LightFlow: Leveraging Combination of Hash and Wildcard Tables for High Performance Flow Switching in Large Number of Flow Entries

Nobutaka Matsumoto, Michiaki Hayashi, and Itsuro Morita
KDDI R&D Laboratories, Inc.
{nb-matsumoto, mc-hayashi, morita}@kddilabs.jp

1. Introduction

The flow switching capability is attractive in software defined networking (SDN). Today, variety of flow switches are rolled out utilizing existing switching platforms, however, the number of flow entries is effectively limited up to a hundred thousand in many cases due to the limitation of costly devices, such as TCAM. To apply flow switching to a large-scale network, we choose software-based implementations with the commodity hardware which are expected to relief the limitation of the number of flow entries by using low-cost DRAM [1]. However, the software approach derives a new issue: a performance of the packet processing. Several researches have tackled speeding up packet forwarding (e.g. [2]), but scaling to the large number of flow entries is still one of the remained issues.

We demonstrate a software flow switching implementation especially designed for the large-scale flow entry. Based on LightFlow [3] flow table lookup engine, we carefully investigated GPU offloading overhead and enhanced its architecture. We successfully developed the prototype switch node that can keep packet forwarding throughput at 14.6Mpps under large-scale flow environment up to four millions, which is the world fastest record in such an environment as far as we know.

2. Enhanced LightFlow Architecture

Figure 1 shows the enhanced LightFlow architecture. Hash-based exact match table (HMT) is automatically maintained by the lookup result of wildcard-aware flow table (WFT). For each packet, lookup on HMT is invoked at first and lookup on WFT is invoked if no entry is matched in HMT. To accelerate the lookup process, LightFlow utilizes the parallel processing capability of GPU in hash calculation for lookup on HMT and parallelized linear search for lookup on WFT.

In the enhancement of LightFlow, placement of HMT and WFT is optimized. Since one of the major drawbacks using GPU is the overhead of copying data between host memory and GPU memory, the placement of the tables is an important factor to obtain good performance. In lookup procedure on HMT, offloading only hash calculation requires copying hash values from GPU, while offloading whole lookup procedure requires copying packet action and destination in flow entry from GPU. We took the former approach considering the amount of data copy, and that HMT is located on the host memory as shown in Figure 1. On the other hand, since lookup on WFT is carried out by referring WFT entries, WFT is located on GPU memory to avoid data copy of WFT entries between host and GPU. We also found that the suitable placement of the tables depends on many factors such as the bus bandwidth, the operating frequency and the degree of GPU parallelism.

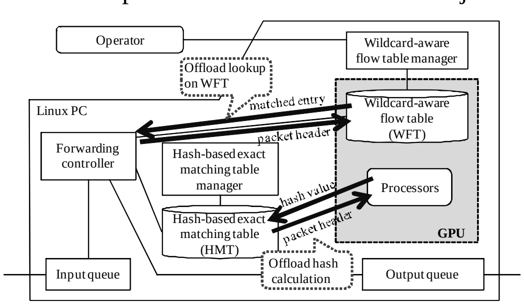

Figure 1. Enhanced LightFlow architecture.

3. Performance

Figure 2 shows the throughputs of enhanced LightFlow switch, conventional GPU-accelerated switch, and Open vSwitch with the packet size of 64Bytes. Thanks to the artifices of applying HMT to most packets while leveraging WFT's maintainability, the throughput of LightFlow switch is stable at 14.6Mpps regardless of the entry size.

4. Conclusions

We proposed LightFlow for high-speed software flow switching which is applicable to large-scale flow entries. Although it is an ongoing work, prototype implementation shows a good performance regarding scalability of the flow entry size. In the poster and demo session, we will demonstrate four million flow switching of LightFlow switch installed in our laboratory.

References

[1] R. Bolla, et al., PRESTO'08, August 2008.
[2] S. Han, et al., SIGCOMM 2010, September 2010.
[3] N. Matsumoto, et al., HPSR2012, July 2012.

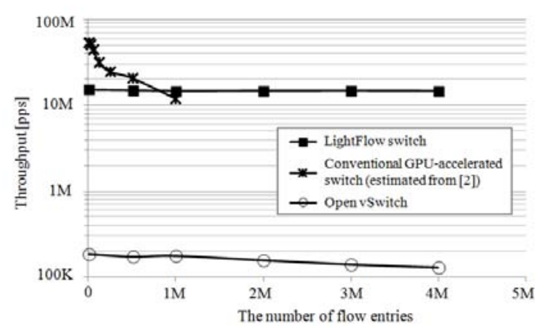

Figure 2. Throughputs as a function of the flow table size.