Research on Dynamic Migration Technology of OpenFlow Switch

Li-Jun MAO\textsuperscript{a} and Yan ZHANG\textsuperscript{b}

Intelligent Science & Information Engineering College, Xi'an Peihua University, Xi'an, China
\textsuperscript{a}35909250@qq.com, \textsuperscript{b}410325568@qq.com

Keywords: Switch migration; Openflow; Load balancing.

Abstract. To implement load balance on controllers that are connected to each other in the OpenFlow control plane, we propose a dynamic OpenFlow switch migration algorithm based on attractor selection to solve the performance bottleneck experienced by a controller when the flow fluctuates on OpenFlow switches. Firstly, we model the problem of the dynamic OpenFlow switch migration. Then, we redefine the parameters of attractor selection based on the switch migration model and propose a dynamic OpenFlow switch migration algorithm based on the attractor selection. Finally, we use the proposed algorithm to determine the destination to the migrated switches to balance the load on the controllers. Simulation results show that the proposed algorithm can guarantee that the load on the controllers is balanced with a low cost.

Introduction

The Internet has become an important infrastructure for the modern information society. It provides a common platform for information exchange, resource sharing and network computing, and has a profound impact on people's work, study, consumption, and social activities. However, with the continuous expansion of the Internet and the continuous expansion of service types, the popularity of cloud computing applications has become more prominent. The drawbacks of traditional network architectures have become more prominent. The network architecture based on TCP/IP technology has been exposed more and more. The insurmountable problems, including the scalability, reliability, dynamics, security, and flexibility of network management, have caused a serious mismatch between network functions and market demands.

In order to achieve load balancing on interconnected controllers in the OpenFlow control plane, this paper proposes an OpenFlow switch dynamic migration algorithm based on attractor selection. Software defined network (SDN) has the separate features of network control and transmission, which can simplify network control, rapidly develop and deploy new applications, and has been widely used in IP networks, optical networks, and wireless networks [1]. The OpenFlow protocol is a protocol commonly used in SDN technology [2]. It can send a flow table to an OpenFlow switch through a centralized controller to realize network control and transmission separation. However, when large-scale deployment of OpenFlow switches to build complex SDN networks, centralized controllers can become a bottleneck in network performance [3]. In order to solve the performance bottleneck problem of OpenFlow controller when SDN is deployed on a large scale, Onix is proposed as a logically centralized and physically distributed control plane [4]. In Onix, each controller is deployed independently and connected to each other to form a control plane. Controllers share information about the entire network of SDN by sharing information with each other. Each controller is connected to several OpenFlow switches. The controller sends a flow table to its connected OpenFlow switch according to the SDN network information to control and optimize network traffic. On the basis of Onix, [5] proposed an elastic control plane architecture, determined the number and location of controllers deployed in SDN through static programming, and built an SDN control plane; and then dynamically migrated OpenFlow switches according to the load on the OpenFlow switch.

According to the characteristics of the switch migration and the attractor algorithm model, the switch migration problem is modeled, the physical meaning of each parameter in the attractor algorithm is determined, and a dynamic migration algorithm is proposed. According to the solution
of the attractor algorithm, which switches are migrated from the original controller. Go to the target controller to balance the load on the controller.

Switch Migration Model

The SDN network is defined as \{ S, C \}, where S is a set of switches, and the number of switches included is nS; C is a set of controllers, and the number of controllers included is nC, and s represents any switch in the switch set. Cj represents any controller in the controller set. The connection relationship between any switch and controller is defined as bij, taking 0 or 1, 0 means that switch si is not connected to controller cj, and 1 means that switch si is connected to controller cj. For any controller cj, the total number of switches connected to it is ncj and:

\[ ncj = \sum_{S_i \in S, C_j \in C} b_{ij} \]  

(1)

Since the Open Flow switch sends the Packetin packet to the controller only when the flow table is newly established, the controller is required to send the flow table to it. Therefore, the load on any controller cj is defined as the total number of Packetin packets sent to that controller by all Open Flow switches connected to it:

\[ L(C_j) = \sum_{S_i \in S} P(S_i) \]  

(2)

For the dynamic migration problem of OpenFlow switches, the goal is to ensure load balancing on the controller after dynamic migration of the switch, which is described by the standard variance of the load on the controller:

\[ \min \sqrt{\sum_{j=1}^{nC}(L(C_j) - \frac{1}{nC} \sum_{j=1}^{nC} L(C_j))^2} \]  

(3)

s. t. \[ \min(\max(\text{dij})) \leq TD \]  

(4)

\[ e < TE \]  

(5)

\[ \forall i, \exists j, \ b_{ij} = 1, \ b_{ij'} = 0, \ \text{if } j' \neq j \]  

(6)

\[ \sum_{j=1}^{nC} ncj = nS \]  

(7)

In order to obtain the dynamic migration method of OpenFlow switches, when solving the switch migration problem, delay constraints, cost constraints, binary constraints, numerical constraints, etc. need to be considered.

OpenFlow Switch Migration Algorithm Based on Attractor Selection

The attractor selection model describes the kinetic model of E. coli growth process, as shown in equations (8) and (9), where xi represents the level of gene expression, representing the different evolutionary directions of E. coli, ensuring that E. coli can be lactose The highest concentration of movement, in which E. coli produces the fastest protein and the fastest growth rate.

\[ \frac{dx_i}{dt} = \frac{v_g}{1+e^{-\mu(\sum_{j} w_{ij} x_j - \theta_i)}} - x_i v_g + \eta_i \]  

(8)

\[ v_g \propto \min\{y_1, y_2, \ldots, y_m\} \]  

(9)

Among them, vg indicates the growth rate of E. coli; Wij indicates the influence coefficient between different gene expression levels; \( \theta \) is the threshold of gene expression level; \( \eta_i \) is the random noise of \( x_i \), which ensures the gene expression level has certain activity; \( y_m \) indicates protein production The amount.
The Open Flow switch dynamic migration algorithm based on attractor optimization ensures that the attractor can migrate to the optimal solution in the dynamic migration algorithm, so the growth rate \( v_g \) in the attractor selection model is defined as the parameter positively related to the optimization target \( r \), i.e., migration. The more OpenFlow switches, the more balanced the load on the controller, and the smaller the value of the optimization target.

\[
v_g = \frac{1}{1 + e^{\delta (\varphi - \gamma)}}
\]  (10)

Equation (10) shows that in a large-scale deployment of SDN, the smaller the standard deviation of the load on the controller, the faster the growth rate in the attractor optimization algorithm, and the more stable the attractor selection model, the more the number of Open Flow switches being migrated less.

In equation (10), \( r \) is the standard deviation of the controller load after the last switch migration operation is completed; \( \delta, \gamma, \varphi \) are constants, and the effect of \( r \) on \( v_g \) is adjusted.

The above determines the connection relationship between any pair of switches and controllers. On this basis, the dynamic migration algorithm flow of OpenFlow switches based on attractor optimization is given:

1. At time \( t_0 \), according to the connection relationship \( b_{ij} \) of switch \( i \) and controller \( j \), the gene expression level \( x_{b_{ij}} \) in the attractor optimization model is generated. If switch \( i \) is connected to controller \( j \), \( x_{b_{ij}} \) is larger; if switch \( i \) and controller \( j \) are not connected, \( x_{b_{ij}} \) is smaller.

2. For all controllers, the standard deviation of the load on the controller according to equation (3); for any \( i \) and \( j \), the calculation \( x_{b_{ij}} \).

3. For any \( i \) and \( j \), in order to improve the robustness and stability of the algorithm, the probability selection method is used to determine the connection relationship between switch \( i \) and controller \( j \), that is, the probability of \( b_{ij} = 1 \) is \( p(c_{ij}) \). The probability selection method is:

\[
p(c_{ij}) = \frac{x_{c_{ij}}}{\sum_{i,j} x_{c_{ij}}} \]  (11)

4. For any given \( i \) and \( j \), compare the connection relationship between all switches and controllers at time \( t \) \( b_{ij} \) and the connection relationship between all switches and controllers at time \( t - 1 \), determine the switch to be migrated and the destination controller to be migrated.

5. Record the migration cost at time \( t \), and determine the interval for the next time the Open Flow switch is dynamically migrated according to formula (11), to ensure that the Open Flow switch migration action is triggered not too frequently, resulting in excessive migration cost; The dynamic migration action of the Flow switch cannot be triggered in time, which makes the algorithm unresponsive to the load change of the controller, causing the controller to become a performance bottleneck.

Simulation and Performance Analysis

The typical 34-node network topology of the Internet 2 project is used as the simulation environment. The static controller deployment algorithm is used to determine the number of controller deployments and deployment locations, and the performance of the proposed algorithm is verified. The discrete event-driven simulation platform written in C language verifies the controller static deployment and switch migration algorithm by simulating the load on the switch and the delay between the switches. As shown in Fig. 1, the simulation environment includes 34 OpenFlow switches and 4 controllers. The delay threshold is defined as 1200 ms, and the controllers are deployed at nodes 7, 14, 24, and 33. During the simulation process, each independent switch changes the Packetin packet sent to its connected controller once every 10 time units, and the probability of change obeys an even distribution between (5, 10).

In the simulation process, when determining the relevant parameters in equations (10), (11) and
(12), in order to avoid a strong influence on the gene expression level when strong correlation between bij and bsc, \( \eta_{bij} \) is set to mean during the simulation. Gaussian noise with a variance of 0.5 and \( \theta_{bij} = 0.5 \); in order to amplify the weak correlation between bij and bsc, the effect of the system changes, \( \mu = 10 \); in order to amplify the gene expression level on the growth rate of E. coli impact, \( \delta = 10 \), \( \gamma = 1 \), \( \varphi = 0 \).

Simulation experiments show that the average running time of the proposed AS-DSM algorithm is the shortest. This is because the K-DSM, DHA-DSM, CT-DSM, and GD-DSM algorithms convert the dynamic migration problem of the switch into a static optimization problem, every time on the switch. The optimization problem needs to be initialized when the load changes. The AS-DSM algorithm mentioned in this paper only needs to provide simple feedback when the load on the switch changes, and the solution of the switch migration can be obtained. Therefore, as time progresses, the average time of the proposed algorithm is shorter, while the average running time of other algorithms is longer.

![Network Topology](image)

**Fig. 1. The Network Topology of the Simulation.**

**Conclusion**

In this paper, the attractor algorithm can solve the characteristics of dynamic selection problem by simple feedback, analyze the dynamic migration problem of OpenFlow switch, and then model the attractor algorithm, and solve the dynamic migration problem of OpenFlow switch under the condition that the load of OpenFlow switch cannot be accurately described. The simulation results show that the proposed algorithm has better load balancing effect and less migration cost. The OpenFlow switch migration model established by the proposed algorithm provides the basis for solving the OpenFlow switch migration by using the existing selection optimization algorithm; the attractor selection model used can adapt to the environment with dynamic changes and cannot be described by analytic, and provides solutions for solving similar problems.

**Acknowledgment**

This work was supported by the 2019 Xi'an Peihua University Key funding project "Application Research of big data visualization technology in university educational administration system" (No: PHKT19002).

**References**

[1] Wang L.X., H. Qu, J.H. Zhao, et al. A strategy of controller placement in software defined networks using binary particle swarm optimization [J]. Journal of Xi'an Jiaotong University, 2015, 49 (6): 67-71.
[2] Cheng, G.Z., H. Chen, H.C. Hu, et al. Toward a scalable SDN control mechanism via switch migration [J]. China Communications, 2017, 14 (1): 111-123.

[3] Zhao, Jun, Bao Congxiao and Li Xing. Design of Overlay Network Router Based On Openflow Protocol [J]. Journal of Tsinghua University (Natural Science Edition), 2018, 2.

[4] Zhao Jihong, Zhang Bin, Wang Li, Qubi and Zheng Lang. Dynamic Switch Migration Algorithms Based On Q-Learning in SDN. Television Technology [J]. 2016, 6.

[5] Zhang Yanzhi, Ye Xiaojing and Liang Gang. Based On Openflow Software Definition Network Virtualization Scheme. Computer Application Research [J], 2017, 11.