Future Generation Computer Systems 68 (2017) 183–190

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

LBBSRT: An efficient SDN load balancing scheme based on server response time

Hong Zhong, Yaming Fang, Jie Cui [*]

School of Computer Science and Technology, Anhui University, Hefei, 230039, China

ARTICLE INFO

Article history:
Received 22 May 2016
Received in revised form
5 September 2016
Accepted 1 October 2016
Available online 7 October 2016

Keywords:
SDN
OpenFlow
Load balancing
Server response time

ABSTRACT

The response time is the most important factor determining user experiences in the service provision model involving server clusters. However, traditional server cluster load balancing scheme are limited by the hardware conditions, and cannot completely exploit the server response times for load balancing. In order to effectively resolve the traditional load balancing schemes, we propose a load balancing scheme based on server response times by using the advantage of SDN flexibility, named LBBSRT. Using the real-time response time of each server measured by the controller for load balancing, we process user requests by obtaining an evenly balanced server loads. Simulation experiments show that our scheme exhibits a better load balancing effect and process requests with a minimum average server response times. In addition, our scheme is easy to implement, and exhibits good scalability and low cost characteristics.

1. Introduction

Achieving optimum load balancing is of significant importance whilst combating network overhead issues in any distributed processing architectures. Service availability is paramount in measuring end user satisfaction [1], which is heavily impacted by the level of achievable load balancing among the process clusters. In general, a well-balanced load in the network helps to optimize the utilization of the available resource by the ways of maximizing the throughput, minimizing the response time, and avoiding overloading resources in the network [2]. For the purposes of alleviating heavy-traffic network flux and reducing the risk of single server becoming the main overhead contributor, many datacentres adopt dedicated hardware resources to achieve load balancing whilst supporting a large number of users [3]. However, the increasing costs and technical complications in the deployment of such hardware systems often require human intervention to ensure consistent functioning of such strategies [4].

Software-Defined networking (SDN) is one of the notable forms of computer networking [5,6], facilitating a simple and conveniently maneuverable network flow control method requiring minimal investment costs whilst availing maximum benefits for a massive number of users. SDN controls the data transportation by deploying the network switches as a software implementation, whereby a flow table lookup operation will be carried out whenever a data flow arrives at the switches. Flow tables [7] ([Header: Counters: Actions]) are widely used in SDN. The headers and counters of the flow table are updated accordingly whenever actions relevant to flow changes are imposed. During this update process, the header information is usually recorded onto the database and the OpenFlow switches process the data flow in accordance with the header records. Based on the SDN model with a centralized controller, an OpenFlow switch [8] is designed with different rules to control the network traffic using the header records. Balancing the network load at the software tier is now practically realizable using the SDN facilitated flow control system. To this end, Handigol [9] proposed plug, a load balancing model based on SDN. Based on the Openflow environment, Kaur [10] achieved network load balancing using polling algorithm. Further, Zhang [11] achieved the minimum number of connections in the network using the polling algorithm of load balancing under the SDN framework. Shang [12] incorporated a middlebox based on the SDN architecture to achieve load balancing by collecting the server information. Despite the existing implementations of SDN to resolve high cost and poor flexibility issues in achieving effective load balancing, notable drawbacks are still prevalent in the aforementioned schemes. To add a few, Kaur and Zhang [10,11] applied traditional load balancing algorithms to the SDN architecture, and so the two schemes cannot effectively reduce the server response time. Though Shang [12] can effectively reduce the server response time, this scheme relies on the server information which increases the complexities of the server architecture.

[*] Corresponding author.
E-mail address: cuijie@mail.ustc.edu.cn (J. Cui).

http://dx.doi.org/10.1016/j.future.2016.10.001
0167-739X/© 2016 Published by Elsevier B.V.