Enabling Memory-intensive Network Functions on Programmable Switches

Daehyeok Kim[1], Zaoxing Liu[1], Yibo Zhu[2], Changhoon Kim[3], Jeongkeun Lee[3]
Vyas Sekar[1], Srinivasan Seshan[1]
[1]Carnegie Mellon University, [2]Bytedance, [3]Barefoot Networks

Network functions (NFs) are an important component in today's network infrastructures. Until recently, many common NFs were implemented in software that runs on CPUs. However, there is a growing trend to move NFs, such as load balancers [5] and network monitoring [6], onto switch ASICs. The key enabler of this trend is the Protocol Independent Switch Architecture (PISA) [2]. It turns switch ASICs into programmable microprocessors that are highly optimized for (and limited to) packet processing. With similar price and power consumption, these switch ASICs can run NFs at orders of magnitude higher packet processing rates than their general purpose CPU counterparts.

While the computational capabilities of switches have advanced dramatically with the addition of PISA ASICs, switch memory has seen little improvement. On the most popular commodity switch chips today, there is only up to tens of MB of SRAM memory, similar in size to the CPU caches in modern x86 CPUs. Only NFs that can fit their state in this limited memory can be implemented fully on the switch. As a result, the limited memory on current switches has become a major bottleneck in moving NFs, especially memory-intensive NFs, such as full-cone Network Address Translation (NAT) or network monitoring, onto switch hardware.

One possible solution would be to add more memory to switches. However, this is impractical due to *cost*, *flexibility*, and *scalability* concerns. Adding more SRAM, which can support the high bandwidth that switch ASICs needs, is very expensive. Similarly, a DRAM-extensible ASIC that's equipped with DRAM controllers and I/O channels to meet the memory-access bandwidth demand is a very expensive option. Moreover, in this case since DRAM can be used either for one or a few select look-up tables depending on how the DRAM is physically connected to the associated modules within the ASIC chip, it leads to an inflexible usage of DRAM. Neither approach can easily scale the size of memory on demand.

In this talk, we will present an alternative approach in which NFs implemented on a programmable switch can make use of DRAM on servers connected to the network. We call this architecture GEM (Generic External Memory) for programmable switches. This design is driven by the observation that in data centers, DRAM and network resources are underutilized [7, 1, 3, 4]. This gives us an opportunity to leverage those unused resources to extend switches' memory capacity with *low cost*.

While this low cost solution is appealing, there are several technical challenges before we can realize this in practice. These include performance, load balancing, and fault-tolerance. In the talk, I will describe key challenges and how we address them in GEM. We will also introduce our prototype implementation and preliminary evaluation results to demonstrate the practicality of GEM.

We hope this talk will spur the P4 community to think about more innovative in-network applications with GEM.

References

[1] Alibaba. Alibaba production cluster trace data. https://github.com/alibaba/clusterdata, 2017.

[2] P. Bosshart, G. Gibb, H.-S. Kim, G. Varghese, N. McKeown, M. Izzard, F. Mujica, and M. Horowitz. Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. In *ACM SIGCOMM*, 2013.

[3] Google. Google cluster-usage traces. https://github.com/google/cluster-data, 2011.

[4] S. Kanev, J. P. Darago, K. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. Brooks. Profiling a warehouse-scale computer. In *ISCA*, 2015.

[5] R. Miao, H. Zeng, C. Kim, J. Lee, and M. Yu. Silkroad: Making stateful layer-4 load balancing fast and cheap using switching asics. In *ACM SIGCOMM*, 2017.

[6] S. Narayana, A. Sivaraman, V. Nathan, P. Goyal, V. Arun, M. Alizadeh, V. Jeyakumar, and C. Kim. Language-directed hardware design for network performance monitoring. In *ACM SIGCOMM*, 2017.

[7] Q. Zhang, V. Liu, H. Zeng, and A. Krishnamurthy. High-resolution measurement of data center microbursts. In *ACM IMC*, 2017.