15-745: P4CKET Milestone Report

Sai Jaydeep Kudumula (skudumul), Darshil Kaneria (dkaneria)

20 November 2024

1 Project Progress

1.1 Project Link

The project repository can be found here.

1.2 Major Changes

There have been no major changes so far apart from the development environment. We were uncertain about where we would be setting up a P4 software switch and running traffic traces on. However, we have a concrete testing environment which involves setting up a single switch in mininet that connects multiple hosts (for the sake of simplicity during initial testing phases where the topology is not of paramount importance). We have structured our testing around the BMv2 software switch which has support for the V1Model architecture.

1.3 Accomplishments so far

We have met the milestones that we have had described in the project proposal. Listed are the various tasks that we had proposed and accomplished over the past few weeks.

- ✓ Conduct research on existing P4 optimization techniques and profile-guided optimization approaches. Literature review includes, but is not limited to, the following papers:

  1. **RMT**: Studying the structure of a switch with reconfigurable match-action units [3]
  2. **P4**: Understand the inner workings of the P4 programming language [2]
  3. **P4 Spec**: Go over the spec to further understand the structure of the language [1]
  4. **P2GO**: Explore a compiler-assisted profile-guided approach for optimizations in P4 programs [8]

5. **Morpheus**: Explore run time optimizations for P4 programs [7]
6. **OpenFlow**: Understanding current control plane solutions [6]
7. Understanding the structure of a switch in context of the data plane [5]
8. Understanding how compilers maps structures to the hardware [4]

✓ Setup the development environment and P4 Compiler

1. Setup the mininet environment which creates a single BMv2 switch topology
2. Create sample P4 programs along with table entries.

✓ Modify the P4C compiler to instrument code with profiling headers

1. By adding a pass into the middle-end of the P4C compiler (for BMv2 simple switch), we are able to insert a profiling header into the incoming packets.
2. The header collects information about the actions performed on the packet headers and the tables that the packet traversed. Each field in the profiling header corresponds to an action. Every time and action is performed on a packet, the field is updated with that information. This header is inserted into the packet as the innermost header.
3. The above information gives us the ability to create hot paths that packets of certain types traverse, allowing us to optimize away certain elements in the switch.

1.4 Unexpected challenges and Adaptations

We expected the setup of the software to be relatively straightforward. However, we had issues with our initial setup (setting up machines on Cloudlab according to the topology) so we had to fall back to using a software switch for testing and evaluation. Apart from this, we did not face significant challenges.

1.5 Revised Schedule

1. Week 4 - 5 : Adaptive Instrumentation
   (a) Member 1 and Member 2 : Implement hot path identification techniques and use it to adapt the instrumentation in the code.
   (b) Member 1 and Member 2 : Finding a sweet spot in terms of the frequency of recompilation and deployment to the switch
2. Week 6 : Evaluation
   (a) Member 1 and Member 2 : Integrate all the components and evaluate the performance of the implementation using benchmarking applications.

(b) Member 1 and Member 2 : Prepare the documentation and work on the final report

1.6 Resources Needed

Yes, we do have all the resources needed for completing the project. We are emulating the required hardware (i.e switches and servers) using software.

References

[1] P416 language specification. 2017.

[2] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. P4: programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95, July 2014.

[3] Pat Bosshart, Glen Gibb, Hun-Seok Kim, George Varghese, Nick McKeown, Martin Izzard, Fernando Mujica, and Mark Horowitz. Forwarding metamorphosis: fast programmable match-action processing in hardware for sdn. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, page 99–110, New York, NY, USA, 2013. Association for Computing Machinery.

[4] Lavanya Jose, Lisa Yan, George Varghese, and Nick McKeown. Compiling packet programs to reconfigurable switches. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, NSDI'15, page 103–115, USA, 2015. USENIX Association.

[5] Nick McKeown. Fast switched backplane for a gigabit switched router. 1997.

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[7] Sebastiano Miano, Alireza Sanaee, Fulvio Risso, Gábor Rétvári, and Gianni Antichi. Domain specific run time optimization for software data planes. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '22, page 1148–1164, New York, NY, USA, 2022. Association for Computing Machinery.

[8] Patrick Wintermeyer, Maria Apostolaki, Alexander Dietmüller, and Laurent Vanbever. P2go: P4 profile-guided optimizations. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, HotNets '20, page 146–152, New York, NY, USA, 2020. Association for Computing Machinery.