IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 7, Issue 10, October 2018

A Review on: Network Load Balancing in Dynamic Data Center

Prof. Nitin Chopde[1], Shrikant Deshmukh[2]

Asst. Professor CSE Dept, G.H.Raisoni, AMT[1]

ME. Student, CSE Dept, G.H.Raisoni, AMT[2]

Abstract: Achieving efficient resource utilization and guaran-tees of required performance in data centers is difficult. Efficient load balancing, on various parts of data center, is the key to overcome these problems. This paper presents load-balancing in two parts of the data center: network traffic load balancing and virtual server load balancing. We review some of the proposed load balancing systems, and discuss their advantages and challenges.

Keywords: Network traffic load balance, Data center network, served system, virtual server load balance and load balance mechanism.

I. INTRODUCTION

The free lunch is over: recent challenges in the hardware development have forced the software engineers to search performance gains from concurrent, multi threaded and par-allelized programming. [1] At the same time, there are clear limitations in the speed up gains that parallel programs can achieve. [2] The challenges in parallelization also lead to new challenges in the utilization of the available computing resources.

The development of data center technology and virtual-ization, together with faster telecommunication networks and proliferation of mobile devices have lead in the adoption of cloud computing. Cloud computing paradigm provides a shared pool of easily configurable and flexible computing resources via convenient, on-demand network access. [3] One of the driving forces for cloud computing has been its promise of economics of scale; cloud computing centers can be built using cheaper hardware, cooling, electricity, network capacity and smaller number of administrators per computer, while at the same time alleviating the problem of inefficient resource usage.

However, to efficiently utilize data center resources, and to provide the required performance guarantees, efficient load balancing mechanisms are needed on various locations in the data center, all the way from single server core utilization up to the efficient network routing.

This paper presents load balancing solutions from two parts of the data center. In the Section II, we will present the challenges and proposed load balancing solutions related to the scaling of data center network. Section III presents the general idea and challenges behind virtual server load balancing. Finally, in Section IV we conclude our findings.

II. NETWORK TRAFFIC LOAD BALANCING

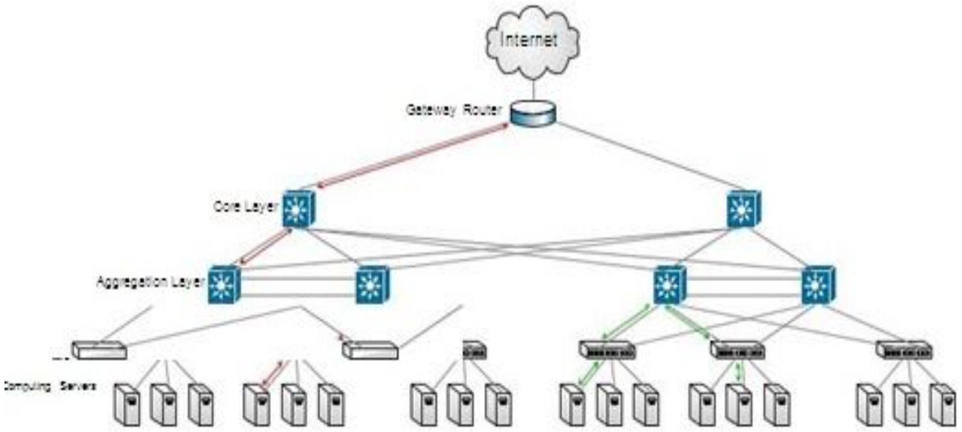

Figure: 1 Data Center Network Topology

Today's cloud computing data centers consist of hundreds of thousands of servers, supporting large number of services, such as big data applications, high performance computing and web services. Many of these applications have different com-ponents divided across the servers, and the network bandwidth becomes one of the major bottleneck in data centers.

A. Data Center Networks: For compatibility and cost reasons, the networks are often built from commodity Ethernet switches and routers (scaling out), rather than from expensive high end hardware (scaling in). Due to the limits in the switch port densities even in the high end hardware, the traditional single rooted topologies are replaced with multi-rooted tree topologies such as fat-tree or leaf-spine. Multi-rooted network design allows to scale the data center networks' bisection bandwidth by adding new spine switches into the network. Figure 2 presents an example of 3-level data center network topology, consisting of the core layer, aggregation layer, and access layer. Two-level tree is another typical choice. However, to efficiently utilize the theoretical bandwidth gains from multi-rooted design, the traffic needs to be load balanced across all the available network paths.

B. Load Balancing Mechanisms: Several different mechanisms have been proposed to over-come the load balancing problem. Many of the popular load balancing schemes, such as Equal-Cost Multi-Path (ECMP) forwarding [4], are based on flow hashing. In flow hashing schemes, the even load balancing is achieved by splitting the packet flows randomly across multiple network paths. Hashing based load balancing algorithms have been dom-inant in recent years, due to their easy implementation and good performance. However, the problem with hashing based designs is that the forwarding decisions are done purely locally and unaware of the congestion state of the system. The forwarding decisions done in asymmetric networks, for example when there is a problem in some of the paths, often cause congestion to the system. [5]

In ECMP the switch routing table contains several, equal cost, next-hop addresses for the same destination, and the forwarding decisions are done hop-by-hop based on the hash of the packets' header (5-tuple). The benefit of hash-based schemes is that the packets are automatically kept in order, which crucial for certain protocols such as TCP. [4]

Several different proposals of congestion aware load balanc-ing systems, such as CONGA [5] and Presto [6], have been made to overcome the issues in the hash based schemes. One of the problems of non-hashing schemes is often packet reordering. Both CONGA and Presto overcome the packet reordering problem by doing the load balancing decisions on per-flowlet, instead of per-flow or per-packet basis. A flowlet is a burst of packets belonging to the same flow, that are separated from other bursts in the same flow by a large enough gap, such that splitting them on a separate paths do not cause reordering problem. [5], [6]

CONGA makes the load balancing decisions based on the congestion state of the system. The congestion state of each path is maintained in the leaf nodes and is maintained with fast feedback loops carried directly in the hardware data plane of the switches. It utilizes the network virtualization overlay headers, VXLAN, to carry the information between the nodes, updating the congestion information on the path. [5]

Main drawbacks of CONGA, and many other reactive distributed load balancing schemes, such as Juniper VCF [7], is that they require customized networking hardware. Presto routes the flowlets through the network using round robin algorithm and implements all the functions in the soft network edge (virtual switches), thus not requiring any specialized infrastructure or changes to the transport layers. Presto's is implemented with only a few changes in the hypervisor soft-edge: 500 and 900 lines of code changed in Open Virtual Switch and Linux Generic Receive Offload, respectively. [6]

Other proposals, such as Hedera [8] and Planck [9], are based on centralized load balancing mechanism. Centralized mechanisms, with more precise information about the system state, is able to make better routing decisions in the best case, compared to local or host based mechanisms. However, the centralized schemes are often very slow and due to the large overhead of their control loops. [5], [6]

III. VIRTUAL SERVER LOAD BALANCING

Despite of the recent progress in the hardware efficiency, the data centers consume huge amount of electricity. [10],One of the main promises of cloud data centers is the efficient resource usage achieved through server consolidation and virtualization. To fully exploit the possibilities of virtual-ization, the server hardware needs to efficiently adapt to the workload, while still maintaining the required throughput and time requirements.

A. Load Balancing Method: Majority of the load balancers studied for this paper, tackle the load balancing problem by monitoring the physical server nodes and the resident virtual machines. The monitored met-rics, e.g. CPU-,

DOI 10.17148/IJARCCE.2018.71020

memory-, network usage, are then used both for efficiently allocating resources for new VMs, as well as migrating the existing VMs between the nodes. Data center load balancing algorithms are often categorized by how the load balancing responsibility is distributed in the system, and how the system state is considered when doing the load-balancing decisions. In centralized systems, the information and decisions are held by central load balancers, where as in the decentralized system, such as ..., the load balancing decisions are done also in the server level. The responsibility of the load balancing can also be a combination of centralized and decentralized schemes. In dynamic load balancing systems, the load bal-ancing decisions are based on the system state, whereas in the static approach, the load balancing decision are based on parameters known a priori to the deployment. There exists various different load balancing algorithms for different data center types, depending on their scale, service structure, and many others.

B. Surveyed Systems: Load imbalance metrics have large effect on the accuracy and adaptivity of the load balancing mechanism. In [12], Arzuaga et al. present a new metric for physical server load, that is a function of the VMs residing in the server. Based on this new load imbalance metric, they propose a framework for VM migration. Their solution is able to predict the future system state, and in some cases outperform the current state-of-the-art load balancing systems by over 5%. [12] Beloglazov et al. list four problems related to virtual ma-chine load balancing. The first, higher level problem is to find an optimal solution for the energy-performance trade-off problem. The second problem is raised by the fact, that virtual machine migration induces overhead.

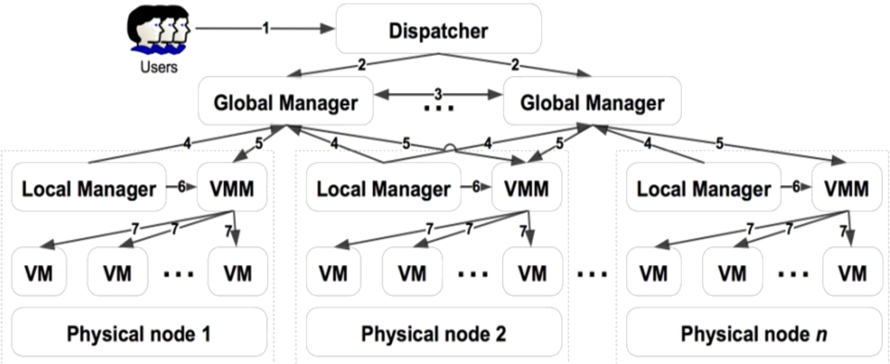

Fig. 2. The system architecture of the system presented in [11]

The problem is to find an efficient way of migrating VM’s between the servers, i.e. when and which virtual machine should be migrated, and where? The third question is how to develop efficient decentralized and scalable algorithms for resource allocation? How to develop a solution by combining multiple different allocation policies with different objectives? [11] Beloglazov et al. [11] then present a decentralized solu-tion for the problem and evaluate it with simulations using CloudSim [13] data center simulator. Their software system architecture is divided into three levels: dispatcher, global and local managers. The dispatcher distributes the requests between the global managers. The global managers, each attached to a set of local managers, distributes their own local manager data between other global managers, and apply heuristics to load-balance the nodes. The local managers reside inside each of the physical server nodes, and are responsible for the continuous monitoring of the resource utilization in the node. [11] The orchestration of the server migration requires commu-nication between the servers, which directly relates to the problem presented in the previous section. In [14], Sonnek et al. proposes a decentralized, communication efficient, mi-gration technique to allocate virtual machines on the available physical resources. Their technique is based on monitoring network affinity between VMs, and distributing the VM-relocation algorithm. In their distributed algorithm, physical servers independently negotiate VM relocations on the basis of local information. [14]Wood et al., in [15], present Sandpiper, a migration automa-tion and tracking tool for virtualized data centers. Sandpiper automatically detects the overloaded virtual machines, deter-mines a new mapping of physical to virtual resources and initiates the migrations. They propose two different strategies for monitoring the server: a blackbox strategy and a gray-box strategy. In black-box strategy, the state of the servers, on which the load balancing decisions are made, must com-pletely be observed from outside the system, without OS-level support. Their experiments show that utilizing OS-level infor-mation, i.e. gray-box strategy, can improve the responsiveness of the system

CONCLUSION

This paper reviewed different load balancing mechanisms and presents some of the problems at two parts of the data center: network traffic and server node load balancing. The high level goal of load balancing in both network traffic and server load, is the efficient resource utilization while still providing the required performance guarantees. However, the subgoal for these seems to be different. In the network traffic load balancing, the goal is to be able to efficiently scale the system, where as in the server load balancing, the discussion is more about energy efficiency. Based on the

DOI 10.17148/IJARCCE.2018.71020

literature, the network traffic load balancing is currently the largest bottleneck in the data center scaling. Network traffic in data centers is growing fast and even with high-end switching fabric, efficient scaling of data center networks is nontrivial. Nearly all the solutions are able to answer to the performance requirements in the optimal system conditions, but either require specialized hardware or perform poorly under non-optimal system state.Major proportion of today's data center energy consumption occur from the server nodes. Based on the surveyed papers, the goal for the server load balancing to minimize the server usage to lower the energy consumption. Efficient virtual machine migration seems to be the most promising way to achieve those goals, but it brings several challenges with it.

REFERENCES

[1]. H. Sutter, “The free lunch is over: A fundamental turn toward concur-rency in software,” Dr. Dobb’s Journal, vol. 30, no. 3, 2005.

[2]. G. M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities,” in Proceedings of the April 18-20, 1967, Spring Joint Computer Conference, ser. AFIPS ’67 (Spring). New York, NY, USA: ACM, 1967, pp. 483–485.

[3]. P. M. Mell and T. Grance, “Sp 800-145. the nist definition of cloud computing,” Gaithersburg, MD, United States, Tech. Rep., 2011.

[4]. M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” in Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 19–19.

[5]. L. A. Barroso, J. Clidaras, and U. Hlzle, “The datacenter as a computer: An introduction to the design of warehouse-scale machines, second edition,” Synthesis Lectures on Computer Architecture, vol. 8, no. 3, pp. 1–154, 2013.

[6]. E. Arzuaga and D. R. Kaeli, “Quantifying load imbalance on virtualized enterprise servers,” in Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering, ser. WOSP/SIPEW ’10. New York, NY, USA: ACM, 2010, pp. 235–242.

[7]. J. Sonnek, J. Greensky, R. Reutiman, and A. Chandra, “Starling: Minimizing communication overhead in virtualized computing platforms using decentralized affinity-aware migration,” in Proceedings of the 2010 39th International Conference on Parallel Processing, ser. ICPP ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 228–237.

[8]. www.google.com

[9]. www.w3school.com (it helps HTML, AJAX,JavaScript & etc)

[10]. J. Dorn and T. Naz, “Integration of Job portals by Meta-search,” in Proc. 3rd International Conf. onInteroperability for Enterprise Software and Applications, Funchal, Portugal, 2007, pp. 401-412.

DOI 10.17148/IJARCCE.2018.71020