS. Software as a service provides an application to the user as a service so that the user can access the application [5-7]. Generally, the applications are hosted by cloud datacenters so that it reduces the maintenance and infrastructure cost to the user. At the same time, the user cannot modify the application and all the necessary changes will be performed by the service provider only. PaaS provides all the essential resources to a user to develop an application. The services are accessed through the internet and user don't need to install any specific application to access the resources. Application design and development, testing, deployment, and hosting can be done in PaaS [8]. Compared to other services, the cost of PaaS is quite high but it provides more benefits to the community developers.

The finding of the appropriate permutation for all permutations with the least completion time is an NP-hard problem. Many similar studies were performed in conjunction with cloud planning using meta-heuristic algorithms; however, one of the key problems in task planning was the fair resource allocation applications that are

Authorized licensed use limited to: Seoul National University. Downloaded on November 09,2022 at 19:39:54 UTC from IEEE Xplore. Restrictions apply.

compliant with application requirements. In order to balance the load, machine output can be reduced by using the scheduling algorithm.

II. Literature Review

A. M. Nakai et al., [9] have developed Load balancing protocol for global web servers with a new server. It uses a middleware to enforce a procedure which forwards requests without overloading them to the nearest remote servers. This helps to reduce the response times. A heuristic also helps to support the web server overload.

The Join Idle Queue has provided a load balancing algorithm to Lua et al. [10] for web services that could dynamically be expanded. This algorithm offers a huge load balancing with distributors. The role of load balance is removed from the crucial processing path to reduce device load effectively. In addition, there is no contact burden when employees arrive, nor does the actual response time increase.

X. Liu et al., [11] have proposed a 'lock-free load balancing approach' to discourage use of common memory relative to other load balancing criteria for multiprocessing. The reformation of the Linux kernel was accomplished. By executing different load balance channels in a load balancer, the overall load balancer output is improved in a multicore setting..

J. Hu et al. [12] have suggested a load balancing of VM resources using genetic algorithm based scheduling. This approach uses the historical and current state of the scheme, to compute required VM resource allocation on the system. By selecting the least-affective solution, it supports in analyzing the problem of load-imbalance and expenses of migration, thereby attaining reasonable resource utilization.

SmaranikaParida et al. [13] proposed a machine learning technique for load balancing that results in efficient load balancing in cloud infrastructure. Load balancing is classified into two types: static load balancing, which is appropriate for homogeneous environments with low load variations, and dynamic load balancing, which is more versatile and appropriate for heterogeneous environments.

Zhang J et al. [14] present a series of research studies on how to speed up the training of a distributed machine learning (ML) model using cloud services. Today's ML models are trained using distributed ML, which has become the industry standard. Also proposed is an adaptive quick reassignment load balancing technique (AdaptFR). Based on this, we created the adaptive-dynamic synchronous parallel distributed parallel computing model (A-DSP). In the distributed ML model training, the model can effectively increase training speed while maintaining model accuracy.

Wang JB et al. [15] suggested a machine learning system for resource allocation with cloud computing assistance. An example of beam allocation in multi-user massive multiple-input multiple-output (MIMO) systems demonstrates that the proposed machine-learning-based resource allocation approach outperforms traditional methods.

III. Proposed Scheme

A dynamic algorithm of load balancing in the cloud setting has been proposed in this paper. Using different machine learning algorithm to put the request on the machines for reducing the makespan. For implementing the proposed scheme is conducted by CloudSim toolkit has been used for modelling in a cloud environment.

1. Provide an adequate way to plan applications using SVM technology to minimize the time and use of resources;

2. The use of machine learning classification methods for equitable distribution of requests;

3. Targeted research to demonstrate the efficacy of the algorithm as compared with previous algorithms`;

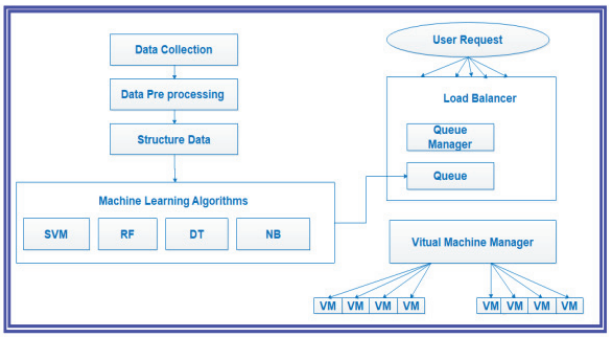


Fig. 1. Architecture of proposed scheme

In figure 1 shows that the proposed architecture flow diagram, which represent that initially the collected data's are preprocessed to remove the unrelated data to give the structured data, then applying the machine leaning model to classify the data as (Type 1, Type 2, Type 3 and Type 4). In which four types of data's have different load condition. In dynamic scheduling algorithm of machine learning model is to classify the data under different load condition to give the load balancer. The projected dynamic task scheduling systems and load distribution are carried out at any time based on the most recent details from previous and current server status. Virtual Machine Manager completely manages the operations of a virtual environment through a centralized interface that provides the status, availability, and performance of the VMs. And also provide an optimized approach of moth flame optimization algorithm to solve the task scheduling problem in cloud environment focusing to minimizing the makespan and increase the load balancing.

IV. LOAD BALANCING

Load balance is a method of allocation of workloads in distributed systems such as cloud computing, which ensures that no computer is underloaded or idle [16 and 17]. Load balancing attempts to speed up various parameters such as response time, runtime, device stability etc. [18 and 19].

Load balancing can be classified as the Static Load Balancing (SLB) and the Dynamic Load Balancing (DLB) techniques. In SLB, the allocation of the tasks to processors is done deterministically or probabilistically at the compile time using a previous information about the tasks and the system, whereas in DLB, load distribution decision is made at the run-time by making use of the current processes and the system state information. SLB can further be classified into Optimal SLB & Sub-Optimal SLB and DLB can further be classified into two categories namely Distributed and Non-distributed

Authorized licensed use limited to: Seoul National University. Downloaded on November 09,2022 at 19:39:54 UTC from IEEE Xplore. Restrictions apply.

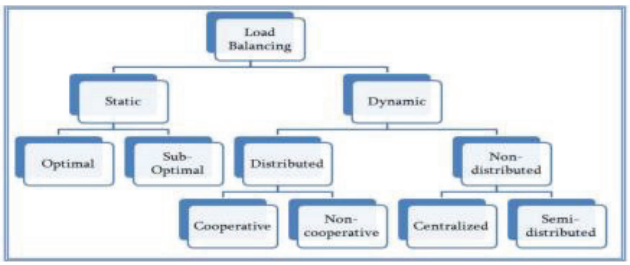


Fig. 2. Classification of Load balancing

The researchers' planned numerous load balancing approaches concentrate more on task planning, task allocation, planning of resources needs, allocation of resources and resource management. We could not find an in-depth and complete literature dealing with factors that create an unbalanced situation for load to the best of our knowledge. A systemic classification of the methods and techniques could not be provided by load balance survey reports. The main aim of this work is to evaluate current work along with its benefits and setbacks. A distinction is also made between various current load balancing scheme and thus the cloud load balancing challenges.

Activities involved in load balancing

- User task requirements Identification
- VM Resource Identification Information
- Task scheduling
- Resource allocation
- Identification of user task requirements
- Migration

User task requirements Identification

The Resources are Identified for the incoming task for execution on VM

VM Resource Identification Information

Details of a VM resource state, current use of resources of VM and unallocated resources are provided in this process. On this basis the state of VM is balanced, overloaded or under loaded.

Task scheduling

Task scheduling is important in cloud computing, and many scheduling policies are used by the master node for computing nodes to effectively distribute their tasks in many critical applications. One of the most critical issues of cloud computing is resource scheduling and utilization. Using a scheduling algorithm, activities are scheduled to boost cloud computing services in VMs.

Resource allocation

The personnel are assigned to planned execution activities. In order to do this, a resource allocation strategy is used. Although the schedule is necessary to speed up implementation, allocation policy is used to properly manage resources and to improve the efficiency of resources.

Migration

Migration is a critical stage in cloud load balancing and latter is incomplete without cloud load balancing. The migration is of two cloud types depending on the organization that takes VM migration and task migration into account.

Cloud load balancing uses the scalability and versatility of the cloud to meet new demands for work loads and improve overall availability. In addition, cloud load balancing technology is extremely effective in providing health checks for cloud applications when it comes to load and traffic distribution.

Scheduling algorithm

Conventional scheduling algorithms assign tasks without checking the workloads this leads to heavy utilization of few VMs and underutilization of other VMs. This issue is rectified in the proposed approach through its classification and optimization approach. A mathematical model for the proposed dynamic scheduling algorithm is formulated in this section for VM classification and optimization.

The proposed moth flame optimization dynamic task scheduling scheme to initialize pheromone, followed by the task of selecting VM based on that value. Generally, moths fly towards the moon at night in a straight line. In between if any light source interferes it considers that as the moon and started to rotate around the light at a specific angle. Moths are considered as search agents and the light source is considered as the best position. This best position is considered as the optimal position for other moths in the group. We keep two matrices: the Completion Time Matrix and the Pheromone Matrix. The completion time of all tasks can be calculated based on the start time of each task. The first task's start time is chosen at random. The start time of other tasks can be determined by subtracting the completion time of the tasks previously assigned to the respective computer. Finally, the schedule's makespan is the period when all tasks are finished, i.e. the overall completion time. The moth flame optimization technique determines the best mapping of tasks to VMs. When each mission is allocated to every available resource, an ant produces a fully feasible solution. Each assignment is assigned to a particular resource, and no pre-emption is permitted. The algorithm begins by priming the ants with pheromones. The proposed algorithm would provide more refined empirical data on which to base the existing load scheduling process.

A schedule is created by the Scheduling algorithm for a given problem. Some elements that affect the operation of a programming algorithm are:

- Finishing Time: The time when the last task is completed is referred to as the schedule's finishing time.
- Throughput: Amount of processes executed at particular period of time.
- Processor utilization: It refers to the amount of work handled by a processor or the use of processing resources.

V. MACHINE LEARNING ALGORITHMS FOR CLASSIFICATION

In this section, we proposed the machine learning methods to classify the tasks at different load condition, by using the different classifier to analysis the performance of the VM Classification.

Authorized licensed use limited to: Seoul National University. Downloaded on November 09,2022 at 19:39:54 UTC from IEEE Xplore. Restrictions apply.

A. Decision tree

DT is the Supervised Machine Learning procedure. It perform classification process without need of much computation, also able to handle both categorical and continuous variable. It splits the data set into and minor subsets, while expanding the prospects for the solution. The end result is a tree with decision nodes and end nodes. Using the decision algorithm, we start with the path tree and divide the object data by the major information gain (IG). In an iterative process, this division process can be repeated on each child node until the leaves become clear. This resource that the samples in every leaf node are all in the equal class. We can set anedge on the depth of the tree to prevent distillation. Here we compromise purity because the last leaves may still be impure.

B. Random forest (RF)

One of the most successful classification methods is now known as random forest (RF). However, given the complexity of data distribution in future large areas, a random forest could contain poor tree classifiers that can lead to false classification results. If a large number of bad trees are found in the random forest, the vote of all trees for a choice of classification would take the wrong step. Bad trees should be identified and removed for optimization in the random forest in order to reduce their negative influence on the random forest results.

C. Naïve Bayesian

The Bayesian Naive is a classification statistical tool. The research shows that this approach has similar performance, in contrast, with higher accuracy and speed than other methods in comparison to other methods, such as the decision tree of classification or neural network classifiers chosen. Suppose there are m classes numbered$\{Y1, Y2, \ldots, Ym\}$ and tuple-X as data. The Classifier predicts that X belongs to the class with the greatest retrograde likelihood, i.e. X to the class Yi. Assume the virtual machines used to $host\ user\ petitions = \{VM1, VM2 \ldots VMm\}$.So, it's a good thing. $Task = \{T1, T2 \ldots Tn\}$ is a group of tasks to be performed on virtual machines.

D. Support Vector Machine

SVM is a monitored algorithm for machine learning used for several classifications and regression issues. It continues to be one of the most widely used robust methods for classifications in many use cases.Designing low-complexity multiclass classifiers necessitates the use of more sophisticated techniques. One promising strategy is to convert the multiclass classification problem into a series of binary classification problems that can be solved efficiently with binary classifiers. So far, the SVM has been regarded as one of the most robust and efficient algorithms for designing low-complexity binary classifiers that evaluate the class of a new feature vector using linear boundaries (hyperplanes) in high dimensional spaces. More precisely, only a few hyperplanes separate the two groups. As a result, the class is defined by which sides of hyperplanes belongs to the new function vector.

In order to precisely distinguish two or more different classes in a classification problem, SVM finds an optimal line called 'hyperplane.' The objective is the optimum separation of hyperplanes by training the linear and nonlinear data with the SVM algorithm.

The vectors of support, however, are the points between the two or more groups that are the nearest to the hyperplane. These are the most difficult to identify data points. As a rule, it is easier to define the algorithm correctly when the gap or distance is greater between the support vectours. Therefore, once the hyperplane is configured, the optimal segregation or the maximum range classification.

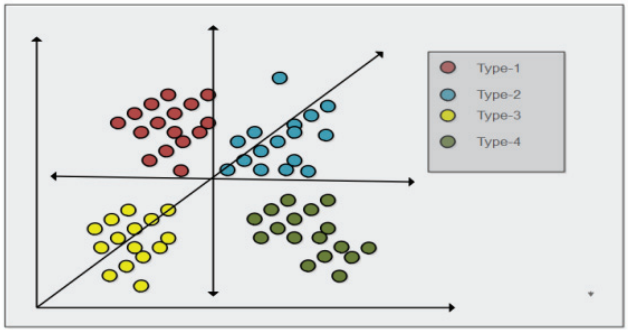


Fig. 3. SVMMulticlass Classification

In figure 3 shows that the classification of four class two factor such as linear and nonlinear factors. To separate the four classes of data points, there are many possible hyperplanes that could be chosen byStable Virtual Machines and unstable Virtual Machines. To find a plane that has the maximum margin, that is the maximum distance between data points of four classes as Type 1 Type 2 Type 3 and Type 4. Maximizing the margin distance provides some reinforcement so that future data points can be Virtual Machines classified with more confidence.

SVMmulticlass uses the multi-class formulation, but optimizes it with an algorithm that is rapid in the linear case. For a training set (x1, y1) and (x2, y2) ... (xn,yn) with labels yi in [1.. k], it discovers the solution of the subsequent optimization problem during the process of training task.

The linearfactor is related to the average utilization of VM and the nonlinear factor is related tounderutilization or high utilization of VMs in the proposed concept.The process of redistributing VMs from overloaded nodes to loaded nodes is the VM load balance.Here SVM algorithm uses the linear and non- linear classification techniques. The linear factor use the average utilization of VM and non- linear factor is uses the overloaded nodes or under loaded nodes in VM.SVM uses the boundaries between the factors as classification results.

The VM classification categorized into four types

1. **Type 1:** Unstable Virtual Machines-high resource utilization

2. **Type 2:** Unstable Virtual Machines- Moderate resource utilization

3. **Type 3**: Stable Virtual Machines- high resource utilization

4. **Type 4:** Stable Virtual Machines- Less resource utilization

Authorized licensed use limited to: Seoul National University. Downloaded on November 09,2022 at 19:39:54 UTC from IEEE Xplore. Restrictions apply.

Algorithm

VM Classification based on Virtual Machines − resource utilization

Input: *example_file model_file (Virtual Machines − resource utilization* $[x1, y1]$ *and* $[x2, y2]$)

$x1 =$ *Unstable Virtual Machines − high resource utilization*

$y1 =$ *Unstable Virtual Machines − Moderate resource utilization*

$x2 =$ *Stable Virtual Machines − high resource utilization*

$y2 =$ *Stable Virtual Machines − Less resource utilization*

Output: *VM machine classification data type*

{

Begin

wheights $qi, i = 1..l$

bias b

trainingset partition into suppotset(s) , *errorset*(e) *and remainingset*(r)

params: $e, c,$ *kerneltype and kernelparam*

r matrix

sample to remove index (c)

}

{

for all x *in* $[1..k]$: $[\, x1 \bullet y1 \,] >$

$= [\, xn \bullet wy \,] + 100$

for all y *in* $[1..k]$: $[\, x2 \bullet y2 \,] >$

$= [\, xn \bullet wy \,] + 100$

$\Delta(xn, x)$ *is the loss function that returns* 0 *if* xn *equals* x, *and* 1 *otherwise*

{

$\Delta(yn, y)$ *is the loss function that returns* 0 *if* yn *equals* y, *and* 1 *otherwise*

}

}

Update xn and yn

Update classification solution

End

End

When a new request arrives, the load balancer will assign it to the appropriate VM after consulting with the queue manager to maintain balance. The proposed approach employs a machine learning methodology for effective load balancing. In the first stage, various data about the VM will be collected, such as the VM id, power, current load, CPU utilization, I/O, Network In, Network Out, memory utilization, and so on. The data is then organized as required in the following stage, and we must use machine learning to find out information about the VM and store it in a queue. The previous queue is then replaced by this queue.

VI. Simulation Result

Our studies in this document use the ML frame that has been distributed. We used a frame for simulating a cloud environment and the cluster operating system on a cluster of three nodes. Currently every node has two AMD Opteron processors with a main frequency of 2.4 GHz Processor. Per processor has 8 nuclei and 32 GB of memory. The model's success is calculated in terms of classification accuracy and the model training takes time.

The ultimate goal is to enhance system efficiency in a distributed load balancing algorithm. A load balance algorithm should also implement an efficiency index to calculate this increase in performance. As there are more than one index to be used, the range normally varies between algorithms. We performed SVM with the distributed system on a parallel computer. We know well that, the efficiency of the SVM used to interconnect the machines in a distributed system, the distributed system has a greater execution time than the parallel system with the same number of processors. SVM typically has a much greater latency than parallel interconnections. Our SVM tries to reduce this overhead to improve the efficiency of distributed systems.Fig 5 shows VM classification based on resource utilization.

Resource utilization: Resources are costly and must be used optimally. This criteria is used to check the percentage time for which the resource was used. The value should be as high as possible with maximum of 100.

Average Resource Utilization

$$\sum_{i=1}^{n} \frac{\text{Time taken by resource i to finish all jobs}}{Makespan\ X\ n}$$

where$n$ is denote as the number of resources

Makespan: It is the most important optimization criteria that designates the finish time of the last task on the data centre. Makespan indicates the concluding time of the last task. It comes under the category of minimization function.

$$Makespan = maxi\epsilon tasks\,\{Fi\},$$

Where,Fi denotes the finishing time if task i.

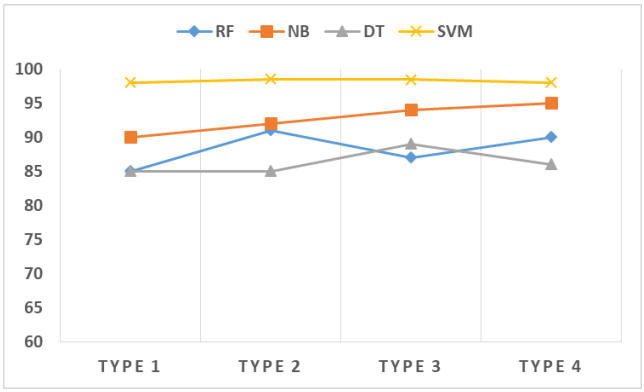


Fig. 4. VM Classification using different machine learning classifier

In figure 4 represent that the performance analysis VM Classification of different classifier model with proposed SVM algorithm, in this comparision SVM attained the better classification result under different load condition

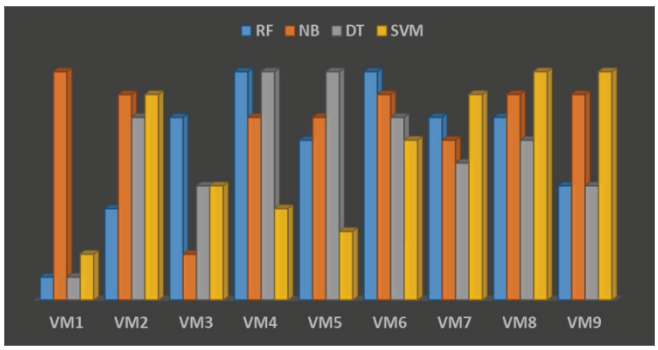


Fig. 5. VM Utilization in different machine learning model.

In figure 5 represent that the performance analysis VM Utilization of different classifier model with proposed SVM

Authorized licensed use limited to: Seoul National University. Downloaded on November 09,2022 at 19:39:54 UTC from IEEE Xplore. Restrictions apply.

algorithm, in this comparision SVM attained the better VM Utilization result under different VM.

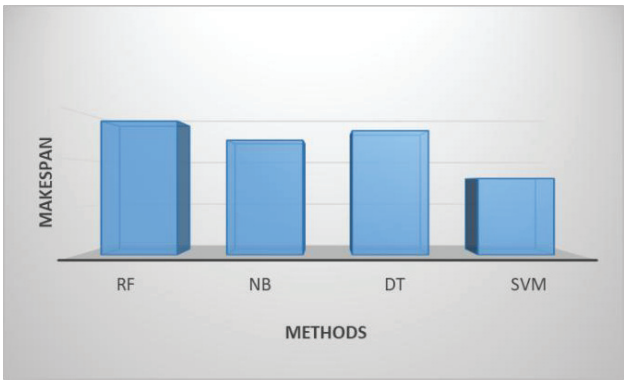


Fig. 6. The makespan values on different method

In figure 6 represent that the performance analysis of makespan time of different classifier model with proposed SVM algorithm, in this comparision SVM attained less make span time result than other machine learning algorithm.

VII. CONCLUSION

Cloud computing has been facing many challenges such as providing protection, an effective load balancing strategy, scaling, proper resource planning, data center energy usage, performance monitoring. Load balancing algorithms focus entirely on what workload situations are allocated during compile time or runtime. Our architecture, as a whole, may provide resources for various computer-intensive applications running on a heterogeneous cluster as a generic model.In this study, we compared the different machine learning model, by this comparison analysis SVM attained the better performance than another machine learning algorithm.In addition, by specifying application specific kernels, it is possible to increase the exactness of the SVM model. Also provide the moth flame optimization technique for dynamic task scheduling. The model will assist in predicting the correct node for a new mission. In general, our model will serve as a model to provide tools for a variety of computer-intensive applications.

REFERENCES

[1] .K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," in Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on, pp. 137-142, 2012.

[2] Mishra AK, Umrao BK, Yadav DK. A survey on optimal utilization of preemptible VM instances in cloud computing. The Journal of Supercomputing. 2018 Nov;74(11):5980-6032.

[3] Xu M, Tian W, Buyya R. A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience. 2017 Jun 25;29(12):e4123.

[4] Pal AS, Pattnaik BP. Classification of virtualization environment for cloud computing. Indian Journal of Science and Technology. 2013 Jan;6(1):3965-71.

[5] Mathew T, Sekaran KC, Jose J. Study and analysis of various task scheduling algorithms in the cloud computing environment. In2014 International conference on advances in computing, communications and informatics (ICACCI) 2014 Sep 24 (pp. 658-664). IEEE.

[6] Cao Q, Wei ZB, Gong WM. An optimized algorithm for task scheduling based on activity based costing in cloud computing. In2009 3rd international conference on bioinformatics and biomedical engineering 2009 Jun 11 (pp. 1-3). IEEE.

[7] Jena UK, Das PK, Kabat MR. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University-Computer and Information Sciences. 2020 Feb 1.

[8] Shafiq DA, Jhanjhi NZ, Abdullah A, Alzain MA. A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications. IEEE Access. 2021 Mar 10; 9:41731-44.

[9] A. M. Nakai, E. Madeira, and L. E. Buzato, "Load balancing for internet distributed services using limited redirection rates," in Dependable Computing (LADC), 2011 5th Latin-American Symposium on, pp. 156–165, IEEE, 2011.

[10] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, "Join-idlequeue: A novel load balancing algorithm for dynamically scalable web services," Performance Evaluation, Elsevier, vol. 68, no. 11, pp. 1056–1071, 2011.

[11] X. Liu, L. Pan, C.-J. Wang, and J.-Y. Xie, "A lock-free solution for load balancing in multi-core environment," in Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on, pp. 1–4, IEEE, 2011.

[12] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in 2010 3rd International symposium on parallel architectures, algorithms and programming, pp. 89–96, IEEE, 2010.

[13] Parida S, Panchal B. An efficient dynamic load balancing algorithm using machine learning technique in cloud environment. International journal of scientific research in science, engineering and technology. 2018;4(2018):1184-6.

[14] Li M, Zhang J, Wan J, Ren Y, Zhou L, Wu B, Yang R, Wang J. Distributed machine learning load balancing strategy in cloud computing services. Wireless Networks. 2019 Jul 6:1-7.

[15] Wang JB, Wang J, Wu Y, Wang JY, Zhu H, Lin M, Wang J. A machine learning framework for resource allocation assisted by cloud computing. IEEE Network. 2018 Apr 2;32(2):144-51.

[16] Achar R, Thilagam PS, Soans N, Vikyath PV, Rao S, Vijeth AM (2013, December) Load balancing in cloud based on live migration of virtual machines. In: 2013 annual IEEE India Conference (INDICON), pp 1–5

[17] Magalhães D, Calheiros RN, Buyya R, Gomes DG (2015) Workload modeling for resource usage analysis and simulation in cloud computing. Comp Elect Eng 47:69–81

[18] Dam S, Mandal G, Dasgupta K, Dutta P (2015, February) Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing. In: Proceedings of the 2015 third international conference on computer, communication, control and information technology (C3IT), pp 1–7

[19] Dave A, Patel B, Bhatt G (2016, October) Load balancing in cloud computing using optimization techniques: a study. In: International Conference on Communication and Electronics Systems (ICCES), pp 1–6

Authorized licensed use limited to: Seoul National University. Downloaded on November 09,2022 at 19:39:54 UTC from IEEE Xplore. Restrictions apply.