Programmable Data Plane Architecture for the Network Edge

Mario Baldi, Diego Crupnicoff, and Silvano Gai
Distinguished Technologist
Pensando Systems, Inc.
Goals/Outline

- Distributed Services at the Network Edge: the Pensando Platform
- Distributed Services Card Architecture
- Representative use cases
  - How the card architecture supports them
- Performance evaluation of a sample use case
Pensando Distributed Services Platform

Centrally Managed
Policy and Services Manager

Policy
Orchestration & Provisioning

Automation
REST API

Observability
Troubleshooting & Security

Ecosystem
Compute, Analytics, IT Ops

Stateful Firewall
Encryption & TLS Offload
Micro Segmentation
Load Balancer
Networking
Storage Services

Telemetry

Pensando Distributed Services Cards

P4 Programmable Processor
DSC Architecture

Host Interface

PCle
Memory

Coherent Interconnect

P4 Packet Processing Dataplane
ARM Cores
Service Processing Offloads

Packet Buffer
Traffic Manager

Ethernet Port

Network Interface

Ethernet Port
Network Interface
Sample Use Cases
TLS Offload

- Protocol processing offload is a typical SmartNIC application
  - The DSC is not a SmartNIC, but can be used as such
- Especially the ones requiring significant resources
  - Memory
  - CPU
- TLS Offloading is a great example
  - TCP connection management
  - TCP state handling
  - TLS session management
  - Data encryption and decryption
Possible Implementation: Proxy

TCP port 80
Proxy running on DSC
TCP port 443
HTTPS
TLS Offloading Support

- ARM involved only in connection/session setup/tear down
- Pipeline ensures
  - Wire speed throughout
  - Minimal delay
  - Minimal jitter

1. Packets are generally processed by the pipeline
2. TCP connection and TLS session initiation packets are forwarded to the ARM cores for software processing
3.1. Encryption/decryption performed by service processing offload
3. Subsequent packets are processed in the pipeline

- Handle connection establishment
- Install state in pipeline tables

- PCIe
- Memory
- Coherent Interconnect
- P4 Packet Processing Dataplane
- ARM Cores
- Service Processing Offloads
- Packet Buffer Traffic Manager
- Ethernet Port
- Network Interface
- Host Interface
- Network Interface
NVMEoF/TCP
Non-Volatile Memory Express Over Fabric over TCP Transport

- Access a remote disk as if it were local
  - Through a regular NVMe driver
- Multiple transports including
  - RDMA - Remote Direct Memory Access
  - TCP
NVMEoF Offloading Support

(1) NVMe commands

(2) NVMe commands translated into NVMEoF capsules

- Load balance across remote controllers
- Encapsulation
- TCP segmentation

(2.1) encryption/decryption; data digest generation/verification

(3) encryption/decryption; of data at rest

© 2020 Pensando Systems
Distributed Stateful E-W Firewall

Firewalling E-W traffic is particularly challenging

- Large volume compared to N-S
- Applications expect small latency
- Appliances are not suitable as they would create “traffic tromboning”
The DSC is the perfect spot where to implement this

It is on the path of each packet

Flow caching to reduce latency

- Evaluate rules on first packet
- Install entry in flow cache table for handling following packets
Distributed Stateful Firewall Support

(1) packets belonging to a known flow are forwarded directly (flow cache table)

(2) packets of new flows are further processed in the pipeline to evaluate rules

(3) packet and corresponding action are passed to ARM cores

(4) packet is passed to pipeline for processing based on newly installed flow cache entry

Memory shared by ARM cores and pipeline ensures that entry is up-to-date

Software creates forward and reverse flow entries in the flow cache table

Ethernet Port

Host Interface

Network Interface

Network Interface

Ethernet Port

Dataplane

Packet Processing

P4

Packet Buffer

Traffic Manager

Coherent Interconnect

ARM Cores

Service Processing Offloads

Memory

PCIe

Packet Processing Dataplane
New Flow Installation Options

Most critical task

- Pipeline extracts relevant metadata and passes them to ARM
- Pipeline creates forward and reverse entries

1 M new flows per sec

- Packet and action passed to ARM
- ARM software parses packet
- ARM software creates forward and reverse flow entries

3 M new flows per sec

- Pipeline extracts relevant metadata and passes them to ARM
- ARM software creates forward and reverse entries

ARM Heavy

ARM Light

Pipeline Heavy

Work in progress

Unique features of pipeline processing units

Tight integration (hardware architecture)
Performance
Host Adaptor Mode

![Diagram showing Host Adaptor Mode](image)

- **Bandwidth**
- **Packets per second**

![Graph showing RX + TX Packets per second vs. TCP MSS size (bytes)](image)

- **RX + TX Packets per second (Mpps)**
- **TCP MSS size (bytes)**
- **RX + TX Bandwidth (Gbps)**

© 2020 Pensando Systems
Bump-in-the-Wire Mode

Throughput: 40 Mpps
Latency: 3 μs
Jitter: 35 ns
In summary ... to conclude

- Distributed Services Card Architecture
- How it can be leveraged to implement diverse services and applications
  - Possibly offloading the host
  - Moving them from somewhere else
- Achieve very high performance
  - High throughput
  - Low latency
  - Low jitter
- No performance hit on the host
Thank You

baldi@pensando.io
www.pensando.io
blog.baldi.info
linkedin.baldi.info