ENHANCED NEURAL NETWORK SCHEDULING FOR LOAD BALANCING IN MULTI-CLOUD

S. Selvakumar
Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, India

Abstract
In the cloud, numerous problems are encountered, owing to the fact that this environment is a closely connected environment. As the nodes continue to switch from one stage to the other in respect of specifications, problems can also be found in the field of service distribution and in the level of service delivered. The positioning of contact nodes remains unchanged. The efficiency of cloud service is influenced by criteria such as throughput, load, latency and even more adversely. The above parameters are linked to the user requests for a load balancing technique. Certain load balancing techniques have been addressed in this paper to increase the level of operation in the cloud world. This paper introduces and changes a structure for the provision of resource and load balancing. The system suggested relies on an optimization algorithm for binary ant colony to perform ideally as regards cost and cost. A workflows system was suggested for preferably loading and advancing the use of underused VMs. The findings show a stable low cost trend with a minimal number of VMs and steadily grows with a rise in VMs.

Keywords:
Recurrent Neural Network, Load Balancing, Workflow Execution

1. INTRODUCTION

Cloud helps people to access their services from anywhere through the internet. A customer will feel autonomous when using the cloud services and, in this situation, fixed sites are not needed. In this cloud world, many problems are encountered, due finally to the looseness of this environment [1]. When the nodes continue to switch from one point to the other about their specifications, the position of the communicating node is never secure, so difficulties are found in the field of service distribution and the level of service delivered. The efficiency of cloud service is influenced by criteria such as throughput, load, latency and even more adversely [2]. The above parameters are linked to the user requests for a load balancing technique. Certain load balancing techniques have been addressed in this paper to increase the level of operation in the cloud world [3].

The rapid growth of the cloud ecosystem depends solely on the creation of IT. The cloud ecosystem is a reservoir of different services. The users can deploy and access these services through the Internet. An appropriate membership in the applicable cloud setting can be accessed by the connectivity facility. In contrast of the company's tremendous cost to buy supercomputers, for instance an enterprise may rely on the cloud environment to do job computing work. By integrating the cloud, the above condition can be easily overcome [4].

Any individual who would like to use those services will rely on the cloud environment by registration or through paying. The Cloud environment contains multiple costly resources. Registry refers simply to the method of authorising a person to control the services deployed. Cloud access to resources has been made through a variety of services feasible. For example, a range of cloud processors may be deployed; users wishing to reach these processors have to address the cloud environment using a defined collection of services specifically configured for processors. The cloud ecosystem provides multiple levels of facilities, including the Service infrastructure (IaaS), the Service platform (PaaS) and the Software as a Service (SaaS).

You can conveniently access the above services through the internet and numerous other service providers such as the CSP (Cloud Service Provider). It is important to understand that such facilities can be operated at numerous sites. Service providers would operate on separate virtual computers, each of which would offer multiple services. This service providers can handle a variety of requests. Each service provider's request number is then limited to a given number. The computer does not process requests further than the allocated number. The cloud requests would never stay as frequent as numerous requests would be submitted for the same task and so the cloud environment is obliged to control the order and number of queries in order to make it easier for the cloud environment customer to approach them. Maintenance of an ecosystem that is balanced will ensure the customer that the requests are processed. The processing between the virtual machines and their resources can be regulated in a balanced cloud environment. We will only concentrate on SaaS for the enhancement of the efficiency of the cloud environment among the different kind of cloud services that occur in the environment.

Load balancing is achieved by sharing between the different virtual machines the load that progresses to the cloud controller. Sharing will be begun here on the basis of the available traffic. Before the start of the sharing process, the capability of the different virtual machines must be first examined. The adoption of online services would make it easier to access services from this environment.

For better use of cloud resources (processors, memory, circles), load balancing is necessary and is superior to the machine. VMs that are made available on physical computers are reserved and used for services. When companies are booked on VMs, it could be that part of the VMs are over-used, whilst others are under-used. Load balancing strategies are used to ensure that a cloud computer conducts at any time around the equivalent number of tasks. In cloud computing, user demands are incredibly specific in nature, and multi-tenure has to be constrained by different customers from each other and the cloud base. Different heuristic and metaheuristic strategies for traffic between open VMs and ideal execution of VMs were used. The computational costs of meta-heuristics are more than heuristic approaches, as they have a greater room for investigation and their hunting parameters rely on the subjective driven approach to achieve a scan solution for this particular planning problem. Heuristic devices are used to decrease the scanning space for meta-heuristic devices to obtain the optimal arrangement in a minimal possible period. Nevertheless, load balancing is a multi-objective query, with the overarching aim that the basic goal would be to convey occupations or instructions through available infrastructure in order that the relative lopsidedness is restricted.

This paper introduces and changes a structure for the provision of resource and load balancing. The system suggested relies on an optimization algorithm for binary ant colony to perform ideally as regards cost and cost. A workflows system was suggested for preferably loading and advancing the use of underused VMs.

2. RELATED WORKS

Different strategies for the load balance issue, where few were discussed, are therefore available. The Honey Bees Inspired Approach for Optimization [5] is among them, and it takes an optimization algorithm known as the Bees Algorithm to an optimum solution, which has been adequately inspired by the real bees forage behaviour. The algorithm was found to be paired with a random exploratory check for an exploitative neighbourhood. This article clarified and thus applied to optimise many benchmark functions and contrasts the results with those already obtained from various optimization algorithms, the normal drying activity of honeybees, the fundamental bees algorithms and the improved versions thereof.

Next is a genetic algorithm (GA), which relies solely on cloud computing load balancing techniques, [6] which implements a new load balance mechanism based on the genetic algorithm (GA). The algorithm attempts to balance the load of a cloud infrastructure while also aiming to reduce the scope of a certain mission. The Cloud Analyst simulator has demonstrated the GA load management technique.

Next solution is the Cloud Balancing Mechanism – A Soft Computing Approach, [5] aims at offering a load-balancing approach based on the Stochastic Hill climbing mechanism. A local optimisation technique called Stochastic Hill climbing is used to delegate incoming jobs to servers. A cloud analyst is used to analyse both the qualitative and the quantitative efficiency of the algorithm.

Cross-Breed Work Timetable is another method to minimise server load using RBAC at cloud [8], which is based on a mix of FCFS and priority and primarily controlled by RBAC. Cross-Breed Job scheduling strategy (Role based access control). The RBAC would be responsible for checking the responsiveness of the user to a given content. If the user does not have sufficient access rights to the relevant content, the user is refused immediate use rights and the server is then minimised.

Another solution depicts a Partitioning Public Cloud Load Balancing Model [9] which proposes to improve the public cloud's load balancing model in relation to the cloud partitions principle. Strategies should be chosen according to circumstances; this was achieved using a transfer process. The job of compensating for the burden on the server was to break the public cloud into separate units and tactics. This document provides a configuration with a main controller, juggling computers and servers. The key controller's role is to pick an appropriate balance sheet for a specific task. The balancer's function is to pick a minimally loaded node. The above strategy will spread the load effectively among the servers, by selecting the least loaded node, and create balanced cloud environment [7].

A Server-based Load Balancing Model [10] demonstrates a stronger Cloud Partitioning mechanism for the Job Seekers Web Portal where the allocated works and partitions are based on the arrival date. Here, Main Controller (Admin) is used for the load balance mission. The Improved Load Balancing Solution was planned to prevent Deadlocks in cloud [11] and a load balance algorithm was developed to ignore the Virtual Machinery (VM) deadlocks as VM migration processes applications coming from users. This paper also provides the predicted outcomes in accordance with the implementation of the algorithm suggested. Failure to solve impasse would potentially boost the number of jobs served by the cloud service provider, thereby increasing the working efficiency and industry of the respective cloud service provider.

The next methodology was the threshold – Hierarchical Cloud Computing Distribution Scheme [12]. Instead of researching the framework for mapping physically accessible resources to virtual resources in order to achieve optimal usage of resources in cloud computing settings, this analysis is carried out at application level. The following approach was adopted. In addition to allocating resources required to satisfy the high demand, a threshold-based hierarchical resource assignment system has been implemented to ensure that the virtual resources are automatically distributed between cloud computing systems in order to maximise resource resource reallocation decisions.

The above-mentioned methods lead to poor balance of load and further increase latency and result in a lower throughput ratio.

3. PROPOSED METHOD

The neural network scheduling algorithm (NNS) built in this research is based on concepts of improving learning or learning. NNS algorithm is an agent designed to effectively plan activities across computer nodes. The principal component of the algorithm is a neural network to determine and choose currently the most advantageous task assignment for a resource (action). The architecture of the network is shown such that m tasks and n processing resources are appropriate. Input state staking, which takes into account the state of the computer environment, working-flow configuration and present timetable, chosen activities and resources are coded into vector parameters.

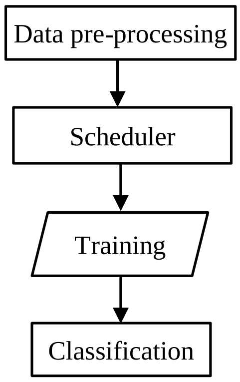

Fig.1. Neural Network Scheduling (NNS) scheme

The parameter vector content consists of several categories: general working flow parameters; work parameters; resource parameters; resource assignment parameters. This planner architecture allows for complex planning without regard to the problem dimension. It should be remembered that, if the measurements are necessary, multiple schedulers will function concurrently and be responsible for numerous activities or services, including their capacity to structure differently and to be qualified under various circumstances.

The agent of the algorithm communicates with the computation world, receives knowledge about the load, workload, current schedules and output models for assessment of tasks execution from it in the current state. If input status s is converted into vector parameters, the input in the neural network is supplied. The most successful behaviour is predicted by the neural network- a task selection planned at this time, and a node selection.

NNS agent sends this behaviour to the applicable computer environment and adds the assignment to the schedule for the given resource. If the assignment has been prepared, the computer environment measures the existing workflow, its new status, and assesses a NNS agent incentive *r*. Agent NNS saves {*s*,*a*,*r*,*s*} in the memory of obtaining the sequence. While obtained state is not terminal, the NNS agent carries out the new input state behaviour. SARS stores data in a form of *s*,*a*,*r*,*s* for learning in the context of the neural network.

This thesis focuses primarily on forming a series of criteria for coding environmental conditions. The algorithm is expected to be able to schedule the heterogeneous computer loads in a complex fashion. Since workflows vary, input data and scale can differ accordingly. This results in the algorithm needing to deal with relative values. To that end, we logically describe the worst workflow time for WT to go over to the relative values of the parameters (W_f). Theoretical assessment of the worst runtime is the amount of the execution time of all tasks delegated to the lowest by capacities and the time it takes, regardless of the need for those transfers, to transferring all data through the network. The majority of the parameters in the following state are specified by WT consumption.

In addition, we add column to display the parameters used during the experiment. Parameters like 'pan,' height and the number of parents or children have not been used because the weight of these values cannot be displayed properly when multiple W_fs are planned. While our algorithm has generated improved results without them, we will try to incorporate them in a relatively different form in our future work. The relationship between tasks is determined by the parameter data in the current form of our input status.

The measurements of the input state vector determined with the set of parameters used are 5+*n*+3 is the totaling, 5+*n** is both being where *n* is the number of nodes on which our NNS agent was designed For instance, architecture with five tasks and 4 nodes has an input vector of 64. For selection by NNS schema, only accessible tasks can be considered. When tasks are less than m in size, the parameters of the empty tasks may be set to zero.

4. RESULTS AND DISCUSSION

A hybrid task has to be performed to offset the load between the VM's in the cloud service. Measured by increasing the amount of VMs in the cloud, the current and proposed method performance.

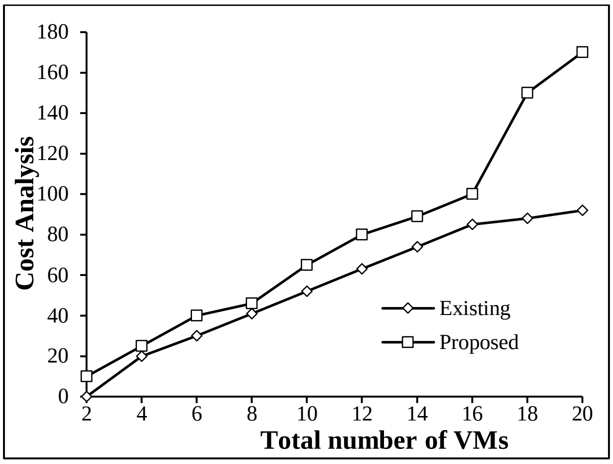

Fig.2. Average Makespan (ms) analysis

The Fig.2 compares the average makespan analysis of current and proposed approaches when the Genome Workflow is running in the presented process. As Ligo executes workflow, when the two VM's are used with the current system as a resource, the ms have received a cost analysis of 80 and reduced by 4 VMs and then raised up to 14 VMs. The effect is that it rises significantly by 160 and finishes by around 120 at 20 VMs. As 2 VMs was used as the resource of the current system, which entered the cost analysis at near proximity to zero and soon increased by 4 VMs when deployed and eventually decreased by 14 VMs, finally the number of VMs increased a little further by around 100 when 20 is achieved.

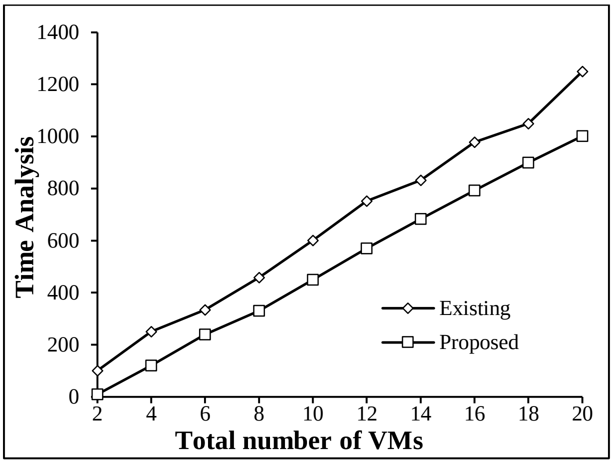

Fig.3. Average Cost analysis

The Fig.3 shows a comparison of the average time study of the current and the proposed system during workflow in the HDD-PLB setting. In the case where 2 VMs was used as a resource for current process, the time analysis obtained was 2.5 and soon decreased when used by 4 VMs, and later decreased to 14 VMs, when the number of VMs reached 20 it eventually increased marginally to 1.5. Although 2 VMs are used in the proposed system as a resource, the time analysis of 80 increases when 4 VMs are implemented and then increases until 14 VMs are used, it eventually increases significantly by almost 1 when the numbers of VMs are 20. The findings of the time study measured by the three scientific workflows are higher than the present method. The findings show a stable low cost trend with a minimal number of VMs and steadily grows with a rise in VMs.

5. CONCLUSION

Load balancing methods are used to ensure that the same number of tasks is carried out at every time on any system in the cloud data centre. In cloud computing, user demands are incredibly specific in nature, and multi-tenure has to be constrained by different customers from each other and the cloud base. VMs that facilitate the physical system distribute and utilise their services. The system suggested relies on the optimization algorithm for binary ant colony. The findings of the time study measured by the three scientific workflows are higher than the present method. The findings show a stable low cost trend with a minimal number of VMs and steadily grows with a rise in VMs.

REFERENCES

[1] V. Baby and Subhash N. Chandra, "Privacy-Preserving Distributed Data Mining Techniques: A Survey", *International Journal of Computer Applications*, Vol. 143, No. 10, pp. 37-41, 2016.

[2] P.V. Krishna, "Honey Bee Behavior Inspired Load Balancing of Tasks in Cloud Computing Environments", *Applied Soft Computing*, Vol. 13, No. 5, pp. 2292-2303, 2013.

[3] J. Brickell and V. Shmatikov, "Privacy-Preserving Classifier Learning", *Proceedings of International Conference on Financial Cryptography and Data Security*, pp. 1-6, 2009.

[4] D. Puthal, M.S. Obaidat, P. Nanda and M. Prasad, "Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing", *IEEE Communications Magazine*, Vol. 56, No. 5, pp. 60-65, 2018.

[5] B. Yuce, M. Packianather and E. Mastrocinque, "Honey Bees Inspired Optimization Method: The Bees Algorithm", *Insects*, Vol. 4, No. 2, pp. 646-662, 2013.

[6] K. Dasgupta, B. Mandal, P. Dutta and J.K. Mandal, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", *Procedia Technology*, Vol. 10, pp. 340-347, 2013.

[7] E. Bertino, I.N. Fovino and L.P. Provenza, "A Framework for Evaluating Privacy Preserving Data Mining Algorithms", *Data Mining and Knowledge Discovery*, Vol. 11, No. 2, pp. 121-154, 2005.

[8] A. Kaur and N. Bansal, "Cross-Breed Job Scheduling for Reducing the Server Load using RBAC at Cloud", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 3, No. 5, pp. 1-13, 2013.

[9] D. Thazhathethil, N. Katre, J. Mane Deshmukh and M. Kshirsagar, "A Model for Load Balancing by Partitioning the Public Cloud", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, No. 1, pp. 2466-2471, 2014.

[10] R. Agrawal and R. Srikant, "Privacy Preserving Data Mining. ACM SIGMOD", *Proceedings of International Conference on Management of Data*, pp. 439-450, 2000.

[11] K.S. Rashmi, V. Suma and M. Vaidehi, "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud", *International Journal of Computer Applications*, Vol. 6, No. 2, pp. 31-35, 2012.

[12] R. Varatharajan, G. Manogaran and M.K. Priyan, "A Big Data Classification Approach using LDA with an Enhanced SVM Method for ECG Signals in Cloud Computing", *Multimedia Tools and Applications*, Vol. 77, No. 8, pp. 1-24, 2018.