The Co-Evolution of Data Center Hardware and Systems Software

Lintao Zhang | Principle Researcher
Microsoft Research Asia

25th June, 2019
Introduction

Many classic systems software are designed decades ago, with many common assumptions:

- CPU is the Central Processing Unit, it is fast, and is getting faster
- Disk is slow, especially for random accesses
- Network is unreliable
- Memory hierarchy is consisted of CPU cache, DRAM, and disk paging.
- Accessing hardware needs to go through OS kernel
- Locality is important, data should be stored close to compute

Many of the assumptions are no longer true in modern data centers
Boot Your PC at 15GB/s With Gigabyte's New PCIe 4.0 SSD Adapter

The debut of AMD's third-gen Ryzen chips hastened the arrival of blazingly-fast PCIe 4.0 SSDs, but we'll soon be jaded about SSDs with a "paltry" 5GB/s of throughput, especially with new models coming with even faster speeds. But what if you could boot your PC with an SSD adapter that hits 15GB/s?

That's the purpose of Gigabyte's prototype quad-SSD adapter. This new add-in card snaps into your PCIe slot to get access to the full blazing 64GB/s of theoretical throughput available through the PCIe 4.0 x16 connection. After you add in the annoying, yet unavoidable overhead of the PCIe interface and RAID, Gigabyte's new prototype has hit up to 15GB/s when paired with four of the company's new Aorus PCIe 4.0 SSDs. Each of those SSDs hit up to 5GB/s, so the card could potentially push up to 20GB/s with further tuning.
The rise of domain-specific computing

CPU: End of frequency & Dennard scaling

Energy Efficiency (GigaOps/Watt)

Estimated Year of Deployment

Note: assumes power consumption of 160W/TPU2 chip (not confirmed)

Moore’s Law (2x per 1.5 years)
Still working for specialized hardware

Credit: Doug Burger
Cloud Architecture built around Smart NIC
Kernel can be by-passed for High Speed IO

Traditional file system calls and network IO are usually in OS kernel
- May incur too much overhead

RDMA, DPDK and SPDK try to move data plane to user space and by-pass kernel
- But they introduce new APIs

Credit: https://blog.selectel.com/introduction-dpdk-architecture-principles/
Agenda

1. Introduction
2. Two examples that take advantage of new hardware trends
3. Going forward and Conclusion
Agenda

1. Introduction
2. Two examples that take advantage of new hardware trends
3. Going forward and Conclusion
Example 1: Key-Value Store in Data Centers

Key Value Store is a key piece of systems infrastructure:

- Traditionally used as a cache in the cloud
- Now often act as a share data structure
- Example usage: database, graph processing, parameter server,

Modern applications require:

- High throughput
- Low tail latency
- Write intensive
- Support vector and atomic operations
Key-Value Store Architectures

**Kernel TCP/IP**

- Bottleneck: Network stack in OS (~300 Kops per core)

**Kernel Bypass**

- Bottlenecks: CPU random memory access and KV operation computation (~5 Mops per core)

**One-sided RDMA**

- Communication overhead: multiple round-trips per KV operation (fetch index, data)
- Synchronization overhead: write operations

**KV-Direct: Leverage SmartNIC**

- Offload KV processing on CPU to Programmable NIC
KV-Direct Architecture

Hardware Configuration

KV Processor
KV-Direct Performance Characteristics

Uniform workload throughput

Skewed workload throughput

Latency

Scalability
<table>
<thead>
<tr>
<th></th>
<th>Tput (Mops) (GET / PUT)</th>
<th>Power (Kops/W)</th>
<th>Architecture</th>
<th>Latency (us) (GET / PUT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memcached</td>
<td>1.5 / 1.5</td>
<td>5 / 5</td>
<td>TCP/IP</td>
<td>50 / 50</td>
</tr>
<tr>
<td>MemC3</td>
<td>4.3 / 4.3</td>
<td>14 / 14</td>
<td>TCP/IP</td>
<td>50 / 50</td>
</tr>
<tr>
<td>RAMCloud</td>
<td>6 / 1</td>
<td>20 / 3.3</td>
<td>Kernel bypass</td>
<td>5 / 14</td>
</tr>
<tr>
<td>MICA (12 NICs)</td>
<td>137 / 135</td>
<td>342 / 337</td>
<td>Kernel bypass</td>
<td>81 / 81</td>
</tr>
<tr>
<td>FARM</td>
<td>6 / 3</td>
<td>30 (261) / 15</td>
<td>One-side RDMA</td>
<td>4.5 / 10</td>
</tr>
<tr>
<td>DrTM-KV</td>
<td>115 / 14</td>
<td>500 (3972) / 60</td>
<td>One-side RDMA</td>
<td>3.4 / 6.3</td>
</tr>
<tr>
<td>HERD</td>
<td>35 / 25</td>
<td>490 / 300</td>
<td>Two-side RDMA</td>
<td>4 / 4</td>
</tr>
<tr>
<td>FPGA-Xilinx</td>
<td>14 / 14</td>
<td>106 / 106</td>
<td>Standalone FPGA</td>
<td>3.5 / 4.5</td>
</tr>
<tr>
<td>Mega-KV</td>
<td>166 / 80</td>
<td>330 / 160</td>
<td>GPU</td>
<td>280 / 280</td>
</tr>
<tr>
<td>KV-Direct (1 NIC)</td>
<td>180 / 114</td>
<td>1487 (5454) / 942 (3454)</td>
<td>Programmable NIC</td>
<td>4.3 / 5.4</td>
</tr>
<tr>
<td>KV-Direct (10 NICs)</td>
<td>1220 / 610</td>
<td>3417 (4518) / 1708 (2259)</td>
<td>Programmable NIC</td>
<td>4.3 / 5.4</td>
</tr>
</tbody>
</table>
New Milestone for KVS Performance

1.22 billion KV op/s
357 watts power
Takeaway for KV-Direct

KV-Direct leverage SmartNICS hardware to implement KV Store
• Solve the semantic mismatch of RDMA and KV access
• CPU is used only for control
• Push performance to the limit of hardware
• Achieve orders of magnitude power efficiency improvement
• Published in SOSP17

Many other workloads can be improved in a similar manner
• Such as DNN training, storage, NFV ......
• This is an emerging paradigm change in data center
Example 2: Socket for Communication

Socket is the universal communication primitive

- Designed half a century ago
- Complicated semantic
- Low performance, high overhead

Challenges:

- How to be fully compatible with native socket
- How to leverage modern hardware such as RDMA and multicore
- How to maintain isolation and security

Socket wastes CPU

Source: Luiz Barroso, Mike Marty, David Patterson, Parthasarathy Ranganathan.

Attack of the Killer Microseconds, Communications of the ACM, April 2017
SocksDirect: Fast and Compatible User Space Socket

Kernel Bypass with Dedicated Monitor Process
SocksDirect Supports Different Transports for Data

Host 1

Host 2

Host 3
No SocksDirect
SocksDirect Performance

Intra-host throughput

Intra-host latency

Inter-host throughput

Inter-host latency
Application End-to-End Performance with SocksDirect

Latency (µs, log scale)

Message size (Bytes)

64B 512B 4K 32K 256K 1M

Nginx HTTP End-to-End Latency
Takeaway for SocksDirect

SocksDirect implements a high-performance Socket system in user space:

- Bypass OS kernel to reduce overhead
- Dedicates a core for monitor process (take advantage of multi-core CPU)
- Can act as a drop-in replacement for classic socket without app modification
- Achieve orders of magnitude performance improvements
- Published in SIGCOMM19

Other traditional OS systems functions can be improved in a similar manner:

- Such as file systems
How is this related to OCP

Microsoft is contributing hardware innovations
• SmartNIC: FPGA enabled programmable NIC
• Zipline: Compression Engine based on FPGA
• Cerberus: Hardware root of trust
• Denali: Cloud optimized Flash storage

New hardware in Data Centers require us to rethink systems software design
• Which abstraction to use for non-volatile Storage Class Memory?
• Will CPU be relegated to become a second-class citizen in data center?
• How can we evolve the systems abstraction while keep software modification to a minimum?

Hardware and Systems software needs to evolve together
• We demonstrate two cases that can greatly improve systems performance by taking advantage of new hardware
Thank you