Efficient Dynamic Load-Balance Flow Scheduling in cloud for Big Data Centers.

Snehal Ghodake[1], Sulochana Sonkamble[2]

[1]*Department of Computer Engineering*
Rajarshi Shahu School of Engineering and Research
Pune ,Maharashtra, India.

[2]*Dr,Department of Computer Engineering*
Rajarshi Shahu School of Engineering and Research
Pune ,Maharashtra, India.

---***---

Abstract - *Big data centers in cloud, large amount of data needs to be transferred frequently among thousands of interconnected servers. In which load balancing and flow scheduling is a challenging issue. The Open Flow is a auspicious solution to balance data flows in big data center network through programmatic traffic controller. Existing solution can able to statically set up routes only at initialization stage of data transmissions, which experiences from dynamical flow distribution and network changing state so it results in decrease system performance. In this paper, we will propose a new dynamical load-balanced scheduling (DLBS) approach for increase the network throughput to dynamically balance workload. This approach formulate the DLBS problem, and then develop a set of improved heuristic scheduling algorithms for the two typical Open Flow network models, which balance data flows time slot by time slot. Experimental results demonstrate that our DLBS approach signicantly outperforms other load-balanced scheduling algorithms Round Robin and LOBUS; and the higher imbalance degree data flows in data centers exhibit, the more improvement our DLBS approach will bring to the data centers.*

***Key Words*: Big data centre, Open Flow, Dynamical load balanced scheduling.**

1.INTRODUCTION

In computing, the distribution of workloads across multiple computing resources, such as com-puters, a computer cluster, network links, central processing units, or disk drives. Load balancing aims to optimize resource use minimize response time, maximize throughput and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may in-crease reliability and availability through redundancy. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process.

Traditional hardware-based load balancing techniques cannot be widely used due to the high cost and the deficiency in programmable ability. Therefore, more and more researchers pay more attention on software-defined networking(SDN)techniques (e.g.,OpenFlow) that can improve transmission capacity of data centers through programmable load balanced flow control [1].
Many schemes have been proposed for load-balanced flow scheduling in OpenFlow based networks [8] [10]. They focus on the initial route selection only before the flow transmission. Network states and work load, however, often dynamically change because during a data transmission, a part of links may become unavailable, new data flows can arrive and some existing data flows have completed . As a result, the existing proposals cannot meet the needs of dynamical load balance during data migrations. On the other hand, as data center networks become more large and more complex, the time that these proposals need for the initial path selection will increase tremendously [4] [5].

Motivated by the above observations, we in this paper propose a efficient novel dynamical load-balanced scheduling (DLBS) approach to maximize the network throughput through dynamically balancing data flows. Aiming at the most popular two OpenFlow network models, three-layer non-blocking fully populated network (FPN) and three-layer fat-tree network (FTN), we will propose and develop different scheduling algorithms, which quantitatively analyze the imbalance degree of data center networks at the beginning of each time slot and then schedule unbalanced data flows once a load imbalance happens. Traditional OpenFlow framework has a few limitations [10], for example, it does not support hash-based routing [9] to spread traffic over multiple paths. So, we rely more on flexible load balancing algorithms to fit different imbalance situations. To the best of our knowledge, this is the first work to switch flows in the midway of data transmissions in OpenFlow networks. The main contributions of this paper are summarized as follows.

- We identify a new flow scheduling problem in big data centers in clouds, i.e., dynamical load-balanced scheduling (DLBS), and formulate the DLBS problem. The objective is to optimize network throughput on condition that load balancing is guaranteed on all links during every time slot.

- We will propose a trigger mechanism for dynamical data flow scheduling. We firstly propose a factor δ(t) to capture the load imbalance degree of data center networks, and then define the link scheduling trigger threshold δ*. δ(t) is calculated slot by slot, and the OpenFlow controller initiates our DLBS scheduling algorithms once δ(t)≥δ*.

- We will propose a set of heuristic scheduling algorithms to address the DLBS problem. They are implemented in two representative OpenFlow architectures: FPN and FTN. These algorithms dynamically migrate the flows which occupy the largest amount of bandwidth on the most congested link to the lightest links.

- We will implement a system to simulate a cloud data center and evaluate our DLBS approach through comparing the DLBS with other classical methods.

2. RELATED WORK

Flow scheduling has been widely studied over the past years. In recent years, related researches paid more attention on OpenFlow-based and cloud data center oriented resource Scheduling

1. Flow Scheduling :
   Existing researches on flow scheduling can be classified as static and dynamical schemes. Static load balancing schemes distribute the traffic mainly based on a fixed set of rules according to characteristics of the input traffic, which cannot feedback real-time information of traffic and network states on links
2. OpenFlow-Based and Data Center Oriented: OpenFlow is a leading software-defined networking architecture, which allows for quick experimenting and optimizing switching/routing policies. Handigol et al. [8] proposed the OpenFlow-based LOBUS algorithm to balance the load, which applies greedy selection strategy to pick the (server, path) pair that yields the least total response time at every request.

3. PROBLEM STATEMENT

Big data centers in cloud large amount of data needs to be transferred frequently among thousands of interconnected servers, to balance dynamically data flows in a data center and maximizing the network throughput is a key and challenging issue.

4. EXISTING SYSTEM

In this paper we can use the load balancing method. Load balancing means that all resources in a system are equally shared by all tasks in some measures. It can be mathematically described by means of a performance criterion. In general, the objective of load balancing is to optimize resource utilization, maximize throughput, minimize response time, and avoid overload of any single resource. Network load balancing aims at evenly distributing traffic across multiple links without using complex routing protocols.

This ability balances network sessions over multiple connections in order to spread out the amount of bandwidth used by each data flow, thus increasing the total amount of available bandwidth.

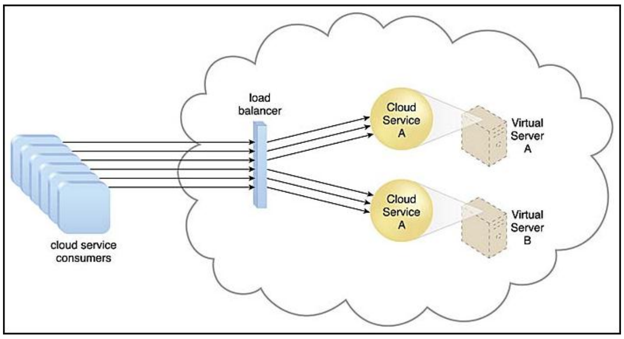

Fig -1: Workload Distribution Architecture

5. PROPOSED SYSTEM

In this paper propose a novel dynamical load-balanced scheduling (DLBS) approach to maximize the network throughput through dynamically balancing data flows. Traditional OpenFlow framework has a few limitations [10], for example, it does not support hash-based routing [9] to spread traffic over multiple paths. So, we rely more on flexible load balancing algorithms to fit different imbalance situations. To the best of our knowledge, this is the first work to switch flows in the midway of data transmissions in OpenFlow networks.

We propose a set of heuristic scheduling algorithms to address the DLBS problem. They are implemented in two representative OpenFlow architectures: FPN and FTN. These algorithms dynamically migrate the flows which occupy the largest amount of bandwidth on the most congested link to the lightest links.

A. Architecture

Network load balancing aims at evenly distributing traffic across multiple links without using complex routing protocols. This ability balances network sessions over multiple connections in order to spread out the amount of bandwidth used by each data flow, thus increasing the total amount of available bandwidth. Network load balancing is achieved through various scheduling schemes, falling into static scheduling and dynamical scheduling. In the static load balancing scheduling, managers set up and maintain a set of rules in advance, using Round Robin, Ratio-based and Priority-based or other algorithms.

B. Mathematical model

We model a cloud data center (CDC) network as an undirected graph G = (V, E). V is the union of the switch set (V_S) and the host set (V_H) such that $V=V_S \cup V_H$. Note that switches are interconnected in a layered structure and each host $h \in V_H$ is connected with a home switch that directly connects the h with the switch network. E is the union of E_S (the set of links among switches) and E_H (the set of links between any host and its home switch) such that $E=E_S \cup E_H$

C. Software Requirement Specification

HARDWARE REQUIRMENTS:

- System :Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Sony.
- Ram : 512 Mb

Software Requirements:

- Operating system : Windows 7 .
- Coding Language : ASP. Net with C
- Data Base : SQL Server 2005.

D. ALGORITHM

Table -1 Notation

Notations	*Description*
i, j	switches i and j, respectively
K	the number of concurrent data flows in T slots
f_k	the kth data flow f_k=(s_k, d_k) ($1 \le k \le K$), where s_k and d_k are source and destination hosts of f_k, respectively
$t^{f_k}(t)$	traffic transmitted from s_k to d_k for f_k during a slot t
$t^{f_k}_{i,j}(t)$	traffic from i to j for f_k during a slot t
$B_{i,j}$	Capacity of the link $\ell_{i,j}$ between i and j
N^i	the set of neighboring switches of i
$\lambda_{i,j}(t)$	bandwidth utilization ratio of $\ell_{i,j}$ during a slot t

Algorithm 1: Improved DLBS-FPN

Input: S2SPT and ART tables, δ^*
Output: load-balanced scheduling
update $\delta(t)$ using the formula (8) in each time slot t;
MAX=-1;
Temp= $\sum_{k=1}^{K} t^{f_k}(t)$;
while ($\delta(t) \ge \delta^*$) OR (MAX>Temp) **do**
  $\ell_s \leftarrow$ the busiest link $\ell_{i,j}$;
  $f_s \leftarrow$ the biggest flow on ℓ_s;
  find out substitute sub-paths $P_{i,j}=\{p_1, p_2, \ldots, p_m\}$ for f_s through S2SPT;
  select the lightest sub-path $p_k \in P_{i,j}$ in terms of ART;
  schedule f_s to p_k;
  update $\delta(t)$;
  MAX=Temp;
  Temp= $\sum_{k=1}^{K} t^{f_k}(t)$;
end while

Algorithm 2: Improved DLBS-FPN

Input: S2SPT and ART tables, δ^*
Output: load-balanced scheduling
update $\delta(t)$ using the formula (6) in each time slot t;
MAX=-1;
Temp= $\sum_{k=1}^{K} t^{f_k}(t)$;
while ($\delta(t) \ge \delta^*$ and $\Sigma(t) \ne \Phi$) OR (MAX>Temp) **do**
  $f_s \leftarrow$ the flow that covers the largest subset of $\Sigma(t)$;
  $P_s \leftarrow \{p_1, p_2, \ldots, p_m\}$;
  $p_s \leftarrow p_{best} \in P_s$;
  schedule f_s to p_s;
  update $\delta(t)$;
  MAX=Temp;
  Temp= $\sum_{k=1}^{K} t^{f_k}(t)$;
end while

6. EXPECTED RESULTS

DLBS exhibits a higher bandwidth utilization rate than LOBUS and RR in three transmission patterns during the initial stage, and delivers more transmission load before corresponding critical time. The goal is DLBS can efficiently balance the global load so that it significantly improves throughput, transmission delay and bandwidth utilization rate especially under non-uniform network transmission patterns.

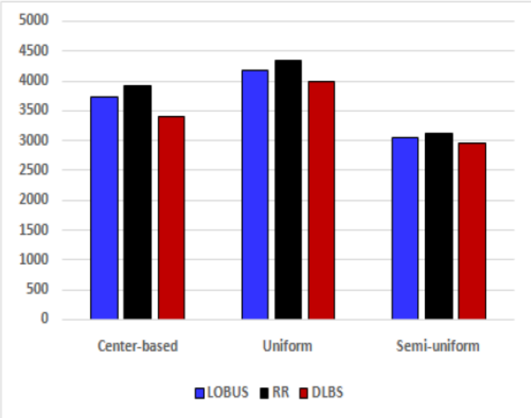

Fig -2: Average end-to-end delay in the three flow scheduling schemes

7. CONCLUSIONS

With the proposed system presented the load-balanced scheduling problem through balancing transmission traffic dynamically and globally in cloud data centers. Aiming at two typical OpenFlow architectures: FPN and FTN. Our algorithms can globally balance transmission traffic in the whole network by means of evaluating link, path and network bandwidth utilization ratio.

ACKNOWLEDGEMENT

I express true sense of gratitude towards my project guide Dr. Sulochana Sonkamble, of computer department for her in valuable co-operation and guidance that he gave me throughout my research, for inspiring me and availing me all the lab facilities, which made this research work very advantageous and easy. I would also like to express my appreciation and thanks to our HOD Prof. R.H.Kulkarni and Director Dr. D.S. Bormane and all my friends who knowingly or unknowingly have assisted me throughout my hard work.

REFERENCES

[1] Z.Z.Cao, M.Kodialam and T.V.Lakshman. Joint Static and Dynamic Traffic Scheduling in Data Center Networks. in Proceedings of IEEE INFOCOM 2014, pp.2445-2553.
[2] J.Lu, D.Li, Bias Correction in Small Sample from Big Data, IEEE Transactions on Knowledge and Data Engineering, Vol.25, No.11, 2013,pp.2658-2663.
[3] A.G.Erdman, D.F.Keefe, R. Schiestl, Grand Challenge: Applying Regulatory Science and Big Data to Improve Medical Device Innovation, IEEE Transactions on Biomedical Engineering, 60(3) (2013) 700-706.
[4] X.Han, J.Li, D.Yang et al., Efficient Skyline Computation on Big Data, IEEE Transactions on Knowledge and Data Engineering, Vol.25, No.11, 2013, pp.2521-2535.
[5] A.G.Erdman, D.F.Keefe, R.Schiestl, Grand Challenge: Applying Regulatory Science and Big Data to Improve Medical Device Innovation, IEEE Transactions on Biomedical Engineering, Vol.60, No.3, 2013, pp.700-706.
[6] K.Greene, TR10: Software-Defined Networking, MIT Technology Review,Retrieved Oct. 7, 2011.
[7] R.M.Pathan and J.Jonsson. Load regulating algorithm for staticpriority task scheduling on multiprocessors. Proc. of 2010 IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2010, pp.1-12.
[8] N.Handigol, S.Seetharaman, M.Flajslik, N.McKeown, and R.Johari, Plugn-Serve: Load-balancing web traffic using OpenFlow, Demo at ACM SIGCOMM, Aug. 2009.
[9] Schlansker, Y.Turner, J.Tourrilhes, and A.Karp, Ensemble Routing for Datacenter Networks, In ACM ANCS, La Jolla, CA, 2010.
[10] Wang, D.Butnariu, and J. Rexford, OpenFlow-Based Server Load Balancing Gone Wild, in: Proceedings of Workshop on HotICE, Mar.2011.
[11] N.Gude, T.Koponen, Justin Pettit et al., NOX: Towards an Operating system for Networks, SIGCOMM Comput. Commun. Rev., Vol.38, July 2008, pp. 105-110.
[12] N.McKeown, T.Anderson, H.Balakrishnan, G.Parulkar, L.Peterson,J.Rexford, S.Shenker, and J.Turner, OpenFlow: Enabling Innovation in Campus Networks, SIGCOMM Comput. Commun. Rev., 2008.

BIOGRAPHIES

Snehal Ghodake
Department of Computer Engineering
Rajarshi Shahu School of Engineering and Research
Pune ,Maharashtra, India.