Workshop 3 – Thursday April 22, 14:00 – 17:30

Beyond SDN. Programmable data plane: Abstractions, Architectures and Applications

Software-Defined Networking (SDN) separated the control and data planes from each other. Programmable data plane is the next logical step, allowing the forwarding devices to be equipped with functionality beyond processing a fixed set of network protocols. This has created opportunities for the recent developments of programmable ASIC switches, packet processing languages, FPGA-based reconfigurable switches and routers, and SmartNICs. These all, in turn, give rise to the emerging trend towards in-network processing that can be applied for a wide range of applications spanning from pure network functionalities (e.g., load balancing, query processing and caching, network telemetry, protocol offloading) to more advanced offloading of generic software applications into the network (e.g., in-network computation and Machine Learning Inference in the Data Plane). In this workshop, recent advances in various aspects of programmable data planes will be addressed, trying to identify the current trends and the possible future research directions for both abstractions, architectures and applications that can enable and/or leverage the concept of data plane programmability.

Workshop Chairs:

Jari Nurmi, Tampere University (TAU), Finland

Salvatore Pontarelli, Sapienza University of Rome, Italy

_____ WORKSHOP PROGRAM _____

1. Using high level synthesis to create high speed packet processing pipelines

Robert Wikander, Packet Architects, Sweden

Having developed switching/routing chips for a number of different companies (SwitchCore, Realtek semiconductor) I had seen first-hand how painful it had been to add new features and support addition ports and port speeds in new chips for different markets. Upon starting packet architects we considered what would be a good path forward when developing a new switch chip architecture which would span over a 10000x range in bandwidth (from a few megabits to 10s of terabits) and support from simple L2 switching to complex network edge switching / routing.
Today's HDL languages such as System Verilog and VHDL does provide many high-level features such as configurability/parameterization and object-oriented programming but these features seldom work when doing RTL (register-transfer-level) synthesizable code. What is worse is that different tools might work for different features thereby reducing what can and can't be used to a minimal set of acceptable design primitives.
What is worse is that languages which is used to create synthesizable RTL are very constrained and quite hard to change the code. A roughly comparison is that we want to write in python but current language only allows us to write in assembly language.
Therefore we came up with our own language, which starts with a base pretty similar to C and then expanded with a few hardware primitives (ram / registers). Using this language we can then automatically pipeline the hardware for GHz speeds of for slow megabit designs with minimal area, from 130nm to 5nm ASIC technology and for all FPGA technologies out there.

2. P4-based Network Traffic Entropy Estimation for the Detection of DDoS Attacks

Marco Savi, University of Milan-Bicocca, Italy

This talk discusses a novel strategy to estimate network traffic entropy in the programmable switches' data plane, by leveraging the limited number of P4-supported arithmetic and logical operations without any need to store pre-computed values in TCAMs. The proposed strategy is then used as a building block for entropy-based DDoS detection, which is proven to be effective and fully executable in the programmable data plane. Such an approach dramatically reduces the communication overhead with respect to the state-of-the-art solutions, which require switches to provide a substantial amount of monitoring statistics to a centralized collector.

3. Xilinx Labs Open-sourced SmartNIC Trilogy: NIC Shell, NIC-Attached Application, and Acceleration Platform

Chengchen Hu, Xilinx Labs Asia Pacific, Xilinx, Singapore

Due to the slowing of Moore's law, explosive data growth, and unprecedented computational demands, distributed computing systems have been evolving from a CPU-centric model to a network-centric style. Use of SmartNICs is becoming a paradigm in large data centers, which keeps improving the performance of networking and computing coupled systems. This talk introduces Xilinx Labs efforts towards open-sourced SmartNIC hardware and software.

1. Xilinx Labs OpenNIC is an open-source FPGA-based 'skinny NIC' shell empowering users to build custom functions into the control plane and the data plane of a SmartNIC, supplying hardware interfaces, kernel APIs, and user-space APIs.

2. Xilinx Labs Blockchain Machine is an open-source project which explores network-attached acceleration for Hyperledger Fabric, leveraging OpenNIC as its NIC shell. The design illustrates a typical SmartNIC application, which retrieves block/transaction data directly from the network interface to be processed through a configurable pipeline in hardware.

3. Four Xilinx Acceleration Compute Clusters (XACC) have been established at ETHZ, NUS, UCLA, and UIUC, equipped with the latest Xilinx hardware and software. This is a special Xilinx initiative to support research in adaptive computing acceleration for HPC and data center, including SmartNIC and network-attached applications.

4. Architectural Choices for the Programmable Data Plane

Hesam Zolfaghari, Tampere University (TAU), Finland

Packet processing systems are comprised of the control plane and the data plane. The data plane is the entity directly operating on network packets. A programmable data plane is not tied to any set of network protocols. It executes software corresponding to different networking protocols and standards. There are a number of architectural variants and choices that affect the complexity and performance of the programmable data plane. Some architectural variants are better suited to specific packet processing workloads. In this talk, the pros and cons of different architectural variants as well as architectural techniques for boosting performance will be discussed. The talk provides insight for architects of packet processing hardware.

5. Network Telemetry on a Budget

Gianni Antichi, Queen Mary University of London, UK

Commodity network devices support adding in-band telemetry measurements into data packets, enabling a wide range of applications, including network troubleshooting, congestion control, and path tracing. Unfortunately, including such information on packets adds significant overhead that impacts both flow completion times and application level performance. In this talk, I will introduce PINT, an in-band telemetry framework that bounds the amount of information added to each packet. The idea is to encode the requested data on multiple packets, allowing per-packet overhead limits that can be as low as one bit. I will discuss the implications of PINT on congestion control and path tracing and I will conclude with lesson learned and future steps.

6. Enabling Service Chaining (and much more) in eBPF

Fulvio Risso, Politecnico di Torino, Italy

The importance of eBPF is growing recently, with many companies providing network services (e.g., Cloudfare, Google, Facebook, Covalent) using this technology. However, eBPF focuses on data plane only, hence making the creation of complex network services (which require a control plane) more difficult than expected. Moreover, although eBPF allow creating multiple chained services, its solution (with "tail calls") is far from being easy to use. This talk will highlight the most important limitations of eBPF when creating complex chains of services and will introduce how these have been addressed in Polycube, an open-source project that delivers a built-in control plane for eBPF, easy and flexible service chaining, and a model-driven service abstraction that can automatically generate the skeleton of the service, enabling the programmer to focus on his added value logic, while delegating ancillary tasks to the Polycube framework.

7. Using PISA switches to accelerate data-centric applications

Alberto Lerner, University of Fribourg, Switzerland

Scaling-out has long been a strategy to deal with increasing volumes of data. Rather than relying on single, increasingly beefier machines, the idea is to interconnect regular machines in large numbers and rely on distributed processing. The approach makes the network a natural component of many data management and analytics platforms. The role of the network, however, has so far been that of data transport. With In-Network Computing (INC), this story started to change.

In this talk, we discuss our experience incorporating INC, particularly programmable data planes (PDP), in data and analytics applications. We argue that PDPs can be treated as accelerators from an application's architecture point of view, surprisingly flexible in some ways but highly restrictive in others. In the process, we describe the challenges and opportunities coming from this integration.

8. Keynote speech: When networks are programmable top-down and end-to-end: what then?

Nick McKeown, Stanford University

Large network owners commonly write or download the software that controls their networks, allowing them to determine how they are managed from the top level intent all the way down to how packets are processed in the forwarding plane. At some point, they will decide how packets are processed end-to-end, from user-space, through the kernel, NICs, switches and network elements. Which brings them to a place where they are fully in control of their own deeply programmable network platform. In this talk, I will talk about what they might do with this new found flexibility and control.