Hardware Security for Performance Based TCP Attached Storage

Scott Schweitzer
Solarflare a Xilinx Company
30-Seconds of Background

- 17 years with IBM, first 11 at Research
- 3 years with NEC as Itanium Product Mgr
- 8 years with Myricom as HPC/HFT Sales/Mktg
- 6 years with Solarflare as Sales/Mktg
- Now Xilinx outbound DCG Mktg
Software Based Acceleration
Our Increasing Appetite for Data

- Since the start of this century, digital storage has grown exponentially
- “Big Data” apps now consume petabytes of data at once
  - Data Mining/Warehousing
  - Real-Time Analytics
  - Artificial Intelligence
  - Business Intelligence

Kernel I/O Slows Down Data

- Big Data requires fast I/O solutions
- Normal I/O solutions run all I/O traffic through the OS Kernel
  - Memory copies
  - Context switch – 1400ns Intel E5520*
  - Lag and increased CPU usage
- While great for managing “uncontrolled” environments, this approach impacts performance of cloud environments
First There Was User Space Acceleration

- By bypassing the kernel, user space I/O solutions overcome this issue
  - No context switching
  - No memory copies
- User space I/O increases bandwidth while decreasing CPU utilization
  - Improved CapEx and OpEx
  - Better solution scalability
User Space Performance is Proven

- Pioneered user-space acceleration for the most demanding markets:
  - FinTech
  - Managed Service Providers

- Onload in the cloud provides performance gains for:
  - IMDBs
  - Software Load Balancers
  - HTTP and Reverse Proxy Servers

- Primarily software based application acceleration
SSDs Have Changed Storage

![Latency Comparison Graph]

- **HDD**: 5000 μS
- **SSD**: 100 μS
- **SCM**: 10 μS

This graph illustrates the significant reduction in latency between traditional HDDs, SSDs, and SCM (Solid State Memory). SSDs offer a substantial improvement over HDDs in terms of access times, making them the preferred choice for modern storage solutions.
# NVMe-oF Options – Protocol Stacks

<table>
<thead>
<tr>
<th>Application</th>
<th>NVMe Transport / Driver</th>
<th>File System / Block IO</th>
<th>Application</th>
<th>NVMe Transport / Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVMe FC Plug-in</td>
<td>NVMe RoCE Plug-in</td>
<td></td>
<td></td>
<td>NIC Driver</td>
</tr>
<tr>
<td>FC4 Stack</td>
<td>RDMA OFED Stack</td>
<td></td>
<td></td>
<td>TCP</td>
</tr>
<tr>
<td>FC Driver</td>
<td>UDP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIC Driver</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diagram:**
- **Fibre Channel Switch**
- **DCB Ethernet Switch**
- **L2 Ethernet Switch**
- **NVMe/TCP Plugin**
NVMe/TCP Testing on SuperMicro Server

Initiator “App” Server
Red Hat RHEL Linux 7.4
SAMSUNG 1750 SSD

Target Storage Server
Red Hat RHEL Linux 7.4
SAMSUNG 1750 SSD

10/25/50/100G
TCP

10/25/50/100G
TCP

 Ethernet Network

VPN

Tested by Solarflare

High Throughput | Low Latency

<table>
<thead>
<tr>
<th>FIO Test Results</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Thread 4K READ Throughput</td>
<td>35MB/s</td>
</tr>
<tr>
<td>32 Thread 4KB READ Throughput</td>
<td>1010 MB/s</td>
</tr>
<tr>
<td>Random 4K READ Latency</td>
<td>99 μs</td>
</tr>
</tbody>
</table>
NVMe/TCP Benchmarking Read Latency

![Graph showing NVMe/TCP benchmarking read latency with latency in µsec on the y-axis and threads on the x-axis. The graph compares local, nvme-tcp-sfc, and rdma.]
Hardware Based Acceleration
FPGAs are a Critical Acceleration Component
Accelerated Apache Spark

ETL, Streaming Analytics, Spark SQL, Kafka, Hive

Zero code change

Intelligent, automatic computation across CPUs and Alveo

2x-30x acceleration

Average 4x-5x on Alveo
Hardware Based Security for Network Storage
Four Keys to Hardware Based Security

- Tamper Resistant NIC platform
- Agentless Enforcement & Telemetry
- Secure External Control Plane
- Centralized Orchestration
Tamper Resistant NIC Platform

- Firmware must be digitally signed & checked
- Log ALL attempts to load tampered firmware
- Be capable of terminating a secure connection without requiring an agent
- NO Root access to security functions
- Binding process to link NIC to central controller
Agentless Enforcement & Telemetry

- Agentless is means NO host based software is required
- Enforcement on all traffic, Rx and Tx
  - Five tuple, Regx or both depending on HW
  - Filtering & Alerting
  - Support for millions of rules and IPv6
  - Nominal latency impact
- Flow reporting to central controller
  - New flows
  - Packet and byte counts
  - Rates/flow if HW can support
Secure External Control Plane

- Pathway needs to exist so centralized management can be enabled

- On the NIC side this TCP management needs to be secure & fully autonomous
Centralized Orchestration

- Terminate secure control plane for all bound NICs under management
- SDN “Neutron” like controller with RestAPI
- Support for high level abstractions
- Cluster capable database to store all objects
- Double log everything, standard Linux system logs, and proprietary
Why Hardware Enforcement?

- Because software solutions are easily beat
- Hardware enforcement should be transparent to root users
- Adds another layer to the security onion
- Zero attackable local surface area
- Cloak server from all but service ports
  - Becoming a high performance Honeypot
- Extremely high performance ~200ns
  - Fortinet appliances by contrast 2500ns
- Compliance validation solution
Rise of the SmartNIC

All the above is possible and very probable with the next generation of SmartNICs