Direct Universal Access: Making Data Center Resources Available to FPGA

Ran Shu\textsuperscript{1}, Peng Cheng\textsuperscript{1}, Guo Chen\textsuperscript{2}, Zhiyuan Guo\textsuperscript{1,3}, Lei Qu\textsuperscript{1}, Yongqiang Xiong\textsuperscript{1}, Derek Chiou\textsuperscript{4}, Thomas Moscibroda\textsuperscript{4}

Microsoft Research\textsuperscript{1}, Hunan University\textsuperscript{2}, Beihang University\textsuperscript{3}, Microsoft Azure\textsuperscript{4}
FPGA Deployment in Data Centers

- **Wide deployment**
  - Major cloud service providers
    - Microsoft, Amazon, Facebook, Alibaba, Tencent, Baidu, IBM, etc.

- **Accelerated applications**
  - **Computation**
    - Web search ranking
    - Deep neural networks
    - Big data analytics
  - **Networking**
    - Network processing
  - **Database/Storage**
    - SQL
    - Key-value store

Image from A. Caulfield et al., Micro 2016

Image from D. Firestone et al., NSDI 2018
Resource Access Requirements

- Heterogeneous resources
  - CPU
  - Memory
  - Other FPGAs
  - GPU
  - SSD
FPGA Board in Data Center

Current FPGA Communication Architecture

- Application Layer
- Transport Layer
- Data Link Layer
- Physical Layer

Diagram:
- FPGA Board
- FPGA Chip
- DDR 4
- DDR 4 IP
- QSFP
- QSFP IP
- PCIe Gen 3
- PCIe Gen 3 IP
- LTL
- DMA
- NVMe
- DCQCN (in LTL)
Current FPGA Communication Architecture

Image from L. Zhang et al., CCR 2014
Problem #1 – Programming Interface

Lines of code to use each interface

- Host DRAM: 294
- Host CPU program: 205
- Onboard DRAM: 517
- Remote FPGA: 1356

FPGA Chip

- Application
- DDR Stack
- LTL
- Host DMA
- NVMe Stack

- DDR IP
- Ethernet IP
- PCIe IP
## Problem #2 – Accessibility

<table>
<thead>
<tr>
<th>FPGA Chip</th>
<th>Application</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR Stack</td>
<td>LTL</td>
<td>Host DMA</td>
</tr>
<tr>
<td>NVMe Stack</td>
<td></td>
<td>PCIe IP</td>
</tr>
</tbody>
</table>

- **On-board naming space**
  - DDR IP

- **Data center naming space**
  - Ethernet IP

- **Server-area naming space**
  - NVMe Stack
Problem #3 – Multiplexing

FPGA Chip

- Application 1
- Application 2
- Host DMA Mux/Demux

- DDR Stack
- LTL
- Host DMA
- NVMe Stack

- DDR IP
- Ethernet IP
- PCIe IP
Problem #3 – Multiplexing

FPGA Chip

Application 1

LTL

Mux/Demux

Application 2

DDR Stack

LTL

Host DMA

NVMe Stack

DDR IP

Ethernet IP

PCIe IP
Problem #3 – Multiplexing

FPGA Chip

- Application 1
  - DDR Stack
  - LTL
  - DDR IP
- Application 2
  - Host DMA
  - NVMe Stack
  - PCIe IP
  - PCIe Mux/Demux
  - Ethernet IP
Problem #4 – Security

Malicious Application

Access unauthorized address

Host Memory

PCle Fabric
Problem #4 – Security

FPGA Chip

- DDR Stack
- LTL
- Host DMA
- NVMe Stack

Victim Application

- DDR IP
- Ethernet IP
- PCIe IP

Malicious CPU Program

Attack application

PCIe Fabric
Existing Problems

- Complex programming interface
- Separate naming space
- No general multiplexing
- Security issue
Direct Universal Access

Server
FPGA 1
DUA
DDR access
DDR
QSFP
Connect
Host DMA

PCIe Gen3
CPU

DUA
App 1

Intra-server networking fabric

①

Server
FPGA 2
DUA
DDR access
DDR
QSFP
Connect
Host DMA

PCIe Gen3
CPU

App 1

②

Server
FPGA 3
DUA
DDR access
DDR
QSFP
Connect
Host DMA

PCIe Gen3
CPU

App 2

③

Datacenter networking fabric

④
DUA Overview

DUA is an “IP layer”
An abstract overlay network
Leverage all existing h/w stacks
Hierarchical addressing & routing

DUA is an abstract overlay network that leverages all existing hardware stacks and supports hierarchical addressing and routing.
DUA Overview

Efficient Routing
Direct resource access by FPGA, totally bypass CPU
DUA Overview

DUA is an “IP layer”

Efficient Routing

Compatible BSD-socket Interface
for both applications and communication stacks

Intra-server networking fabric

Datacenter networking fabric
DUA Overview

DUA is an “IP layer”

Efficient Routing

Compatible BSD-socket Interface

General Multiplexing
for both applications and communication stacks

Intra-server networking fabric

Datacenter networking fabric
DUA Overview

DUA is an “IP layer”

Efficient Routing

Compatible BSD-socket Interface

Unified Multiplexing

Security

Protect against both inside and outside attacks
System Architecture

Datacenter networking fabric

Intra-server networking fabric

DUA data plane
System Architecture

- DUA Data Plane
  - DDR Controller
  - Host DMA
  - FPGA Host Stack
  - NVMe Stack
  - FPGA Connect
  - LTL

- DUA Underlay
  - FPGA Control Agent
  - DDR
  - QSFP
  - PCIe Gen3

- Intra-server networking fabric
  - NIC
  - FPGAs
  - CPU Control Agent

- Datacenter networking fabric
  - NIC
  - FPGA
  - CPU
DUA Control Plane

• Challenge: large-scale resource and routing info dissemination
  – Limited h/w resource

• DUA solution
  – Hierarchical addressing
  – Hierarchical routing
  – Leverage existing infrastructure

• Fully distributed and lightweight
  – Need no global synchronization

<table>
<thead>
<tr>
<th>UID (serverID:deviceID)</th>
<th>Address /port</th>
<th>Resource description</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.2:1</td>
<td>0x00000001CFFFF000</td>
<td>1st block of host DRAM</td>
</tr>
<tr>
<td>192.168.0.2:1</td>
<td>0x000000019FFFF000</td>
<td>2nd block of host DRAM</td>
</tr>
<tr>
<td>192.168.0.2:2</td>
<td>0x80000000</td>
<td>1st block of FPGA onboard</td>
</tr>
<tr>
<td>192.168.0.2:3</td>
<td>8000</td>
<td>1st application on FPGA</td>
</tr>
<tr>
<td>192.168.0.2:3</td>
<td>8001</td>
<td>2nd application on FPGA</td>
</tr>
</tbody>
</table>

Resource table

<table>
<thead>
<tr>
<th>Src Resource (UID)</th>
<th>Dst Resource (UID) / Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA 1 (192.168.0.2:1)</td>
<td>FPGA 2 (192.168.0.2:2) / FPGA Connect</td>
</tr>
<tr>
<td></td>
<td>Host DRAM (192.168.0.2:3) / DMA</td>
</tr>
<tr>
<td></td>
<td>Onboard DRAM (192.168.0.2:4) / DDR</td>
</tr>
<tr>
<td>FPGA 2 (192.168.0.2:2)</td>
<td>FPGA 1 (192.168.0.2:1) / FPGA Connect</td>
</tr>
<tr>
<td></td>
<td>Host DRAM (192.168.0.2:3) / DMA</td>
</tr>
<tr>
<td></td>
<td>Resources on other servers (<em>/</em>) / LTL</td>
</tr>
</tbody>
</table>
DUA Data Plane

• Overlay
  – Unified interface
  – Routing

• Stacks
  – Leverage all the existing (or adopt future) stacks

• Underlay
  – Efficient multiplexing
  – Security
DUA Data Plane – Overlay

• Efficient & extensible design
  – Switch fabric
    • High capacity cross-bar switch
  – Connector
    • All cached routing tables
  – Translator
    • Protocol translation

• High performance data path
  – Line-rate
  – Near zero-delay
Evaluation – efficiency

Extreme low latency (< 50 ns/fwd)

Round Trip Time through FPGA Connect and DUA for 4 times, LTL twice

<table>
<thead>
<tr>
<th>Packet Size (B)</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency (us)</td>
<td>2.444</td>
<td>2.479</td>
<td>2.545</td>
</tr>
<tr>
<td>DUA</td>
<td>1.266</td>
<td>1.266</td>
<td>1.316</td>
</tr>
<tr>
<td>LTL</td>
<td>0.153</td>
<td>0.176</td>
<td>0.196</td>
</tr>
<tr>
<td>FPGA Connect</td>
<td>1.266</td>
<td>1.266</td>
<td>1.316</td>
</tr>
</tbody>
</table>
Evaluation – Logic Overhead

<table>
<thead>
<tr>
<th>Component</th>
<th>ALMs</th>
<th>DUA Overlay</th>
<th>DUA Underlay</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Switch fabric</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 ports</td>
<td>1272</td>
<td>0.74%</td>
<td></td>
</tr>
<tr>
<td>4 ports</td>
<td>3227</td>
<td>1.88%</td>
<td></td>
</tr>
<tr>
<td>8 ports</td>
<td>9366</td>
<td>5.45%</td>
<td></td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>3011</td>
<td>1.75%</td>
<td></td>
</tr>
<tr>
<td><strong>Stack translator</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPGA Connect</td>
<td>138.4</td>
<td>0.08%</td>
<td></td>
</tr>
<tr>
<td>LTL</td>
<td>255.4</td>
<td>0.15%</td>
<td></td>
</tr>
<tr>
<td>DMA</td>
<td>115.7</td>
<td>0.07%</td>
<td></td>
</tr>
<tr>
<td>DDR</td>
<td>190.3</td>
<td>0.11%</td>
<td></td>
</tr>
<tr>
<td><strong>Stacks: FPGA Connect, LTL, DMA, DDR</strong></td>
<td>431.7</td>
<td>0.25%</td>
<td></td>
</tr>
<tr>
<td><strong>PHY interfaces:</strong> PCIe, DDR, QSFP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Stack</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPGA Connect</td>
<td>620.8</td>
<td>0.36%</td>
<td></td>
</tr>
<tr>
<td>LTL</td>
<td>6395.4</td>
<td>3.72%</td>
<td></td>
</tr>
<tr>
<td>DMA</td>
<td>1347.7</td>
<td>0.78%</td>
<td></td>
</tr>
<tr>
<td>DDR</td>
<td>73.4</td>
<td>0.04%</td>
<td></td>
</tr>
<tr>
<td><strong>PHY interfaces</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCIe</td>
<td>3890.1</td>
<td>2.26%</td>
<td></td>
</tr>
<tr>
<td>QSFP</td>
<td>12726.7</td>
<td>7.40%</td>
<td></td>
</tr>
<tr>
<td>DDR</td>
<td>7369.2</td>
<td>4.28%</td>
<td></td>
</tr>
</tbody>
</table>
**Evaluation – Deep Crossing**

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parall = 32</td>
<td>7.27</td>
<td>6.58</td>
<td>13.30</td>
<td>12.96</td>
<td>40.12</td>
</tr>
<tr>
<td>Parall = 64</td>
<td>4.17</td>
<td>3.48</td>
<td>7.22</td>
<td>7.09</td>
<td>21.95</td>
</tr>
<tr>
<td>Reduction(%)</td>
<td>42.67</td>
<td>47.10</td>
<td>45.75</td>
<td>45.33</td>
<td>45.28</td>
</tr>
</tbody>
</table>

*Single FPGA Board: Parall = 32, 2 FPGA Board: Parall = 64*

**45.28% Latency Reduction**
Evaluation – Regex Matching

- Up to $10^5$ to $10^7$ higher than CPU, Up to $10^5$ lower than CPU
- Up to 3 times throughput and up to 55% latency reduction compared to using CPU to move data between FPGAs
Conclusion

• Current FPGA communication architecture
  – No universal access
• DUA: build the “IP” layer for FPGA in data center
  – Leverage existing data center network
  – Efficient routing
  – Compatible BSD socket interface
  – Unified multiplexing
  – Security
• Open source soon
Thank you!
Questions?