Building a Product Using P4

Simon Capper - Director of Software
Arista Networks Inc.

1st May 2019

Ugh - Tofino doesn’t know what a MAC address is!

- As a software engineer I realised that I’d have
  - to do a lot more work to do and have more responsibility to make the product work
  - more control over the design
- I had several questions
  - Will this change the way we write/design the control plane software?
  - What advantages can we gain from this new control?
  - Would we have a better product?
  - Would the end customer care?

This beats writing software for a “normal” ASIC

- The software team relished the challenge of designing the pipeline
- For traditional ASICs the software team has the task of translating the state of the control plane to the tables the ASIC designers put into the hardware
- On Tofino, using p4, we would get to design the pipeline rather than understanding someone else’s
- Our pipeline design would be closely aligned with the existing EOS software

So We Wrote our Own Pipeline

- Entirely developed in house
  - We didn’t use the reference design at all
- Programming in p4 on a real ASIC is nothing like regular software
  - I have renewed respect for those who do this and commit it to hardware in one spin
  - We had to get used to significant HW resource constraints, hard for your average C programmer who is used to x86 virtual memory address spaces
  - Making a good p4 compiler is hard, we spent a lot of time iterating with the vendor to get it working.
- This was more work for the software team
  - ramp up time on p4, both calendar and engineer hours, was longer than previous ASICs I’ve worked on
  - hard to have several people work on the same code, very easy to break each others build
    - continuous integration development model doesn’t work for shared p4 code

So We Wrote our Own Pipeline

- The pipeline does exactly what we want it to do
  - no extra features we don’t use
  - we have multiple programs that run on the same hardware platform
    - allows features and table sizes are scaled to what the end customer needs
  - feature scale is comparable with fixed function ASICs
  - no pipeline programming hacks to get a feature working, I can change the pipeline code instead
    - e.g. vlan membership check on underlay, incomplete ARP flooding
- Barefoot SDE generates the SDK
  - generated code is dependable for repetitive code (once you get the generator working)
  - we extended the code generation and also generate data structure APIs using the p4 tables
  - we made the ASIC tables look like C data structures where the ASIC is really just a specialized backing store
  - integrates tightly with the existing Arista software stack
  - for pipeline unit testing we generate golang APIs to program the white model
- New feature development is now significantly faster than other ASICs I’ve worked on

So what can we do with all this flexibility

- We built Tofino based switches
  - DCS-7170-64C
    - 2RU 64 x 100G + 2x10G
  - DCS-7170-32C
    - 1RU “32Q-model” - 32x100G + 2x10G
  - DCS-7170-32CD
    - 1RU “32D-model” - 32x100G + 2x10G
- Multiple “profiles” for a customer to choose from, including but not limited to
  - High scale NAT/PAT (dynamic, twice, static, multicast)
  - High scale VPC (support for large scale VxLAN VTEPs, v4 & v6)
  - Stateless Load Balancer (using ideas from the Beamer paper by Vladimir Olteanu et al)
  - Stateful L4 Firewall
  - Partial implementation of SMPTE 2202-7
  - Bring-your-own 3rd party custom pipeline support

Back to my questions...

- *Will this change the way we write/design the control plane software?*
  - It did, once past the initial ramp up period we can bring new features to market much faster than with fixed function ASICs, the tight coupling between the pipeline and control plane software makes a big difference here
- *What advantages can we gain from this new control?*
  - We can do more without involving the ASIC vendor
  - Our software architecture extends into the pipeline design
- *Would we have a better product?*
  - In the data center space no, cost and time to market are still the key factors in selling data center switches. non-standard features are not required
  - new p4 driven ASICs should come up faster, may give a TTM advantage?
  - We have implemented some x86 based network applications to move into the dataplane, our product can do more
- *Would the end customer care?*
  - Mostly no, they care about the product and its quality, not so much how we got there
  - A select set of customers love the pipeline access, for this p4 is the only real answer today

Some Final Thoughts on p4

- p4 is a brilliant way to expose the flexibility of Tofino, or any other flexible pipeline
- p4 benefits from having brilliant minds working on it, it is really good technically, as software engineer building switches I think it is the most exciting change in ASIC programming I have ever seen
- Combined with Tofino, p4 enables Arista to produce a variety of new types of network devices

Thank You

www.arista.com