FlexNet: A Optical Switching Architecture for Optical Data Center Networks

Peng Li, Xiaoshan Yu, Huaxi Gu, Yunfeng Lu

Introduction

Past few years have witnessed the rapid development of Optical Circuit Switches (OCSes), which support high and dynamic bandwidth with low cost in the data center network. As a result, some optical network architectures based on OCSes, such as OSA, Mordia and Rotornet, have been proposed one after another. Rotornet, as a typical optical network architecture, requires all switches to rotate independently through a fixed set of configurations to provide uniform bandwidth between across all endpoints. The configuration algorithm in RotorNet is extremely insensitive to traffic, resulting in poor performance under non-uniform traffic. A mismatch between the network topology and traffic pattern can result in wasted bandwidth on links with minimal or no traffic.

In this paper, an optical switching architecture, FlexNet, is proposed to improve the utilization of optical link resources. In FlexNet, a centralized controller is added to estimate traffic demands, and a more flexible configuration algorithm is designed to change the network topology efficiently, which can also be well applied to Rotornet. We evaluated the latency and throughput performance of FlexNet, Rotornet, and Mordia using the OMNET ++ simulator. The experimental results show that FlexNet has better performance than Rotornet and Mordia in different traffic patterns.

System Design

Figure 1 illustrates the general FlexNet architecture. For example, when we set the rates of electric links and optical links to 25Gbps and 100Gbps respectively, and assume that n MEMS and k ToRs are used, then the number of servers in each ToR should be 4n to ensure the consistency of uplink and downlink bandwidth.

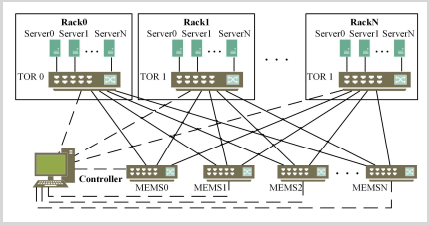

Figure 1

A. MEMS Configuration

MEMS mainly includes transmission period and configuration period. In a complete transmission period, when we set n and k are 4 and 13 respectively, the configuration of the MEMS should be updated by twelve fixed connection modes, such as Figure 2. Each transmission period is evenly divided into three parts, and each part is named td. There are 12 td in four MEMS, corresponding to 12 optical switch configurations. These optical switch configurations are distributed on MEMS0-MEMS3, so that ToR0 can be connected with ToR1-ToR12 in different time slots, thus forming a full interconnection in a period of time. The connection principle of other ToRs is the same as ToR0, such as Figure 2.

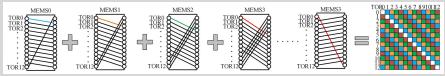

Figure 2

B. Controller

The controller mainly has three processes, including traffic information collection, computing topology, and sending configuration, such as Figure 3.

1) Traffic Collection: In practice, we collect the buffered information sizes of all sockets by netstat and then sum them based on the destination rack. On each node, a user-space management daemon uses netstat to read socket statistics and actively sends them to the controller. The statistics report what quantity traffic is buffered for every destination rack. When the controller receives the statistics report, it integrates the socket statistics into a buffer matrix, such as Figure 3.

2) Computing Topology: When we set n and k are 4 and 13 respectively. Twelve optical switch configurations buffered traffic corresponds to the blocks in the traffic matrix, such as Figure 2. For example, when traffic is steady and uniform, each MEMS can have three configurations, and the duration of each configuration is one third of transmission period. We set the number of td in a transmission period to λ. And we set the duration and the buffer capacity of the twelve configurations to t1-t12 and buffer0-buffer12 respectively. Its specific calculation is as follows:

$$t_i = 4 \times \lambda \times \frac{\text{buffer}_i}{\sum_{i=0}^{i=1} \text{buffer}_i}$$

The corresponding algorithm is explained in Figure 4.

3) Sending Configuration: When the new topology is computed, the controller will check whether the topology is the same as the last one. If not, MEMS will be informed to make a new port configuration. Otherwise, MEMS will not perform any operation. The purpose of this method is to reduce the delay cost caused by optical switch reconfiguration. After MEMS configures the new topology, the controller will inform ToRs to continue data forwarding, such as Figure 3.

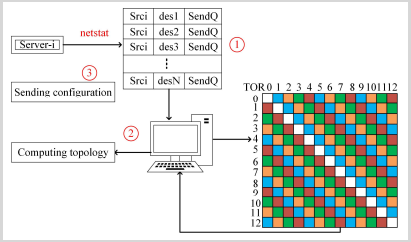

Figure 3

Computing Topology

Input : traffic matrix; the number of td in a transmission period: λ ; Total buffered traffic:buffer;buffered traffic of MEMS connection mode i:buffer_i; the number of MEMS: n; the number of racks:k; the number of server in a rack:m

Output: the number of td in MEMS connection mode i duration:ti

```
Begin  initialize traffic matrix
  for i∈ [1, m*k]
    server i send buffered traffic information to
    controller;
  end for
  λ = (k-1)/n;
  for i∈ [1, λ ]
    buffer += buffer_i;
  end for
  for i∈ [1, λ ]
    ti=buffer_i/buffer;
  end for
End
```

Figure 4

Evaluation

We evaluate the average latency and average throughput performance of FlexNet, Rotornet and Mordia respectively using OMNET++ simulator in three synthetic traffic patterns. We conduct essential packet level simulations with the packet of 1KB size. Besides, we set the same transmission period and configuration period in the simulation of FlexNet, Rotornet and Mordia. The evaluation model is an architecture with 13 racks and 4 MEMS. Each rack has 16 servers and is connected to the ToR switch with 25 Gbps links. Each ToR switch is connected optical switches with 100 Gbps optical fibers. Each ToR switch is connected to the controller with 10Gbps links. Synthetic traffic patterns include:

1) Synthetic Uniform Traffic: The destination rack and destination server of all servers are randomly distributed. In this traffic pattern, the traffic distribution is steady and uniform.

2) Synthetic Neighbor-k Traffic: The destination rack of all servers in rack-i are rack-(i+k), and their destination server is random.

3) Synthetic Hotspot-h Traffic: The destination rack of all servers is rack-h, and destination server is random. The hotspot rack can be arbitrarily assigned.

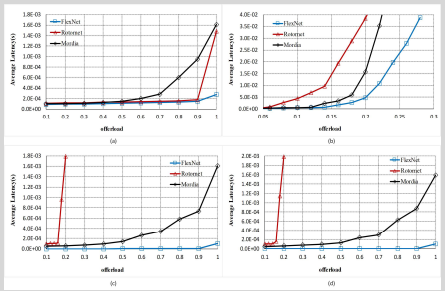

Figure 5 The average latency (a) Synthetic Uniform Traffic (b) Synthetic Hotspot-0 Traffic (c) Synthetic Neighbor-1 Traffic (d) Synthetic Neighbor-2 Traffic

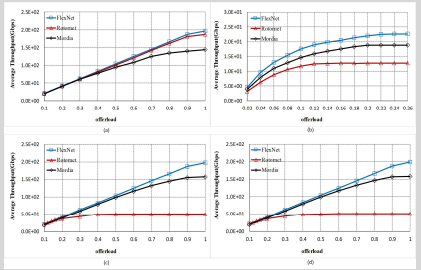

Figure 6. The average throughput (a) Synthetic Uniform Traffic (b) Synthetic Hotspot-0 Traffic (c) Synthetic Neighbor-1 Traffic (d) Synthetic Neighbor-2 Traffic

We analyze and compare the network performance of the FlexNet, Rotornet and Mordia in different synthetic traffic patterns. Figure 5 shows the latency performance. The average throughput performance of different traffic pattern is shown in Figure 6. The simulation results show FlexNet possesses well latency and throughput performance.It is because the controller of FlexNet extracts traffic info, so that the topology can be calculated and adjusted more accurately and timely by the controller, which makes optical links be allocated to ToR pairs with high communications requirement to avoid a number of packets waiting.

Conclusion

We propose a new network architecture based on MEMS named FlexNet, the simulation results show it possesses well latency and throughput performance.

Besides, the FlexNet adds a controller on the basis of Rotornet, which makes the whole network highly flexible that it can adapt its topology to different traffic patterns. The controller can accurately collect the traffic information of the network to allocate optical links reasonably, so that the FlexNet can make full use of costly optical links and make it easier to meet real traffic demands in data center.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2018YFE0202800, and National Natural Science Foundation of China under Grant 61901314, 61634004, and 61934002, the Fundamental Research Funds for the Central Universities under Grant No. JB210110 and XJS200119, the Natural Science Foundation of Shaanxi Province for Distinguished Young Scholars under Grant no 2020JC-26, and The Youth Innovation Team of Shaanxi Universities.