# Evolving Virtual Networking with IO Visor

www.huawei.com

Yunsong Lu Yunsong.lu@Huawei.com
Principal Architect, Virtual Networking


HUAWEI TECHNOLOGIES CO., LTD.

# Content

- IO Visor and eBPF
- Dynamic Network Monitoring
- Micro Data Plane Container for Network Functions
- Network I/O

# IO Visor built on eBPF

- **eBPF is an in-kernel Virtual Machine**
  - Integrated in Linux Kernel since 3.16
  - eBPF introduction at Linux Collaboration Summit 2015
  - Evolution of BPF indeed: far beyond “packet filtering”
- **IO Visor Project:** www.iovisor.org
  - Collaborative Project of Linux Foundation
  - Community committed to Innovate, Develop and Share IO and Networking functions
- **Use Cases** (links in appendix)
  - Tracing, Analytics, and Debugging
  - Networking
  - Hardware Acceleration

# Virtual Network Monitoring

# Virtual Network Monitoring

- Virtual Networking evolving with the “hype” of LXC and Docker
- Application-Driven Networking(**ADN**) rises
  - Network created/destroyed following application deployment
  - Virtual networking extended to socket layer
- **Visibility** of application virtual networks is crucial
- Traditional network monitoring techniques are out-of-date
  - tcpdump
  - port mirroring
  - static probe points , etc.

- How should we monitor virtual networks with high performance and necessary visibility?

# Virtual Network Monitoring w/ IO Visor

- **Monitoring whole virtual network stack** from socket to virtual switch to physical NIC
  - existing hooks (can be extended)
  - eBPF+kprobe
- **Dynamic tracing** programs are loaded on demand with minimum interference
  - Parsing, counting, profiling, and analysis
- **Highly Efficient** in-kernel VM (close to native x86 code)
  - JIT for x86 and ARM64
  - Maps for data sharing between kernel and userspace
- Many tools and helper functions available
  - https://github.com/iovisor

# Canal View powered by IO Visor

- Canal is the Container Networking Framework from Huawei
- Canal View is the topology-based virtual networking monitoring system
- Monitoring **Application-to-Application** network traffic
  - Bandwidth, latency, and packet loss rate, etc.
  - Rating Network SLA quality
- On-demand monitoring all virtual network components in connects
- **Optimizing** network utilization and performance based cluster-wide data

# NFV2.0 Data Plane

# Data Plane of Virtual Networking

- PLUMgrid pioneered on implementing network functions with in-kernel IO Visor
  - https://www.iovisor.org/sites/cpstandard/files/pages/files/io_visor_white_paper.pdf
- Now supports P4, C, etc. front-end programing languages
- Example IO Modules available at https://github.com/iovisor
- Can replace OVS data path with improved performance
- Write your network function in user space, run it in kernel

# NFV Data Plane

- **NFV1.0** architecture separates VF data planes from NFVI data plane
  - Made it easy porting existing embedded software to virtual machine
  - Inefficient because of unnecessary I/O cross domains
  - Pay high price for reusing IT virtualization technology
- **Berkeley E2(Elastic Edge)** as NFV runtime framework
  - Base on BESS, a modular software switch
  - Chain network functions with dynamic scaling and fault tolerance (design goals)
  - Sponsored by Huawei, Intel, and AT&T

- Can NFV be Could Native?

# NFV Data Plane w/ IO Visor

- **Cloud Native NFV** is the way to go
  - Distribute and deploy virtual function with LXC and Docker
  - Eliminating the overhead of Virtual Machine
  - Fully decouple control plane and data plane
  - Use IO Visor as Data Path Container

# Huawei's MDPC for NFV2.0

- **Convergence of NFV, Cloud Native, IO Visor, and E2**
  - IO Visor as MDPC (Micro Data Plane Container)
  - Applications developed and deployed with Docker-like mechanism
  - Micro functions as reusable micro-services for expedited development
  - Converged data planes eliminating cross-domain/cross-VM overhead
  - Auto-scaling on heterogeneous hardware platform (x86 and ARM64)
  - Enabling hardware acceleration (FPGA, Multi-core, NPU, etc.)
  - Open Source infrastructure, protected NF code written in user space

- We are building ecosystem with ISVs, IHVs, and partners together

# Network I/O Framework

# Network I/O

- Move network packet/data between two spaces/domains:
  - Hardware and software (NIC drivers)
  - Host and VM (virtual I/O like virt-io)
  - Kernel and Userspace (mmap, share memory, etc.)
- Network I/O is about driver ecosystem
  - Sustainable hardware drivers for multiple vendors' hardware
  - Balance between manageability functions and performance
  - Resource sharing among many applications
- Also about predictable performance cross platform: x86 and ARM64

# Build Competitive Network I/O Framework

- Challenge to Linux Kernel
  - Network Performance not comparable to DPDK-based application
  - Many projects proposed to bypass kernel stack
  - Kernel has the best driver ecosystem, which is missing anywhere else
- Build Faster Network Data Plane in Kernel
  - XDP proposed by Facebook, currently under IO Visor project
  - Huawei's CETH Driver Framework is the starting point
  - Rebuild high-performance stack from Network I/O
  - Also need to define Network I/O API for userspace applications
  - Hardware vendors only write and maintain one driver per hardware

# Links

- IO Visor Project: www.iovisor.org
- Github: https://github.com/iovisor
- eBPF Introduction: http://events.linuxfoundation.org/sites/events/files/slides/bpf_collabsummit_2015feb20.pdf
- Linux Performance Analysis by Brendan Gregg: https://www.usenix.org/conference/lisa14/conference-program/presentation/gregg

# Thank you

www.huawei.com