A Dynamic Model for Load Balancing in Cloud Infrastructure

Jitendra Bhatia

***Abstract*—Building a private, public or hybrid IaaS cloud infrastructure, load balancing of virtual hosts on a limited number of physical nodes, becomes a crucial aspect. This paper analysis various challenges faced in optimizing computing resource utilization via load balancing and presents a platform-independent model for load balancing which targets high availability of resources, low SLA (Service Level agreement) violations and saves power. To achieve this, incoming requests are monitored for sudden burst, a prediction model is employed to maintain high availability and power aware algorithm is applied for choosing a suitable physical node for the virtual host. The proposed dynamic load balancing model provides a way to conflicting goals of saving power and maintaining high resource availability.**

Keywords*—cloud computing, adaptive load balancing, user request prediction, power aware, generic model.

I. Introduction

CLOUD computing is emerging as future of computing. Software companies are increasingly moving toward Software as a Service (SaaS) from the traditional model. However many existing proprietary and security-sensitive operations might not be entrusted to an external service provider. This gives rise to the need of Platform as a service (PaaS) and Infrastructure as a service (IaaS). The IaaS model is implemented in three different ways viz. private, hybrid and public cloud. A cloud service provider implements public cloud to provide computing and storage services. Enterprises implement an in-house cloud infrastructure called private cloud. In many scenarios enterprise may use a public cloud as an extension of their private cloud; such arrangement is known as a hybrid cloud.

In any such implementation of IaaS to achieve optimal resource utilization is a desirable goal. Effective load balancing is one way to achieve that. This paper presents a model for implementing load balancing in IaaS cloud infrastructure. It sees the set up as a virtual machine provider of various computing capacity and memory. There are many challenges in the form of tradeoffs in achieving the goal. A list of which is presented in section II of this paper.

A. Related Work

Many approaches in the form of load balancing algorithms are published in recent years. Authors of [1] present a systematic review of existing load balancing techniques in cloud computing. Various heuristics approaches are presented for load balancing by researchers using genetic algorithm [12], ant colony optimization [13] and artificial bee algorithm [14]. The paper [4] implements two different algorithms, greedy resource allocation during usual workloads and random resource allocation during bursty workloads. The load balancing strategy presented in [3] focuses on profit maximization and lowering SLA violations. The authors of [5] suggest using HTV dynamic algorithm for load balancing. A load balancer based on statistical prediction mechanism and available resource estimation mechanism is presented in [11].

Prof. J. B. Bhatia is with Computer Science and Engineering Dept., Institute of Technology, Nirma University, Ahmedabad, Gujarat, India Email: jitendra.bhatia@nirmauni.ac.in

B. Our contributions

Most of the above approaches have proposed a single algorithm to improve a specific aspect of load balancing which are based on either resource utilization, load prediction, SLA integrity or power saving. In this paper, authors present a complete unifying load balancing model which incorporates load prediction, high availability and power saving. It also provides enough flexibility for adding additional authentication, pricing, and geographical aspects as per the need of specific implementation. Our work can be summarized as follows: 1) We have introduced a system model which logically separates resource pool operations from user request handling. Now cloud controller which handles user requests is responsible for predicting upcoming demand and reducing SLA violations. While the component maintaining pool of resources namely cluster controller is responsible for ensuring optimal resource utilization and low power consumption. 2) We design an adaptive "on-off" switch which puts a cloud in over provisioning mode at the time of sudden bursts of requests. During the over provisioning mode cloud controller utilizes a user requests prediction model presented in [7]to turn on extra physical nodes to serve high demand. Otherwise, the cloud is in power saving mode. 3) We present a new power-aware resource allocation algorithm for cluster controller which is an improved version of PALB [6] that includes live migration of virtual machines to improve resource utilization and save power.

The remainder of the paper is organized as follows. Challenges for a practical power-aware load balancing algorithms are enlisted in Section II. Our system model for load balancing is proposed and discussed along with an energy conservation algorithm in Section III. In Section IV, we have stated our conclusion and future work.

II. Challenges

An ideal load balancing approach should consider the following factors and thrive to strike a balance among them.

- Instant resource provisioning is essential, delay in serving user request leads to poor QoS (Quality of Service).
- Reliability- data must be available at any time from any server which can be overcome through network coding[15]
- Adaptability – Load balancing should be adaptive in response to changing user request patterns.
- Elasticity – It should be elastic enough to easily scale up or down as per the need.
- Customizability – The algorithm should be customizable to include various business specific logics of security, isolation, billing, etc.
- Multilayer approach – User requests handling, infrastructure management, accounting, and monitoring should be separated in different modules.
- Ensuring high resource utilization – Utilization of each powered on the node must be maximized.
- Cost of VM migration – the cost of migration in terms of QoS parameters and network traffic should be taken into consideration before relocation for the sake of load balancing.
- Power consumption – Actual percentage of power used by a CPU in a node is represented in Fig. 1. It indicates that when a node is powered on 50% of the total power consumption is a fixed cost. Hence, the number of overall powered on physical nodes should be minimized for power saving.

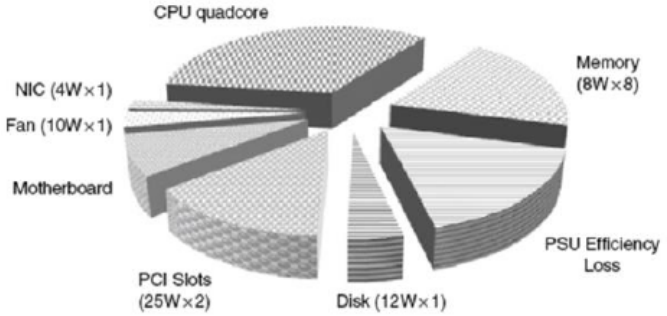

Fig. 1: Power Consumption

[9]

The proposed system model is an abstraction of various existing enterprise cloud and datacenter virtualization platforms. It is a generic model and can be easily implemented on any infrastructure setup. The model separates user management from infrastructure management to provide a greater flexibility.

A. System Architecture

The proposed cloud infrastructure, as shown in Fig.2 is comprised of a Cloud Controller, Cluster Controller, and many physical nodes which are capable of hosting virtual machines (VMs). The Cloud Controller is an administrative node and front-end for cloud infrastructure. It manages user requests of various instances and keeps track of virtual machines throughout their life cycles, including resource provisioning, monitoring, metering, and billing. The Cluster Controller controls the physical nodes and manages the networking between virtual machines, and between virtual machines and external users. It also collects the system data, such as utilization of resources, from the nodes. Cloud controller and cluster controller are logically detached entities; however they can be implemented on a single node for smaller implementations.

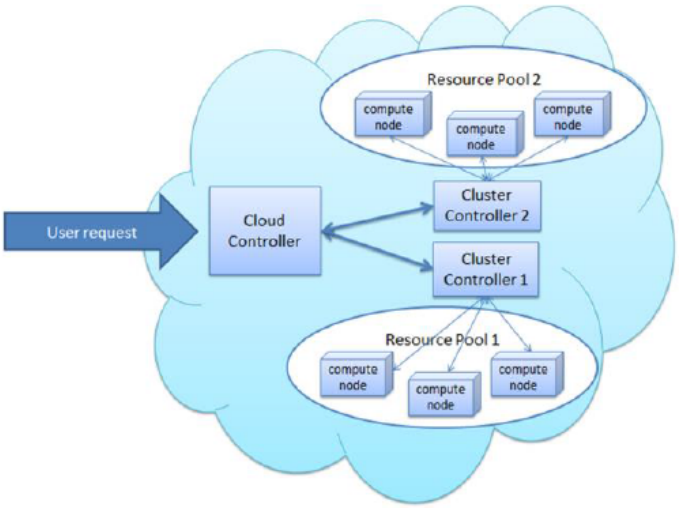

Fig. 2: Proposed System Architecture

The complete resource pool available is divided in one or more clusters for better manageability. The basis of clustering can be chosen by the cloud administrator. However, authors recommend the clustering should be done on the basis of the type of instances they provide. Nodes which serve specific types of instances, e.g. High-Memory Instances [8] should be grouped together for better performance of the model. Other criteria may include geographical, economical, departmental partition of users etc.

B. Notations

For the sake of simplicity, we have considered the user request to be comprised of only two specifications: memory and computational power. However, in real world implementation it may include several other parameters such as the number of cores, storage, networking capabilities etc. The proposed model can be configured to include any such criteria easily. Total capacity and utilization of each node is also measured in same units. Thus each pair (p, m) represent a measure of m GB of memory and p GHz of processing power. $(pi, mi) < (pj, mj)$ if and only if $pi < pj$ and $mi < mj$

- VMR – virtual machine request
- VMR_k – kth virtual machine request (pk, mk)
- NC – node capacity
- NC_p – pth node's capacity (pp, mp)
- I – Index of dispersion
- B – Burstiness threshold for change in I

- C_i – i^{th} cluster
- t_1 – time threshold for over utilization of a node
- t_2 – ideal time threshold for a node
- t_3 – time threshold for under utilization of a node
- The proposed framework of load balancing system is represented in Fig. 3.

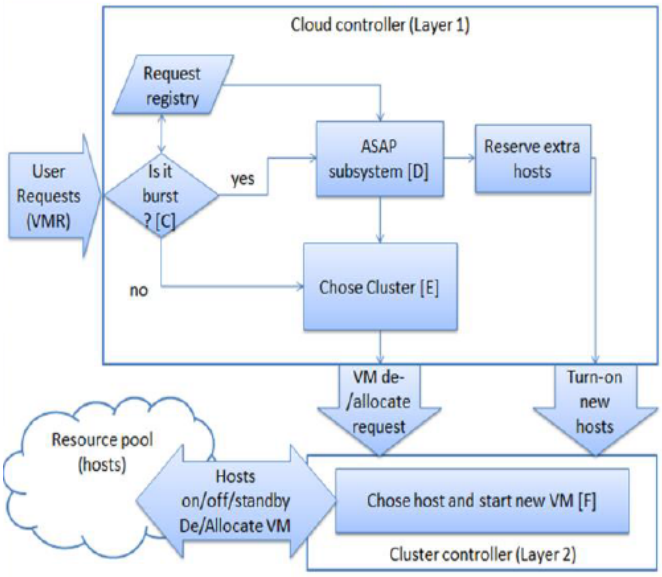

Fig. 3: Proposed Load Balancing Mechanism

C. Mode Switch

The index of dispersion is a powerful measurement of burstiness in workloads which makes it very practical for estimation. We use the index of dispersion I to capture the burstiness of user requests [2]]. The index is defined in Eq. (1),

$$I = \sigma^2/\mu \tag{1}$$

The adaptive mode switch plays the most crucial role in the load balancing model. It detects the upcoming burst of requests and switches on the prediction subsystem. Thus balancing between high availability and high utilization. Algorithm1 describes how an index of dispersion is be used as an on-off switch. Generally the index is calculated over an interval to capture the burstiness of the data. However, when implemented as an on-off switch, I must be calculated using a running mean and various. For that purpose, a Variance window and a Mean window is defined. They represent the number of previous requests which are to be considered while calculating the current value of their respective measure. For a given implementation of the model, these values are characterized by workload patterns. It is also suggested by the authors that the mean window must be larger (about four times) than the variance window. To test the efficacy of the presented algorithm in Algorithm1, we have tested it on various bursty as well as random data series. It is observed each time that whenever there is an upcoming surge of more than usual requests the algorithm switches to "on" mode and when the requests drop back to normal it returns to "off" mode. In both the cases for mean window = 16 and variance window = 4. it shows that the algorithm switches mode before every single spike.

Algorithm 1 ModeSwitch

```
1:  procedure MODESWITCH
2:      Initialization
3:      VarianceWindow: No of previous requests considered
        for variance (σ^2) calculation
4:      MeanWindow: No of previous requests considered for
        mean (μ) calculation
5:      Preferably MeanWindow = 4* VarianceWindow
6:      U_k : k^th previous user request
7:      MeanI: mean value of I
8:      for each new request calculate I do
9:          μ ← (Σ_{k=0}^{MeanWindow} U_k)/MeanWindow
10:         σ^2 = (Σ_{k=0}^{VarianceWindow} (U_k − μ)^2)/VarianceWindow
11:         I ← σ^2/μ
12:         Mode switch:
13:         if I > MeanI then
14:             Mode: On
15:         else
16:             Mode: off
17:     return
```

D. User Request Prediction Subsystem

If there is a burst of new VM requests then existing powered on nodes might not be able to fulfill it [6]. This may result in SLA violation and/or delay in resource provisioning, a scenario which must be avoided. To prevent such failures, we have included a user request prediction subsystem that predicts future requests. For the task, we have included ASAP: A Self-Adaptive Prediction System, which comprises of three-module that implements ensemble algorithm to predict different types virtual machine demands based on past requests[7]. Fig. 4 represents the ASAP subsystem.

In the original work of [7] the authors use the prediction to preempt the creation of virtual machines before the requests ever arrive. However, instead of pre-provisioning the VMs as suggested by Jiang et al. we prefer turning on enough new nodes in appropriate clusters, which will sufficiently satisfy future user requests.

E. Selecting The Right Cluster

Partitioning the resource pool of physical nodes into several clusters has certain obvious advantages in terms of better manageability and networking. It also provides certain scalability to the infrastructure and this load balancing model. It enables adding a large number of new physical nodes into the resource pool by only configuring a new cluster controller. If clusterization of the physical nodes is based on the types of user request they serve, as suggested by the authors, then any

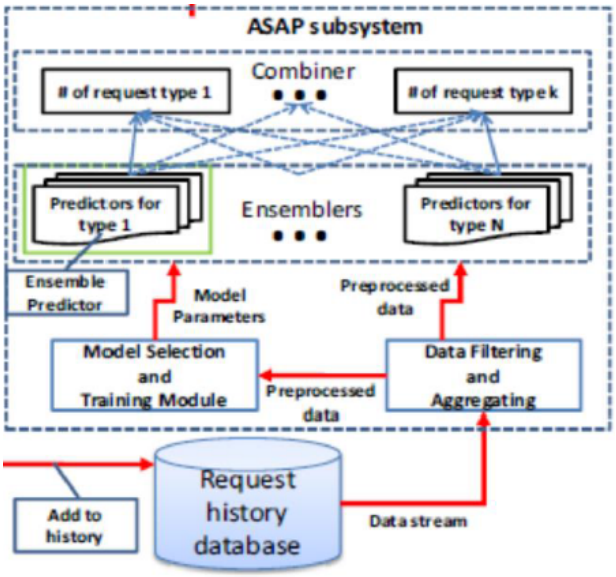

Fig. 4: User Request Prediction Subsystem

request can be directly mapped to a specific cluster based on the type. Otherwise, the algorithm shown in Algorithm2 can be implemented to select most efficient allocation among these clusters.

Algorithm 2 ClusterSelection

procedure CLUSTERSELECTION
  Initialization
  Number of Clusters: n
  $VMR : R \leftarrow (p, m)$
  $Selected\_cluster \leftarrow C_1$
  Send queries to each cluster for information of highest unallocated node capacity ← highest_NC
  / *Cluster Selection Process */
  for `each i = 1 to n` **do**
    if $[R \leq (C_i[highest\_NC])$ AND $(C_i[highest\_NC]) \leq (Selected\_cluster[highest\_NC])]$ **then**
      $Selected\_cluster \leftarrow C_i$
  return

The Select cluster algorithm simply chooses the clusters with highest capacity node available to fulfill the request. Accommodating more requests on single large capacity node helps achieving the goal of reducing overall numbers of physical nodes.

F. Power Saving Mechanism

The power saving mechanism of the proposed model is inspired by referred works of Galloway et al. on PALB:

Power-aware load balancing. The algorithm proposed in [6] claims to save more than 75 % of the power but fails to fulfill the SLA requirements of availability [10]. In the present proposed model, the availability is ensured by components presented in section 3(C) and 3(D). The algorithm in Algorithm3 is an improved version of PALB which utilizes efficient allocation and migration of VMs for power saving and better resource utilization. Cluster controller implements this algorithm as it maintains available node controller for all nodes of the cluster.

The power saving algorithm comprises of broadly three components. Firstly VM allocation finds the node with least residual capacity that can fulfill the user request and assigns a new VM on it. If the newly allocated physical node is the last one of the powered on machines it boots up a new one. Second is scaled up operations it ensures that no node is overworked and there are always enough resources to fulfill the largest possible request. Lastly, scale down operations which finds under utilized and utilized physical nodes and migrate VMs to other nodes shuts them down. Thus ensuring optimizes resource utilization.

Algorithm 3 PowerSaving

procedure POWERSAVING
  /* VM Allocation */
  if $[VMR_i \leq Only\_one\_NC]$ **then**
    Allocate VMR to available node
    Boot a new node with $NC \geq VMR$
  if $[VMR_i \leq More\_than\_one\_NC]$ **then**
    Allocate VMR to least NC_i
  /* Scale up*/
  if $[VMR_i > all\_NC]$ **then**
    Boot a new node with $NC \geq VMR_i$
  for `i=1 to n` **do**
    if $[NC_i < -5\% for\_time > t_1]$ **then**
      Turn on a new node N_k
    Attempt migrating least VM from $N_i to N_k$
  if $[All\_NC < Largest\_VMR\_Possible]$ **then**
    Boot a new node with $NC \geq$ largest_VMR_possible
  /* Scale down */
  if $[NC\_utilization \doteq 0\% for\_time > t_2]$ **then**
    Turn off the node
  if $[NC\_utilization \leq 75\% for\_time > t_3]$ **then**
    Attempt migration of VM to another node
    Put node in a low power mode
  return

III. CONCLUSION AND FUTURE WORK

Due to lack of infrastructure and resources such as real world user request history data, we could not implement or simulate the model and calibrate its potential. However, the proposed model provides a generic, customizable, multi-layer, and elastic approach to load balancing in cloud infrastructures. Our dynamic model is adaptive to a user request and predictive at the time of bursty workloads. The design approach focuses on ensuring high utilization, power saving, high availability, and low SLA violations. Constants and thresholds used in our model can be optimized for specific implementations. Also, we

will simulate our model and provide statistical data as well as a comparison with other approaches

REFERENCES

[1] Nidhi Jain Kansal and Inderveer Chana (2012) Existing Load Balancing Techniques In Cloud Computing: A Systematic Review. Journal of Information Systems and Communication. ISSN: 0976-8742, E-ISSN: 0976-8750, Volume 3, Issue 1, pp- 87-91.

[2] Giuliano Casale, Ningfang Mi, Ludmila Cherkasova, and Evgenia Smirni. 2008. How to parameterize models with bursty workloads. SIGMETRICS Perform. Eval. Rev. 36, 2 (August 2008), 38-44.

[3] Patel, Komal Singh; Sarje, A.K., "VM Provisioning Method to Improve the Profit and SLA Violation of Cloud Service Providers," Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference on , vol., no., pp.1,5, 11-12 Oct. 2012

[4] Jianzhe Tai; Juemin Zhang; Jun Li; Meleis, W.; Ningfang Mi, "ArA: Adaptive resource allocation for cloud computing environments under bursty workloads," Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International , vol., no., pp.1,8, 17-19 Nov. 2011

[5] Bhatia, Jitendra.; Patel, T.; Trivedi, H.; Majmudar, V., "HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud," Cloud and Services Computing (ISCOS), 2012 International Symposium on , vol., no., pp.15,20, 17-18 Dec. 2012

[6] Galloway, Jeffrey M., Karl L. Smith, and Susan S. Vrbsky. "Power Aware Load Balancing for Cloud Computing." Proceedings of the World Congress on Engineering and Computer Science. Vol. 1. 2011.

[7] Jiang, Yexi, et al. "Asap: A self-adaptive prediction system for instant cloud resource demand provisioning." Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE, 2011.

[8] http://aws.amazon.com/ec2/instance-types/

[9] L. Minas, B. Ellison, Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers, Intel Press, USA, 2009.

[10] Galloway, Jeffrey, Karl Smith, and Jeffrey Carver. "An Empirical Study of Power Aware Load Balancing in Local Cloud Architectures." Information Technology: New Generations (ITNG), 2012 Ninth International Conference on. IEEE, 2012.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[11] Zhenzhong Zhang; Haiyan Wang; Limin Xiao; Li Ruan, "A statistical based resource allocation scheme in cloud," Cloud and Service Computing (CSC), 2011 International Conference on , vol., no., pp.266,273, 12-14 Dec. 2011

[12] Zhu, Kai; Huaguang Song; Lijing Liu; Gao, Jinzhu; Guojian Cheng, "Hybrid Genetic Algorithm for Cloud Computing Applications," Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific , vol., no., pp.182,187, 12-15 Dec. 2011

[13] Nishant, K.; Sharma, P.; Krishna, V.; Gupta, C.; Singh, K.P.; Nitin, N.; Rastogi, R., "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on , vol., no., pp.3,8, 28-30 March 2012

[14] Jing Yao; Ju-hou He, "Load balancing strategy of cloud computing based on artificial bee algorithm," Computing Technology and Information Management (ICCM), 2012 8th International Conference on , vol.1, no., pp.185,189, 24-26 April 2012 carb

[15] Jitendra Bhatia; Ankit Patel, "Review on variants of Network Coding in Wireless Ad Hoc Networks," NUiCONE, 2011 Nirma University International Conference on, vol.,pp1-6, 8-10 Dec. 2011

[16] Jitendra Bhatia; Malaram Kumhar "Perspective Study on Load Balancing Paradigms in Cloud Computing," IJCSC, March-2015, vol.6,pp.112,120, March-2015

J B Bhatia received his B.E degree in computer engineering from North Gujarat University, India in 2004. He obtained his M.Tech degree in Computer Science from Nirma University, Ahmedabad, India in 2012. Since 2008, he has been working as an assistant professor, Institute of Technology, Nirma University, Ahmedabad, India. His area of interest includes cloud computing, Ad Hoc networks, Software Defined Networking.