SmartNIC Design Considerations

Latency Optimized vs No Loss Packet Capture

Global STAC Live
1 November 2021
Agenda

• Corporate overview
• Packet Capture Trade-offs
• Challenge: Microburst
• Challenge: Host Applications Cannot Keep Up
• How a Napatech SmartNIC Addresses these Challenges
Napatech Builds SmartNICs

- We design high-performance programmable SmartNICs to accelerate applications and capture data at line rate
- Napatech Link™ software supports Capture, Inline, Virtualization
- US Manufacturer with extremely high reliability
Napatech at a Glance

- 18+-year history delivering FPGA-based technology to customers globally
- Unparalleled expertise accelerating compute-intensive applications on standard open servers
- Targeting rapidly expanding $2.7B Programmable NIC market by 2024
  - Public company NAPA.OL
Some of the Finance Companies Working with Napatech

- PICO
- Morgan Stanley
- Velocimetrics
- Ameriprise Financial
- BNY Mellon
- State Street
- Citadel
- Instinet
- Jane Street
- Bank of America
- Refinitiv
- BME
- Maven
- Jump Trading International, Llc
- Millennium
- TOWER Research Capital
- LDA Technologies
- Invesco
- UBS
The Dilemma

Network interface card architecture cannot be optimized for both ultra-low latency client/server communication and 100% packet capture at all speeds.
Packet Capture Trade-offs and Challenges

Server NICs are designed for client/server communication
- Ultra-low latency, packet by packet delivery to clients
- Packet delivery not guaranteed at high network traffic loads
- Not optimized for full throughput for any network traffic load
- When used as a capture device, packet loss can occur
- Packet ordering not guaranteed

Packet capture NICs are designed for 100% packet capture
- Designed for packet capture of market data for a complete market picture
- Optimized for capturing all packets no matter traffic conditions (bursts, congestion, packet size or overloaded application)
- Architected with large buffers to capture 100% of the network packets, including microbursts
- Nanosecond resolution time-stamping and synchronization
Challenge: Microburst

Problem
• Volatile stock markets put pressure on trading infrastructure
• Microsecond bursts are not always captured
• Packet queues are small in server NICs causing packets drops during a microburst event

On-board Packet Buffering to Absorb the Burst
• A SmartNIC that provides full theoretical throughput for all packet sizes and all network loads
• Advanced host memory buffer management guaranteeing 100% capture, even during a microburst event
Microburst

![Graph showing Microburst performance at different bit rates (20 Gbps, 40 Gbps, 80 Gbps, 100 Gbps) over time. The graph includes lines for average line utilization, time sample window, average received bits per second, and received bits per second. The graph highlights the fluctuation in bit rates and utilization over time, indicating the microburst effect.]
Example: Handling Small Packets in Bursts

The standard NIC provides 61% line rate at 64 bytes compared to full theoretical throughput for the Napatech SmartNIC with Link™ Capture Software.

![Lossless Throughput Test](image_url)

See more: DN-1223 Solution Description Generic on Napatech SmartNICs
Challenge: Host Applications Cannot Keep Up

Problems
- Busy CPUs
- Applications cannot keep up with the network load
- Congestion in the delivery of data to the application

On-Board Memory
- Buffering of frames
- On-board memory for buffering assures guaranteed delivery of data, even when there is congestion in the delivery of data to the application

Large Host Buffers
- Advanced host memory buffer management enabling ultra-high CPU cache performance
How a Napatech SmartNIC Addresses these Challenges

SmartNIC Hardware and FPGA software purposely designed for packet capture

• Architecture optimized to capture 100% of network packets regardless of network utilization or packet size
• Packets are timestamped in hardware for accurate event information and network performance analysis
• Physical ports are merged in FPGA, guaranteeing host applications see packets in exact order they traversed the network
• Large onboard packet buffer ensures:
  • Zero packet loss when network utilization is high (microbursts)
  • Zero packet loss when PCI express bus is busy
• Large host buffers ensure that host applications can keep up network load
• Zero-copy DMA kernel bypass
• Distribution to multiple host buffers based on flow (RSS)
SmartNIC FPGA Capture Architecture

- DDR Memory 12GB
- Time Stamp & Synchronization
- MACs
- Channel Merge
- Packet Decode
- Flow Proc.
- Filter, slice, compare, & buffer split
- Buffer System & Handler
- PCI Express
- Statistics
Napatech Link™ Capture FPGA Software

Napatech NT200A02 running Napatech Link™ Capture Software

Guaranteed Delivery

- Full throughput Rx/Tx for any packet size
- Zero packet loss
- Minimum 500 millisecond receive burst buffer @200Gbps (12GB)
- Optimized PCIe bandwidth utilization
- Guaranteed packet ordering

Low CPU Utilization

- Optimized for Intel/Xeon and AMD architectures
- Higher CPU cache performance, advanced host memory buffer system
- Higher CPU core utilization, advanced CPU load distribution
Advanced Features Examples

**Zero Packet Loss**
Guaranteed zero packet dropped under the most demanding conditions across all packet sizes, even at sustained line speeds.

**Burst Buffering**
Each Napatech SmartNIC has onboard memory to handle network microburst, ensuring that all packets are captured and delivered to the application over the PCIe bus.

**Hardware Timestamping**
Ensures exact timing information for when packets traversed network.

**Traffic Replay**
Allows high fidelity replay of a PCAP file with nanosecond precision to reproduce exact network behavior in test and measurement use cases.

**Excellent PCIe Performance**
PCIe throughput is maximized to so that the best possible network capture performance is achieved.

**NUMA Balancing**
Allows optimization of host processes and application threads to minimize NUMA to NUMA communication over the QPI interface.

**Optimum Cache Utilization**
Transfer of packets over the PCIe bus are optimized for the L3 cache in the processor, minimizing the frequency that the L3 cache is flushed, improving the overall performance of the host application.

**Packet Sequencing**
Ethernet port merging ensure that packets are delivered to application across multiple port in the correct order.

**Traffic Forwarding**
Built in packet broker functionality where incoming packets can be filtered and or load balanced out one or more interfaces, potentially eliminating the need for expensive load distribution solutions.

**Intelligent Multi-CPU Distribution**
Incoming packets can be sorted to 128 host buffers based on flow, L2-L4 filter, or a combination thereof, resulting in efficient utilization of all server CPU cores.

**Traffic Generation**
Generate traffic based on PCAP files, or open-source applications like TrEx at all line speeds up to 100Gbps.
Join Us at the Next Napatech Breakout Session

Packet Capture as the Market Moves to 100G and Beyond

What are the issues encountered when moving from 10/40G to 100G?
How to capture and accurately replay the recorded market data?
Stay Connected With Napatech

On the Web:

Social Media:

Other News:

www.napatech.com

www.napatech.com/newsletter

www.google.com/alerts

www.napatech.com/newsletter
Thank You!
Link™ Capture Software
Packet Processing Overview

- Merge in time stamp order with 1 nanosecond time stamp resolution
- GTP, IP-In-IP, NVGRE, VXLAN, IPv4, IPv6, MPLS, VLAN
- 2/3/5/6-tuple flow match (36K flows)
- Protocol match
- Pattern match
- Error conditions
- Match/action filter
  - Forward to Rx queue
  - Forward to port
  - Discard
  - Load distribute
  - Slice
  - Meta data
- Deduplication
- IP fragment handling
- Optimal utilization of PCIe3 16-lane enables full duplex 100G throughput
- 128 Rx queues, 128 Tx queues
- Contiguous packet bursts for optimized CPU cache utilization

Server Memory
- Rx Queues: Rx000, Rx001, Rx127
- Tx Queues: Tx000, Tx001, Tx127

Application
- CPU Cores

Ports
- Merge & Time Stamp
- Frame Decoder
- Flow-match
- Filtering
- Hash
- Slice
- Dedup
- Meta data
- Burst buffer
- Rx Queue Handler
- Tx Queues
- Tx Scheduler
- Tx Queue Handler
- PCIe

Full throughput / zero packet loss

Time sync:
- OS time
- IEEE1588 PTP
- PPS

Precise Tx timing
Rate control
# NIC Usage Comparison Chart

<table>
<thead>
<tr>
<th></th>
<th>Standard NIC</th>
<th>Latency Optimized NIC</th>
<th>Capture NIC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary use</strong></td>
<td>General data center deployment</td>
<td>Financial trading servers</td>
<td>Network performance and security analytics</td>
</tr>
<tr>
<td><strong>Key requirements</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line rate throughput</td>
<td>Less important</td>
<td>Somewhat important</td>
<td>Very important</td>
</tr>
<tr>
<td>(any packet size)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero packet loss</td>
<td>Less important</td>
<td>Somewhat important</td>
<td>Very important</td>
</tr>
<tr>
<td>(100% packet capture)</td>
<td></td>
<td>Packet loss can cause retransmission</td>
<td>Packet loss critical for security applications</td>
</tr>
<tr>
<td>Low latency</td>
<td>Somewhat important</td>
<td>Very important (100s nsec)</td>
<td>Less important (u secs)</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware type</td>
<td>Standard Ethernet ASIC</td>
<td>FPGA or specialized ASIC</td>
<td>FPGA</td>
</tr>
<tr>
<td>Hardware latency*</td>
<td>Low</td>
<td>300 + nsec</td>
<td>N/A</td>
</tr>
<tr>
<td>FPGA configuration**</td>
<td>(N/A)</td>
<td>Optimized for low latency</td>
<td>Optimized for high throughput: Store and Forward</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Per packet processing</td>
<td>Optimized for zero packet loss: Large buffers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Optimized for CPU utilization: All frame sizes</td>
<td>Optimized for CPU utilization</td>
</tr>
<tr>
<td>Standard Linux network driver</td>
<td></td>
<td>Optimized for low latency</td>
<td>Optimized driver</td>
</tr>
<tr>
<td>Proprietary driver</td>
<td></td>
<td>Optimized for high throughput</td>
<td>Optimized for low CPU utilization</td>
</tr>
<tr>
<td>Proprietary driver</td>
<td>Proprietary for low latency</td>
<td>Proprietary for high throughput</td>
<td>Proprietary</td>
</tr>
<tr>
<td>FPGA code (loaded on the FPGA SmartNIC) can configure the SmartNIC to be either a Standard NIC, an HFT NIC or a Capture NIC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Integration &amp; Deployment</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Considerations</td>
<td>Requires seamless I&amp;D Deployment</td>
<td>Customization is acceptable</td>
<td>Some customization is acceptable</td>
</tr>
<tr>
<td>Driver</td>
<td></td>
<td>Low latency is top priority</td>
<td>Packet capture is top priority</td>
</tr>
<tr>
<td>Standard</td>
<td>Linux network driver</td>
<td>Proprietary for low latency</td>
<td>Proprietary</td>
</tr>
<tr>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Optimized for high throughput</td>
<td>Optimized for high throughput</td>
</tr>
<tr>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Optimized for CPU utilization</td>
<td>Optimized for CPU utilization</td>
</tr>
<tr>
<td>Standard OS API</td>
<td>Linux/Windows</td>
<td>Proprietary for low latency</td>
<td>De facto and proprietary</td>
</tr>
<tr>
<td>Proprietary</td>
<td>Proprietary</td>
<td>PCAP, DPDK, vendor proprietary</td>
<td></td>
</tr>
<tr>
<td><strong>Cost (relative)</strong></td>
<td>$</td>
<td>$$</td>
<td>$$$</td>
</tr>
</tbody>
</table>

*FPGA SmartNIC Hardware has same low latency as a standard NIC (ASIC)

**FPGA code (loaded on the FPGA SmartNIC) can configure the SmartNIC to be either a Standard NIC, an HFT NIC or a Capture NIC