Comparison of Key/Value Store (KVS) in Software and Programmable Hardware

John W. Lockwood, CEO: Algo-Logic Systems, Inc.
Why Share Data by Name (Key) Instead of Address?

- **Key/Value Store (KVS)**
  - Simplifies implementation of large-scale distributed computation algorithms
  - Data Center Servers exchanges data over standard Ethernet

- **Challenges**
  - Operating System delays packets and limits throughput
  - Per-core processing inefficient at high-speed packet processing

- **Solutions**
  - Bypass kernel bypass with DPDK
  - Offload of packet processing with FPGA

---

### Examples:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Phone #</td>
</tr>
<tr>
<td>Algo-Logic</td>
<td>(408) 707-3740</td>
</tr>
<tr>
<td>IP Address</td>
<td>Interface : MAC Address</td>
</tr>
<tr>
<td>204.2.34.5</td>
<td>Eth6 : 02:33:29:F2:AB:CC</td>
</tr>
<tr>
<td>Content Hash</td>
<td>Storage Block ID</td>
</tr>
<tr>
<td>XYZ</td>
<td>948830038411</td>
</tr>
<tr>
<td>Order ID</td>
<td>Symbol, Side, Price</td>
</tr>
<tr>
<td>ATY11217911101</td>
<td>AAPL, B, 126.75</td>
</tr>
<tr>
<td>Virtex</td>
<td>Edge List</td>
</tr>
<tr>
<td>v140</td>
<td>v201, v206, v225</td>
</tr>
</tbody>
</table>
Why the Move to Programmable Hardware?

“There are large challenges in scaling the performance of software now. The question is: ‘What’s next?’ We took a bet on programmable hardware.”
- Doug Burger, Microsoft

• **Driving Metrics in the Data Center**
  
  – Latency:
    - Reduce delay
    - Avoid jitter
  
  – Throughput
    - Processing packets at line rate
    - Handle 10G, 25G, 40G, and 100G
  
  – Power:
    - Driving cost of OpEx

• Field Programmable Gate Array (FPGA) logic moves into the CPU
• Microsoft accelerates BING search with FPGA
• Intel acquires Altera
Servers Accelerated with FPGA Gateware

- **FPGA Augments Existing Servers**
  - Can run on an expansion card (same size as a GPU)
  - Or may be integrated into the CPU socket
- **GDN Applications run on FPGA**
  - Implements low-latency, low-power, high-throughput data processing
Implementation of KVS with Socket I/O, DPDK, and FPGA

- Benchmark same application
  - Key/Value Store (KVS)
- Running on the same PC
  - Intel i7-4770k CPU, 82598 NIC, and Altera Stratix V A7 FPGA
- With three different implementations
  - Socket I/O, DPDK, FPGA
Measured Latency, Throughput, and Power Results

### All Datapaths Summary

<table>
<thead>
<tr>
<th></th>
<th>Latency (µseconds)</th>
<th>Tested Throughput (CSMs/sec)</th>
<th>Power (µJoules/CSM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sockets</td>
<td>41.54</td>
<td>4.0</td>
<td>11</td>
</tr>
<tr>
<td>DPDK</td>
<td>6.434</td>
<td>16</td>
<td>6.6</td>
</tr>
<tr>
<td>RTL</td>
<td>0.467</td>
<td>15</td>
<td>0.52</td>
</tr>
</tbody>
</table>

### GDN vs. Sockets

GDN vs. Sockets 88x less 13x 21x less

### GDN vs. DPDK

GDN vs. DPDK 14x less 3.2x 13x less
KVS Latency in FPGA, DPDK, and Sockets

Latency Comparison 100k packets, 1 OCSM per packet, 1k pps

- KVS in FPGA: Best Latency, No Jitter
  - Altera Stratix V RTL Average: 0.467µs
- KVS in DPDK: Lower Latency, Some Jitter
  - DPDK Average: 6.29µs
- Sockets Average: 41.40µs

Tighter Spread = Less Jitter

Lower Latency = Faster Response
Conclusions: Key/Value Store in Programmable Hardware

- **Lowers Latency**
  - 88x faster than Linux networking sockets
  - 14x faster than optimized DPDK (kernel bypass)

- **Increases Throughput (IOPs)**
  - 3x to 13x improvement in throughput
  - Lowers Capital Expenditures (CapEx)

- **Reduces Power**
  - 13x to 21x reduction in power
  - Reduces Operating Expenditures (OpEx)

*Gateware Defined Networking® dramatically reduces latency and power and improves throughput in the data center*