A BAYE'S THEOREM BASED NODE SELECTION FOR LOAD BALANCING IN CLOUD ENVIRONMENT

Naidila Sadashiv[1] and Dilip Kumar S M[2]

[1]Dept. of Computer Science and Engineering, Acharya Institute of Technology, Bangalore, India
[2]Dept. of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore, India

ABSTRACT

Cloud computing is a popular computing model as it renders service to large number of users request on the fly and has lead to the proliferation of large number of cloud users. This has lead to the overloaded nodes in the cloud environment along with the problem of load imbalance among the cloud servers and thereby impacts the performance. Hence, in this paper a heuristic Baye's theorem approach is considered along with clustering to identify the optimal node for load balancing. Experiments using the proposed approach are carried out on cloudsim simulator and are compared with the existing approach. Results demonstrates that task deployment performed using this approach has improved performance in terms of utilization and throughput when compared to the existing approaches.

KEYWORDS

Load Balancing, Baye's Theorem, Task Deployment, Cloud Computing

1. INTRODUCTION

Cloud computing is a new approach of computing that has been gaining huge popularity. It provides IaaS, PaaS and SaaS services to users and the usage of the service involves minimal effort in terms of cost and time. NIST has defined cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. This has lead to the increase in the demand for cloud computing services and thereby the cloud servers experience a huge amount of requests [2]. These requests are not consistent as the requests are made on the fly and there may arises burst requests frequently. In such scenarios, the incoming requests should be placed in an appropriate cloud node that ensures reliable task requests execution completion. The burst task request deployed causes imbalance of load across the cloud nodes if they are previously loaded. Also, the performance of the cloud node and the service rendered to the users get hampered. By the nature of cloud applications which are usually burst and short time, the existing load balancing approaches for task deployment leads to idle time in cloud nodes and it is a waste of resources.

Load balancing plays a vital role in improving the cloud node performance. Based on the load balancing work in the literature, the approaches are grouped as static and dynamic based on how the changing load and resource status are been considered [3, 4, 5]. Static approach commonly includes round robin, weighed round robin approaches, etc. They are simple and relies on static information and do not include the changes. Hence, they are not much appropriate in the cloud

DOI: 10.5121/ijccsa.2017.7101

environment. It is well known from the literature works that both an efficient capacity planning and load balancing are identified as a means of delivering acceptable service performance for users while maximizing the throughput and resource utilization. With this motivation, a Baye's theorem based node selection for load balancing (BNSLB) is proposed that identifies an optimal node for task deployment. The proposed work relies on the capacity planning presented in the earlier work in [18]. Based on the resource request made by the tasks and the available resource at the cloud node that is greater than the request made is clustered. Later, an optimal node is selected from the cluster to deploy the tasks.

The rest of this paper is organized as follows: Section 2 presents the related work followed by problem formulation in Section 3. In Section 4, the proposed strategy to address the problem formulated is presented. Performance evaluation and simulation results are discussed in Section 5 followed by Conclusion in Section 6.

2. RELATED WORK

This section discusses about the research work carried out to perform load balancing in the cloud environment.

Dam et al. [6] presented a load balancing algorithm for the VMs on the basis of hybrid approach using genetic algorithm and the gravitational emulation local search in cloud. These approaches minimizes the make span and meet the deadlines. Kulkarni et al. [7] proposed VM load balancing algorithm using a reservation table for VM allocation and requests mapping to the VMs with improved response time. Rahman and Graham [8] proposed a hybrid provisioning approach on the basis of static and dynamic allocation. Live migration is adapted for the placement of VMs in order to handle the changing load to improve the efficiency in cloud data centers. In a similar direction, VM migration and placement strategy was used for load balancing in cloud computing in [9]. Dupont et al. [10] presented a resource allocation framework for VM in a cloud data center. It computed best option of VM placement to obtain load balancing in cloud data centers. Zhao et al. [11] presented a self-adaptive multi-objective oriented load balancing approach MOGA-LS. It is a heuristic and self-adaptive approach based on genetic algorithm and the theory of Pareto optimal solutions. The problem of balancing the load was addressed in a decentralized service networks by Ranjan et al. [12]. A software fabric was introduced to balance service provisioning requests among the VMs deployed among the structured P2P cloud environment. Meirong et al. [13] presented a framework with the design and evaluation of decentralized and self-organizing service provisioning. It also demonstrated a self-organizing approach to balance the load among the peers that offered different type of services in a network. Resource allocation relying on VM-multiplexing was proposed by Sheng et al. [14] in a peer-to-peer environment. Proportional share model was incorporated to perform the resource allocation. This approach maximized resource utilization and execution efficiency.

In the above work, the approach is likely to result in a comparatively lower utilization of physical host resources with higher cost of the cloud data center. Hence, in this paper, the proposed BNSLB approach provides a solution to dynamically obtain the optimal load balancing with reduced computation complexity and thereby obtain the expected service performance and efficiency for both the providers and users.

3. PROPOSED PROBLEM AND ITS FORMULATION

3.1 Problem Description

In the cloud environment, every incoming user's requests are handled by deploying them on the available node in the resource pool. The first available node is selected to serve the request. If requirement of the requesting task is more than the serving capacity then the task deployment is unsuccessful. The execution and the performance of the running tasks gets affected by the deployment of the new task. Hence, there arises a need for load balancing strategy that selects an optimal node for task deployment and improves the resource utilization and task throughput.

3.2 Problem Formulation

Assume a set of n cloud nodes on which m user's tasks at time window Δt can be deployed on the cloud node. The tasks request are represented as *R={r1, r2, r3,... rm}*. The set of nodes feasible to be chosen for deployment is denoted as *FN = {fn1, fn2, fn3 ... fnn}*. This set is initially empty and is populated on the basis of two attributes that include the remaining resource amount of the cloud node *RC = {RC1, RC2, RC3 ... Rcn}* and the request made. The set *RC* represents the remaining capacity at each node *i € (1, n)* in terms of CPU and memory. Based on *RC* and the maximum requested resource among the tasks, prior probability of the nodes is set. Baye's theorem given in Equation (1) is considered for computation of posterior probability and is used for identifying the optimal node for solving the load balancing problem.

$$P(B_i|A) = (P(A|B_i)* P(B_i))/P(A), \quad i \in (1, n) \qquad (1)$$

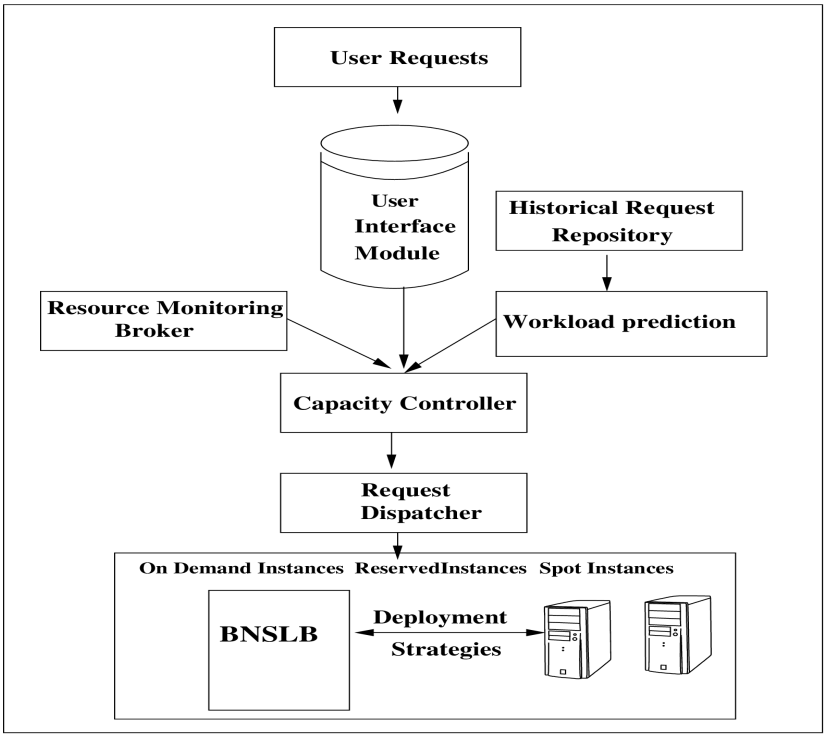

Figure 1. The view of BNSLB's architecture

4. OPTIMAL CLOUD NODE SELECTION USING BAYE'S THEOREM

The architecture of the proposed BNSLB approach in the cloud environment is shown in Figure 1. In coming user's task requests are handled by user interface module. User tasks requirements are queried along with the interaction from historical request repository to provide the request details to the capacity controller. This controller in turn plans the resource capacity among the three pricing plans that includes on-demand, reserved and spot instances resource pools [18]. The request tasks that arrive at these resource pools are deployed on the appropriate cloud node through the load balancer designed on the basis of Baye's Theorem given in Equation (1).

The event A is used to denote that the tasks are executed on a physical node. Event Bi denotes that the node i is selected and the probability $P(B_i)$ of selecting the node *i* is

$$\mathbf{P(B_i) = 1/n,} \quad \boldsymbol{i \in (1, n)} \qquad \mathbf{(2)}$$

Among all the cloud nodes in the pool, initially the node which has larger amount of available resource than the maximum requested resource is identified. It is more suitable for handling the tasks from the view point of performance and load balancing. Hence, based on the remaining capacity of the node and the maximum resource request made by the user's tasks, the prior probability of the node in feasible node set *FN* is set using Equation (3).

$$\mathbf{P(A|B_i) = 1 - Max\ (R_j)/RC_i,} \qquad \boldsymbol{i \in (1, n)\ and\ j \in (1,m)} \qquad \mathbf{(3)}$$

After the prior probability is given to the identified nodes finally, posterior probability is computed through Baye's theorem as given below in Equation (4).

$$\mathbf{P(B_i|A) = (P(A|B_i)* P(B_i))/P(A),} \qquad \boldsymbol{i \in (1, n)} \qquad \mathbf{(4)}$$

where $$\mathbf{P(A) = \sum P(A|B_i) * P(B_i),} \qquad \boldsymbol{i \in (1, n)} \qquad \mathbf{(5)}$$

The computed posterior probability at each node is used for the identification of the optimal node for task execution and load balancing. The deployment strategy incorporated is given in Algorithm 1.

Algorithm 1 Baye's Theorem based Node Selection for Load Balancer

---

Input: Tasks *m*, Serving Nodes *n*, Task Requests *R*, Remaining Capacity *RC*;
Output: Tasks deployment on *FN*;

```
1:      FN=NULL;
2:      for each i € n do
3:              Max-Request=max(Ri);        //Maximum requested resource amount in
the set R
4:      end for
5:      for each i € n do
6:              if (RCi> Max-Request) then
7:                      FN=Ni;              // Feasible set of cloud nodes
8:              end if
9:      end for
        for each i € n do
//Using Equation (2)
```

// Computed based on remaining capacity of the node and the resource request made
11: PP_i=Compute-Prior-Probability( );
12: **end for**
10: **for** each i € n **do**
//Using Equation (3)
11: PP_i=Compute-Posterior-Probability( ); // Posterior probability value obtained
12: **end for**
13: **for** each i € n **do**
14: Sort-descending(PP_i); //Arrange based on highest probability
15: **end for**
16: **for** each j € m **do**
17: **for** each i € n **do**
18: **if** (RC_i-CPU > R_j-CPU && RC_i-MEM > R_j-MEM)
19: Task(R_j) assigned to FN_i ;
20: $RC_{i=}$ RC_i-R_i;
21: **end if**
22: **end for**
23: **end for**
---

5. PERFORMANCE EVALUATION

In this section, the simulation setup for evaluating the BNSLB algorithm, results and discussion of evaluations are presented. Comparison with baseline round robin (RR) load balancer [16] approach are carried out considering different performance parameters.

5.1 Simulation Setup and Evaluation

To evaluate the load balancing algorithms, a discrete event simulator cloudsim [17] is used. Simulation setup used in this paper is according to the setup used in the earlier work in [18]. Cloud resource pool of 50 heterogeneous hosts are simulated to handle 100 heterogeneous applications of different sizes and completion time that range between 1-1000 minutes are considered. The parameters such as throughput, tasks completed and resource utilization are considered.

5.2.1 Comparison of BNSLB with baseline policies on throughput

In the simulation scenario, BNSLB is compared with the baseline round robin (RR) load balancer approach on throughput, which is the count of number of tasks completed at time t among the requested tasks. The simulation results of the two approaches are shown in Figure 2.

In case of RR approach, the requests are deployed in a cyclic fashion among the cloud hosts. During this occasion the current load is not considered due to which the handling capability decreases leading to smaller throughput. In the proposed approach BNSLB, the posterior probability along with individual resource availability at each host node together are used for the identification of the optimal node for request task execution.

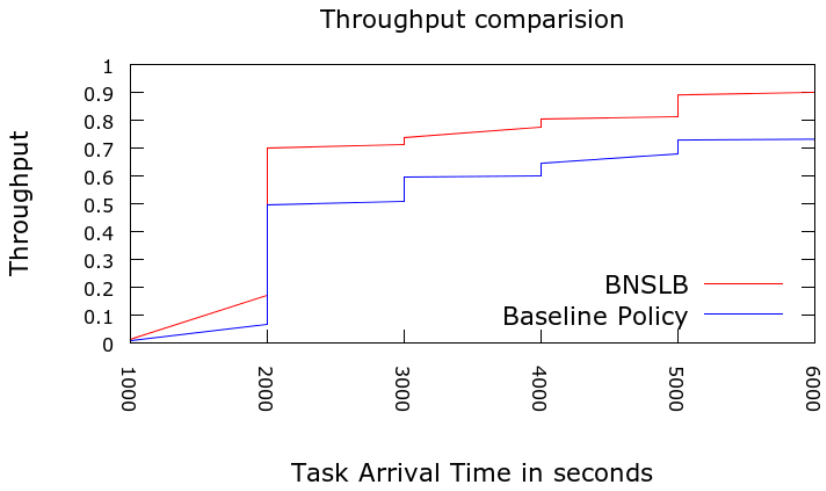

Figure 2. Comparison of the task throughput

This simulation illustrates that BNSLB has better load balancing effect along with good throughput relatively.

5.2.2 Comparison of BNSLB with baseline policies on cancelled tasks

Simulation of cancelled tasks are simulated by Cloudsim by reducing some number of hosts. In, RR while the requested tasks are deployed on the host nodes, few task execution fails due the lack of required resources at the host nodes. In case of RR, the load balancing approach as discussed above will map the incoming session in a cyclic order and some session gets cancelled due to resource unavailability. Whereas in the proposed approach, the posterior probability computed helps to balance the load from the long term point of view leading to few cancelled tasks as shown in Figure 3.

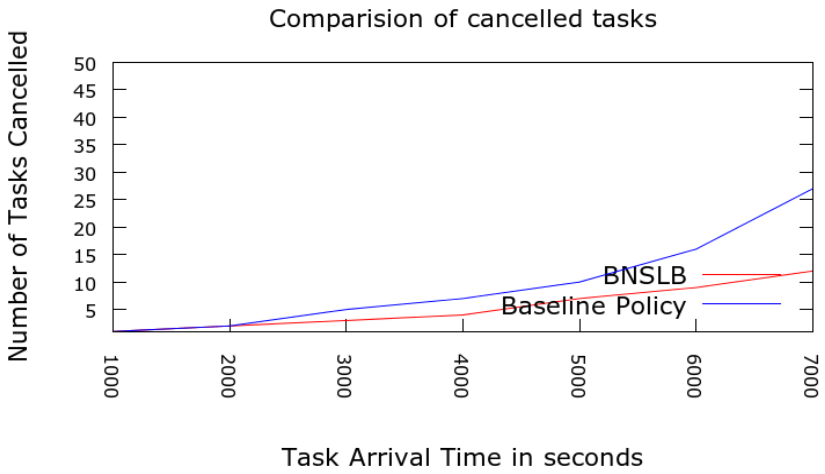

Figure 3. Comparison of the number of tasks cancelled

5.2.3 Comparison of BNSLB with baseline policies on resource utilization

The resource utilization in terms of CPU is presented in Figures 3a and 3b. The results demonstrates that under baseline RR approach, the load is uniformly distributed and is fully utilized that may lead to task cancellation. This is not the scenario in the proposed BNSLB approach as can be seen Figure 3b. The lack of resources for the running tasks under RR causes performance degradation compared to proposed approach.

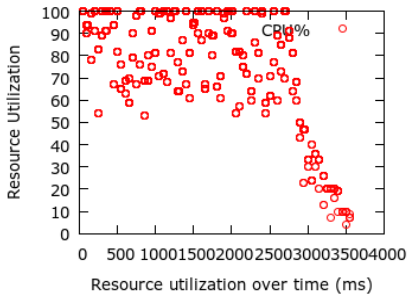

Figure 3a. Utilization Baseline Policy

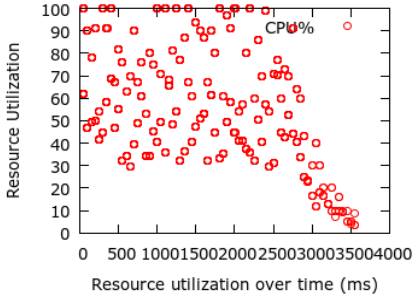

Figure 3b. Utilization BNSLB

Finally, based on the results it can be noticed that better load balancing effect is made using the BNSLB in the cloud environment.

6. CONCLUSION

This paper has proposed an approach to deploy tasks in order to balance the load among the cloud nodes using the Baye's theorem. Posterior probability value of the cloud nodes are computed based on which the optimal node is selected to deploy the incoming user's task. Simulation result reveals that better load balancing is achieved by incorporating the proposed BNSLB algorithm. Tasks are efficiently deployed over the cloud nodes with improved throughput when compared to the existing policy.

The future plan is to implement the algorithm in real cloud setup and test the load balancing effect by considering the real workload.

REFERENCES

[1] P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2012.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "A view of cloud computing", Communication ACM, vol. 53, no. 4, pp. 50–58, April 2010.

[3] Q. Wei, G. Xu, and Y. Li, "Research on cluster and load balance based on Linux virtual server", in Proc. International Computing Applications, vol. 105, pp. 169–176, 2011.

[4] W. Chen, Y. Zhang, and Z. Xiong, "Research and realization of the load balancing algorithm for heterogeneous cluster with dynamic feedback", Journal of Chongqing University, vol. 33, no. 2, pp. 2–14, 2010.

[5] S. Song, T. Lv, and X. Chen, "Load balancing for future internet: An approach based on game theory", Journal of Applied Mathematics, vol. 2014, no. 2014, Article ID 959782, Feb. 2014.

[6] S. Dam, G. Mandal, K. Dasgupta, and P. Dutta. "Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing", in Third International Conference on Computer, Communication, Control and Information Technology (C3IT), pages 1–7, Feb 2015.

[7] A.K. Kulkarni and B. Annappa, "Load balancing strategy for optimal peak hour performance in cloud datacenters". in IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), pages 1–5, Feb 2015.
[8] M. Rahman and P. Graham, "Hybrid resource provisioning for clouds", Journal of Physics: Conference Series 385 (2012) 012004, 2012.
[9] J. Zhao, Y. Ding, G. Xu, L. Hu, Y. Dong, and X. Fu, "A location selection policy of live virtual machine migration for power saving and load balancing," The Scientific World Journal, vol. 2013, no. 2013, Article ID 492615, Sep. 2013.
[10] C. Dupont, T. Schulze, G. Giuliani, A. Somov and F. Hermenier, "An energy aware framework for virtual machine placement in cloud federated data centres," Third International Conference on Future Systems: Where Energy, Computing and Communication Meet (e-Energy), Madrid, 2012, pp. 1-10, 2012.
[11] Jia Zhao, Yan Ding, Gaochao Xu, Liang Hu, Yushuang Dong, and Xiaodong Fu, "A Location Selection Policy of Live Virtual Machine Migration for Power Saving and Load Balancing," The Scientific World Journal, vol. 2013, Article ID 492615, 16 pages, 2013.
[12] Ranjan R., Harwood A. and Buyya, R., "Peer-to-Peer-based Resource Discovery in Global Grids: A Tutorial", IEEE Communications Surveys Tutorials, vol. 10, no. 2, pp. 6-33, 2008.
[13] M. Liu, T. Koskela, Z. Ou, J. Zhou, J. Riekki, and M. Ylianttila, "Super-peer-based coordinated service provision", Journal of Network and Computer Applications, vol. 34, no. 4, Jul 2011.
[14] Sheng Di and Cho-Li Wang, "Dynamic Optimization of Multiattribute Resource Allocation in Self-Organizing Clouds", IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 24, pp. 464-478, 2013.
[15] Nikolay Grozev and Rajkumar Buyya. "Multi-cloud provisioning and load distribution for three-tier applications", ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 9, no. 3, pp. 1–22, 2014.
[16] http://aws.amazon.com/elasticloadbalancing/
[17] R. Buyya, R. Ranjan, and R.N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities", in International Conference on High Performance Computing and Simulation, pages 1–11, Jun 2009.
[18] N. Sadashiv, D. Kumar S M and R. S. Goudar, "Cloud Capacity Planning and HSI based Optimal Resource Provisioning", Second IEEE International Conference on Electrical, Computer and Communication Technologies (IEEE ICECCT 2017), Coimbatore, 2017.

AUTHORS

Naidila Sadashiv is an Assistant Professor the Department of Computer Science and Engineering, Acharya Institute of Technology, Bangalore. She received M. Tech degree in Computer Science and Engineering from Vishweswaraiah Technological University in 2006 and is a Ph. D Research Scholar at University Visvesvaraya College of Engineering, Bangalore University, Bangalore. She is involved in research and teaching B. E and M. Tech students and has more than 14 years of teaching experience. Her current research lies in the area of cloud computing.

Dr. S. M Dilip Kumar is an Associate Professor in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He received M. Tech and Ph. D degree in Computer Science and Engineering from Vishweswaraiah Technoloical University in 2001 and Kuvempu University in 2010 respectively. He is involved in research and teaching B. E and M.E students and has more than 20 years of teaching experience and guiding Ph. D students. He has published more than 40 papers in International Journals and Conferences. His current research lies in the areas of sensor networks and cloud computing.