The Frenetic Project: Declarative Languages for Programming Networks

David Walker

Princeton University
dpw@cs.princeton.edu

For decades, traditional network devices such as switches, firewalls, load balancers and routers have been closed, proprietary platforms controlled by the major hardware vendors. Each device contains a combination of (1) hardware to forward packets efficiently along chosen network paths, and (2) software to run the distributed protocols that decide which network paths to choose. To configure or specialize these devices, network operators have had to learn a myriad of complex, vendor-specific interfaces and protocols. Moreover, there was no easy way to change the basic distributed routing algorithms these network devices implement.

Over the last few years, however, *software-defined networking* (SDN) has taken both the academic and industrial networking communities by storm. In a software-defined network, each switch exports a simple, standard and relatively direct interface to its underlying hardware. These switches are organized and managed by a separate, logically centralized *controller* machine or cluster of machines. A controller runs a general-purpose computation that reacts to network events such as changes in topology or traffic volume and decides how to route packets across the network. Based on these decisions, it sends commands to configure the switches it controls. By standardizing the hardware interface and separating out the decision-making software, this new architecture makes it possible to control and optimize networks in ways that were previously impossible. Google is already taking advantage of this technology to control the global backbone network that connects its data centers together [1]. Many other networking companies, both large and small, have also begun to innovate in this new space. However, despite the genesis of SDN in the networking community, many of the key problems are actually programming problems. Hence, researchers who understand the design of declarative programming languages have much to offer in this important new domain.

In this talk, we will discuss the Frenetic project [2], whose goal over the last several years has been to develop new, high-level, declarative, domain-specific languages for programming software-defined networks. In particular, we will discuss several of the core abstractions and programming language features we have developed, what key problems they solve, their formal semantics, and how to compile them to the underlying switch hardware. We will also touch on the next generation of software-defined networks and future opportunities for declarative language design.

The Frenetic Project is a large collaborative project centered between Cornell and Princeton universities. Since it began in 2009, the project has been led by Nate Foster, Jennifer Rexford and David Walker. We thank the wonderful contributions to the project made by Carolyn Jane Anderson, Shrutarshi Basu, Rebecca Coombes, Michael Freedman, Arjun Guha, Steven Gutz, Rob Harrison, Jean-Baptiste Jeannin, Nanxi Kang, Naga Praveen Katta, Dexter Kozen, Zhenming Liu, Matthew Meola, Matthew Milano, Christopher Monsanto, Nayden Nedev, Josh Reich, Mark Reitblatt, Cole Schlesinger, Emin Gün Sirer, Robert Soulé, Alec Story, Laure Thompson, and Todd Warszawski.

Acknowledgement. This work is supported in part by the NSF under grants CNS-1111520, SHF-1016937 and a Google Research Award. Any opinions, findings, and recommendations are those of the author and do not necessarily reflect the views of the NSF or Google.

References

1. Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. B4: Experience with a globally-deployed software defined WAN. In *ACM SIGCOMM*, 2013.
2. The Frenetic Project, 2013. See `http://frenetic-lang.org`.