Seamless Hardware Accelerated Kubernetes Networking

Mircea M. Iordache-Sica, Tula Kraiser, Olufemi Komolafe

2nd ACM SIGCOMM Workshop on Future of Internet Routing & Addressing (FIRA)
Outline

● Kubernetes Overview

● Hardware Accelerated Kubernetes (HAcK) Architecture & Goals

● HAcK LoadBalancer

● Evaluation & Future Work
Kubernetes Overview (1)

- Kubernetes has become a popular industry standard for deployment of applications and workloads
  - Public Cloud Providers offer managed services
  - On-premises deployments

- Base concepts: Pods and Services
  - Pods -> Containers
  - Services -> Collection of Pods

- IP-per-Pod
  - Traditionally achieved by Software-Defined Network Overlays
  - VXLAN, GRE, Public Cloud implementations
Kubernetes Overview (2)

- Impact on networking
  - Microservice management and orchestration
  - Virtual Network Functions
  - Network emulation

- Leveraging the dataplane too…
  - Offload state storage to the network
Kubernetes Overview (3)

- Kubernetes exposes a powerful paradigm
  - Extensible framework
  - Robust and reliable
  - Industry-proven

- Limitations in State-of-the-Art
  - Networking overlay implemented in software
  - Unification of application and network
  - Network Latency concerns
Hardware Accelerated Kubernetes (HAcK) Objectives

- Intent-driven control and management of application and network
  - User Specifies application and properties
  - Framework configures network

- Expand Software-Defined Networks to hardware
  - Leverage Network ASIC benefits (high bandwidth, low latency)
  - Offloading of Network Functions (e.g., NAT, Load Balancing)

- Avoid (expensive) host packet processing
  - Simplify delivery mechanisms
  - Clear and concise behaviour
HAcK Architecture (1)

- Leverage Kubernetes Event Notification mechanisms to track key state changes
  - CRUD pattern maps well to network automation
  - Controllers and Operators monitor events, make (further) changes

- Example events
  - Pod Create, Update, Delete
  - Service Create, Update, Delete
HAcK Architecture (2) - Examples

- Pod Creation
  - Address Resolution
  - Network Isolation

- Service Creation
  - Route advertisement
  - Address translation
HAcK Architecture (3)

- Configure network elements
  - OpenFlow
  - gNMI
  - SNMP
  - Etc.
Case Study: HAcK Load Balancer (1)

- Use HAcK Architecture to implement Kubernetes Load Balancing behaviour
  - Leverage in-network programmable dataplane for low latency
  - Manage network and container infrastructure simultaneously
  - Route advertisement and network isolation responsibility
  - High-availability networking
Case Study: HAck Load Balancer (2)
Case Study: HAcK Load Balancer (3)

- Network infrastructure
  - Arista 7170 Series
    - Intel Tofino Switch Chip - P4 Dataplane
    - Up to 64 100GbE ports
    - Stateless Load Balancer profile
    - Up to 12.8Tbps throughput, sub μsecond latency
    - Arista EOS API (eAPI) programmability
  - Top-of-Rack deployment
    - Server proximity simplifies packet flow

- Servers
  - MacVLAN networking
    - Multiple (logical) L2 interfaces attached to same NIC
Case Study: HAcK Load Balancer (4)

- HAcK Load Balancer monitors Kubernetes state
  - React to Service and Pod object events
  - Monitor IP assignments
    - External (Advertised)
    - Internal (Private to Kubernetes)
  - Create Match-Action Rules

- Control Networking Appliances (7170)
  - Upload Match-Action Rules in hardware pipeline
  - Maintain connectivity guarantees
Case Study: HAcK Load Balancer (5)

- Arista 7170 Series Stateless Load Balancer (SLB) profile
  - Maintains original packet integrity
  - Prepends L2/L3/VxLAN header with final destination
  - Delivers directly to intended Pod

- On Server
  - Kernel Driver to parse 7170 SLB Headers
  - Pods are aware of the External IP (VIP) via Dummy interface
Deployment of HAcK Load Balancer (1)

- Data Centre environment
  - 3 Linux Servers (master + 2 workers)
  - 2 Arista 7170-32CD-R ToR Switches

- Each Server has Kernel Driver loaded

- Each Pod has MacVLAN network interface
Deployment of HAcK Load Balancer (2)

- Scaling behaviour
  - 32,000 Pods
  - 1,000 Services

- Latency Performance
  - Delivery+wire+ingress
  - TCP Cubic with multiple windows

- Results
  - Sub-millisecond for small (<16kB) windows
  - Server-side congestion for high loads
    - 7170 Retransmissions noticed
    - Servers not optimised for High-performance scenarios
Summary

- Kubernetes opens up many possibilities
  - Improve State-of-the-art Research
  - Simplify workload orchestration
  - Enhance Software-Defined Networking space

- HAcK brings in network infrastructure management
  - Architecture shows serious promise
  - Unify application and network operations
  - Enable Intent-driven networking
  - Simplify operation of end-to-end infrastructure

- HAcK Load Balancer shown as Case Study
  - Deployed in real-world scenario!
  - Use Arista 7170 Series switches for high-performance network workload
Future Work

● Comprehensive evaluation of HAcK Load Balancer
  ○ Kubernetes Event rates
  ○ Switch programmability rates
  ○ Diverse traffic

● Additional HAcK Architecture implementations
  ○ Network isolation
  ○ Service advertisement
  ○ Traffic engineering
Thank You

www.arista.com