A Heterogeneous Stack for Deploying Clusters

Naif Trafaldar and Paul Chow

High-Performance Reconfigurable Computing Group
Department of Electrical and Computer Engineering
University of Toronto
The Holy Grail

Distributed Flow Diagram

Heterogeneous Data Center

A

B

C

D

A

B

C

D

September 10, 2019
Data Centers are Heterogeneous

- Take advantage of unique resources in data center
- Need back doors or abstractions to communicate between different devices
- Scaling this up is difficult as different devices have different programming models, abstractions etc.
- Communicating across these different devices even more complicated
Contributions

• Built a heterogeneous communication layer across multiple devices

• Built on top of heterogeneous stack
  – Allows to communicate to any device that is abstracted in stack

• Functional portability allows the prototyping of cluster in software before deploying on accelerator
BACKGROUND: GALAPAGOS STACK
Heterogeneous Abstraction Stack

Heterogeneous Communication

Hardware Stack

Communication Layer
Middleware/Network Layer
Hypervisor Layer
Physical Hardware

Software Stack

Communication Layer
Orchestration/Network Layer
OS/Hypervisor Layer
Physical Hardware

Galapagos
Heterogeneous Abstraction Stack

Heterogeneous Communication

Hardware Stack
- Communication Layer
- Middleware/Network Layer
- Hypervisor Layer
- Physical Hardware

Software Stack
- Communication Layer
- Orchestration/Network Layer
- OS/Hypervisor Layer
- Physical Hardware

Physical Hardware
Physical Hardware

• Refers to the physical devices and how they are connected
  – E.g: FPGA, CPU, Sensors
  – Wireless, Infiniband

• Example: Nexys 4 Board, Pynq Board, Laptop attached via Ethernet cables to 1G network
Heterogeneous Multi-Tier Cloud

EDGE 1
FPGA
GPU
CPU
EDGE Router

... ...

EDGE 8
FPGA
IOT Sensor
CPU
EDGE Router

Network Backbone

CORE
CPU Servers

September 10, 2019
Heterogeneous Multi-Tier Cloud

EDGE 1

FPGA
GPU
CPU

EDGE Router

EDGE 2

... 

EDGE 8

FPGA
IOT Sensor
CPU

EDGE Router

Network Backbone

CORE

CPU
Servers
Heterogeneous Multi-Tier Cloud

EDGE 1

- FPGA
- GPU
- CPU
- EDGE Router

... 

EDGE 8

- FPGA
- IOT Sensor
- CPU
- EDGE Router

Network Backbone

CORE

CPU Servers
Heterogeneous Multi-Tier Cloud

EDGE 1

FPGA  
GPU
CPU

EDGE Router

...  

EDGE 8

FPGA
IOT Sensor
CPU

EDGE Router

Network Backbone

CORE

CPU Servers

September 10, 2019
Heterogeneous Multi-Tier Cloud

EDGE 1

- FPGA
- GPU
- CPU

EDGE Router

... ... ...

EDGE 8

- FPGA
- IOT Sensor
- CPU

EDGE Router

Network Backbone

CORE

- CPU Servers

September 10, 2019
Our Datacenters

MPSoC Boards Connected via 100 GB/s Cables
~$100,000
September 10, 2019

Nexys, Pynq, Laptop, 1G switch
~$2000
Heterogeneous Abstraction Stack

Heterogeneous Communication

Hardware Stack

- Communication Layer
- Middleware/Network Layer
- Hypervisor Layer
- Physical Hardware

Software Stack

- Communication Layer
- Orchestration/Network Layer
- OS/Hypervisor Layer
- Physical Hardware

September 10, 2019
Galapagos: Hypervisor

- Abstracts device into a streaming device
- All devices with hypervisor now looks like the same streaming device
Heterogeneous Abstraction Stack

- Communication Layer
- Middleware/Network Layer
- Hypervisor Layer
- Physical Hardware

- Communication Layer
- Orchestration/Network Layer
- OS/Hypervisor Layer
- Physical Hardware

Heterogeneous Communication

Hardware Stack

Software Stack
Galapagos: Middleware

- Allows streaming kernels independent of placement and implementation to communicate to any other kernel within a heterogeneous cluster
- This layer refers to how we orchestrate clusters of resources
  - Includes FPGAs and CPUs
- Orchestration includes automating the connections between resources and providing handle to entire cluster
Galapagos: Middleware Layer

- User can define a FPGA cluster using cluster description files and AXI-Stream kernels
Galapagos: Middleware Layer

- User can define a FPGA cluster using cluster description files and AXI-Stream kernels
Galapagos: Middleware Layer

- User can define a FPGA cluster using cluster description files and AXI-Stream kernels.
Galapagos: Middleware Layer

- User can define a FPGA cluster using cluster description files and AXI-Stream kernels
Galapagos: Middleware IP Blocks

Middleware generates additional IP blocks
Galapagos: Middleware IP Blocks

Connects to user provided kernels within application region of hypervisor
Galapagos: Middleware IP Blocks

Network Bridge

- Translates network packets into AXI stream packets
- Current support for TCP and L2 ethernet
Router

- Routes packets on chip and off-chip
- Routing table maps AXI dest to FPGA network address
Galapagos: 
Middleware IP Blocks

Connects to user provided kernels within application region of hypervisor
Communication Layer: libGalapagos

- Create software model of each component
- Galapagos software kernel object wrapper for HLS module
  - Functionally portable, uses same HLS code for software
- Kernels are spawned as individual threads
Communication Layer: libGalapagos

- Create software model of each component
- Galapagos software kernel object wrapper for HLS module
  - Functionally portable, uses same HLS code for software
- Kernels are spawned as individual threads
HETEROGENEOUS MACHINE LEARNING PLATFORM
Single Node Inference Engine

- Xilinx CNN Kernel
- Kernel Instruction provides layer information and addresses for input, weights and memory
- Not streaming based, can’t connect to cluster
Single Node Inference Engine

- Xilinx CNN Kernel
- Kernel Instruction provides layer information and addresses for input, weights and memory
- Not streaming based, can’t connect to cluster
Single Node Inference Engine

- Xilinx CNN Kernel
- Kernel Instruction provides layer information and addresses for input, weights and memory
- Not streaming based, can’t connect to cluster
Single Node Inference Engine

- Controller written in HLS
- Stream in/stream out
- Streams in insns for CNN kernel and DMA instructions for memory
Controller written in HLS
Stream in/stream out
Streams in insns for CNN kernel and DMA instructions for memory
Single Node Inference Engine

- Controller written in HLS
- Stream in/stream out
- Streams in instructions for CNN kernel and DMA instructions for memory
Single Node Inference Compiler

- Using neural net graph description generate the instructions for CNN kernel and DMA addresses
- Our compiler is called Telepathy
Telepathy Details

- Telepathy responsible for partitioning neural and keeping track of addresses of all nodes in cluster
  - Agnostic to CNN kernel implementation
- Generates instructions
  - Specific to CNN kernel implementation
Multi-Node Inference

Telepathy sends instructions and DMA addresses to individual Nodes
Multi-Node Inference

Telepathy sends instructions and DMA addresses to individual Nodes.

Telepathy Instruction DMA Address CNN Kernel Mem

Custom RDMA Controller

CNN Kernel Mem

Custom RDMA Controller

September 10, 2019
Multi-Node Inference

Telepathy sends instructions and DMA addresses to individual Nodes.
Multi-Node Inference

Telepathy sends instructions and DMA addresses to individual Nodes

September 10, 2019
Multi-Node Inference

Telepathy sends instructions and DMA addresses to individual Nodes

Telepathy

Network

DMA Address

Instruction

CNN Kernel

Mem

DMA Address

Instruction

CNN Kernel

Mem

September 10, 2019
Multi-Node Inference

Telepathy sends instructions and DMA addresses to individual Nodes.
Multi-Node Inference

First frame sent from Telepathy to first Node via RDMA and processed.

Telepathy -> Frame -> Network -> Custom RDMA Controller -> Mem

Custom RDMA Controller -> CNN Kernel -> Mem

Custom RDMA Controller -> CNN Kernel
Multi-Node Inference

First frame sent from Telepathy to first Node via RDMA and processed

September 10, 2019
Multi-Node Inference

First frame sent from Telepathy to first Node via RDMA and processed.
Multi-Node Inference

First frame sent from Telepathy to first Node via RDMA and processed
Multi-Node Inference

Processed frame sent to next Node

Telepathy

Network

Custom RDMA Controller

Mem

Frame

Custom RDMA Controller

CNN Kernel

Mem

September 10, 2019
Multi-Node Inference

Processed frame sent to next Node

Telepathy

Network

Frame

CNN Kernel

Mem

Custom RDMA Controller

CNN Kernel

Mem

September 10, 2019
Multi-Node Inference

Processed frame sent to next Node

Telepathy

Frame

Custom RDMA Controller

Mem

CNN Kernel

Custom RDMA Controller

Mem

CNN Kernel

September 10, 2019
Multi-Node Inference

Processed frame sent to next Node

Telepathy

Network

Frame

Custom RDMA Controller

CNN Kernel

Mem

Custom RDMA Controller

CNN Kernel

Mem

September 10, 2019
Multi-Node Inference

Processed frame sent to next Node

Telepathy

Network

Custom RDMA Controller

Mem

CNN Kernel

Mem

Frame

CNN Kernel
Multi-Node Inference

Processed frame sent to next Node

Telepathy

Network

Custom RDMA Controller

CNN Kernel

Mem

Frame

CNN Kernel

Custom RDMA Controller

September 10, 2019
Different Use Case, Different Kernel

- The CNN kernel can be swapped with other CNN kernel implementations as long as Telepathy’s instruction generator is modified.
Different Use Case, Different Kernel

- The CNN kernel can be swapped with other CNN kernel implementations as long as Telepathy’s instruction generator is modified
Current ML Applications

- Currently in collaboration with HLS4ML team use their kernel as a Galapagos kernel
- Second ML app using Xilinx kernel for data center workloads
- Same abstraction layers!
Current ML Applications

• Currently in collaboration with HLS4ML team use their kernel as a Galapagos kernel
• Second ML app using Xilinx kernel for data center work loads
• Same abstraction layers!
Current ML Applications

- Currently in collaboration with HLS4ML team use their kernel as a Galapagos kernel
- Second ML app using Xilinx kernel for data center workloads
- Same abstraction layers!
CONCLUDING REMARKS
Conclusion

• Created heterogeneous abstraction layer stack
  • Makes it easy to connect different types of devices
  • Makes formation of clusters more modular
Conclusion

• Created heterogeneous abstraction layer stack
• Makes it easy to connect different types of devices
  • Makes formation of clusters more modular
Conclusion

- Created heterogeneous abstraction layer stack
- Makes it easy to connect different types of devices
- Makes formation of clusters more modular
Thanks!

- Xilinx
- NSERC
- Huawei
Questions?

Email: naif.Tarafdar@mail.utoronto.ca