Algean: An Open Framework for Machine Learning on a Heterogeneous Cluster

Naif Tarafdar¹, Giuseppe Di Guglielmo², Philip C Harris³, Jeffrey D Krupa³, Vladimir Loncar⁴, Dylan S Rankin³, Nhan Tran⁵, Zhenbin Wu⁶, Qianfeng Shen¹ and Paul Chow¹

University of Toronto¹
Columbia University²
Massachusetts Institute of Technology³
CERN⁴
Fermilab⁵
University of Illinois⁶
Take Aways

• Galapagos: Platform for multi-FPGA application deployment
  – A scalable giant FPGA comprised of individual FPGAs

• Algean: Mapping an ML application onto the giant FPGA
  – Could also be your own applications

• Depending on your area of expertise and interest you can use different parts of this project
Machine Learning

• One of the most popular topics of research
  – In many areas, many applications (e.g. medical, financial, safety, transportation etc.)
  – Also within the computing community

• Wide usage in world pushes limits of devices
  – Metrics include performance and energy
  – Leading many researchers to consider heterogeneity!
Heterogeneity All Around Us

Snapdragon 630 Mobile Platform

This Photo by Unknown author is licensed under CC BY-NC.

This Photo by Unknown author is licensed under CC BY-SA-NC.

This Photo by Unknown author is licensed under CC BY-NC-ND.
Applying Machine Learning to a Heterogeneous Environment

• Challenge: How do you design machine learning algorithms for a heterogeneous space?
  – Hard enough with a homogenous computing environment
  – Is there a framework for such a thing?

• Challenge: If such a framework exists can we get both flexibility and performance?
Outline

• Brief Motivation

• Overview of machine learning frameworks
  – Categorized as an abstraction layer stack

• Overview of Algean
  – HLS4ML
  – Galapagos

• Results
MACHINE LEARNING FRAMEWORKS
Many Popular Examples!

• Such as
  – Tensorflow
  – PyTorch
  – Caffe
  – Intel DLA
  – Xilinx XfDNN

• What do these different frameworks offer?
  – Depends on who you ask!
# Machine Learning Stack

<table>
<thead>
<tr>
<th>Applications &amp; Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Deployment &amp; Communication</td>
</tr>
<tr>
<td>Hardware</td>
</tr>
</tbody>
</table>
Machine Learning Stack

- Applications & Algorithms
  - E.g: Neural net layers, quantization, compression, pruning
- Cluster Deployment & Communication
- Hardware
Machine Learning Stack

Applications & Algorithms

Cluster Deployment & Communication
- E.g: Physical Connections (PCIe, ethernet etc.),
- Communication Protocols

Hardware
Machine Learning Stack

Applications & Algorithms

Cluster Deployment & Communication

Hardware

E.g: Hardware circuit (multipliers, shifters), memory architecture (caching etc.)
Machine Learning Stack

- Allows researchers to pick and choose layers they wish to configure
- Collapsible/Expandable for specific application and infrastructure!
AIGean Introduction

• Like the archipelago and sea
• Combines two existing frameworks:
  – HLS4ML:
    • HLS IP cores of ML IP
  – Galapagos
    • Connects and deploys heterogeneous distributed application across multiple nodes
HLS4ML

• Open source project

• Input:
  – Description of FPGA resources
    • LUT, BRAM, DSP
  – Description of neural net
    • PyTorch, Keras, Onyx support

• Output:
  – HLS synthesizable C++ that fits within resource constraints implementing neural net

• Tunable HLS code, made to fit the FPGA
Galapagos

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

- User can define a FPGA cluster using cluster description files and AXI-Stream kernels

November 13, 2020

H2RC 2020
Galapagos

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

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

- Heterogeneous Stack
- Allows users to create flexible heterogeneous clusters across CPUs/FPGAs
- Seamlessly prototype by implementing both on CPU and FPGA
  - Galapagos ensures functional portability for network communication
  - Essentially "network-connected" HLS kernels
    - For both SW and HW
  - Iterative development, selectively move bottleneck from SW to hardware without modifying code
- Flexibly change communication protocol without modifying user application
  - TCP, UDP, L1 etc
  - User application is agnostic to this
Birth of Algean

- HLS4ML creates HLS IP core to maximize FPGA utilization
- Galapagos can give a multi-FPGA fabric
- Tools combined to deploy neural-net on multi-FPGA Fabric
Algean Tool Flow

Model Training
Keras, PyTorch

Model
HDF5 & JSON

Tuning

AlGean Automated Flow

CPU/FPGA Cluster

Not connected

ML to Galapagos Bridge

C++ & TCL

Partitioner
IP Cluster

November 13, 2020
H2RC 2020
Algean Tool Flow

Model Training
- Keras, PyTorch

Model
- HDFS & JSON

Hls4ml
- C++ & TCL

HLS ML Layers
- C++ & TCL

Partitioner

ML2G

CPU/FPGA Cluster

Not connected

IP Cluster

HLS ML to Galapagos Bridge
- C++ & TCL

AlGean Automated Flow
HLS4ML Modifications

- HLS4ML modified to create independent layers as separate HLS IP cores
  - Each IP core is a streaming core with each stream per dimension of the particular layer
HLS4ML Galapagagos Bridge

- Bridges custom made for the layers used in the network (different bridges needed for different number of dimensions)
- If the user has a different application layer then they would need a different bridge
Algean Tool Flow

Model Training
- Keras, PyTorch

Model
- HDF5 & JSON

Hls4ml
- C++ & TCL

HLS ML Layers
- C++ & TCL

HLS ML to Galapagos Bridge
- C++ & TCL

Partitioner

IP Cluster
- Not connected

ML2G

Tuning

CPU/FPGA Cluster
Partitioner

- Partitioner separates IP cores onto different FPGAs
- Currently using IP resources estimation from HLS Place and route and performing simple greedy approach
- Does not place the bridges as that is Algean specific, and this partitioner is general for all Galapagos IP kernels
Algean Tool Flow

Model Training
Keras, PyTorch

Model
HDF5 & JSON

Tuning

Hls4ml
C++ & TCL

HLS ML Layers

Partitioner

IP Cluster
Not connected

ML2G

HLS ML to Galapagos Bridge
C++ & TCL

AlGean
Automated Flow

CPU/FPGA
Cluster

November 13, 2020
H2RC 2020
Machine Learning to Galapagos (ML2G)

- Adds the appropriate bridges on the interfaces of the FPGAs
- Creates the local connections for kernels on the same FPGA
RESULTS
Experiment Setup

• CPUs
  – Xeon E5-2650
    • 24 Cores at 2.2 GHz

• FPGAs
  – Fidus Sidewinder
    • ZU19EG FPGA
      – ~1 Million logic cells, 35 MB BRAM, 1968 DSP slices
    • 100 GB network interface
  – 100 GB UDP core
Microbenchmarks

- Latency send single flit
- Throughput: maximum throughput of link (varying packet size for software)

<table>
<thead>
<tr>
<th>Link</th>
<th>Latency</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software to Hardware</td>
<td>0.029 ms</td>
<td>0.244 GB/s</td>
</tr>
<tr>
<td>Hardware to Hardware</td>
<td>0.00017 ms</td>
<td>100 GB/s</td>
</tr>
<tr>
<td>Hardware to Software</td>
<td>0.0203 ms</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Microbenchmarks

- Larger the packet, higher the throughput.
- UDP packet size limited
  - No segmentation
  - MTU size
  - Jumbo Frames: 8K

<table>
<thead>
<tr>
<th>Link</th>
<th>Latency</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software to Hardware</td>
<td>0.029 ms</td>
<td>0.244 GB/s</td>
</tr>
<tr>
<td>Hardware to Hardware</td>
<td>0.00017 ms</td>
<td>100 GB/s</td>
</tr>
<tr>
<td>Hardware to Software</td>
<td>0.0203 ms</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Microbenchmarks

- Line-rate, same throughput at small and large packet size

<table>
<thead>
<tr>
<th>Link</th>
<th>Latency</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software to Hardware</td>
<td>0.029 ms</td>
<td>0.244 GB/s</td>
</tr>
<tr>
<td>Hardware to Hardware</td>
<td>0.00017 ms</td>
<td>100 GB/s</td>
</tr>
<tr>
<td>Hardware to Software</td>
<td>0.0203 ms</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Microbenchmarks

- HW at line-rate
- UDP, SW can't keep up and we see packet drop

<table>
<thead>
<tr>
<th>Link</th>
<th>Latency</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software to Hardware</td>
<td>0.029 ms</td>
<td>0.244 GB/s</td>
</tr>
<tr>
<td>Hardware to Hardware</td>
<td>0.00017 ms</td>
<td>100 GB/s</td>
</tr>
<tr>
<td>Hardware to Software</td>
<td>0.0203 ms</td>
<td>N/A</td>
</tr>
</tbody>
</table>

November 13, 2020
H2RC 2020
Small Neural Network: Results

• Single CPU, single FPGA, used in physics application to calculate energy of a particle
• 16K inferences
• SDAccel (without Algean) 3 ms
• Algean 6.3 ms
  – Latency of single inference 0.08 ms, we can do this since streaming, not possible via SDAccel
• Bottleneck: Sending data to FPGA via CPU network link
Small Neural Network: Takeaway

• Comparison vs SDAccel shows that network link for a single FPGA can be competitive with PCIe
  – Network link wins in terms of scalability, many more available FPGAs via network vs PCIe

• Can stream data
  – Latency of single inference a lot faster

• Should target larger application
  – We can do this as we have a large multi-FPGA fabric!
Autoencoder: Results

• Autoencoder implemented in both SDAccel on single FPGA and Algean using 3 FPGAs

• SDAccel: Single FPGA, higher reuse factor to fit logic
  – 0.26 ms

• Algean: Three FPGAs
  – 0.08 ms, more than 3x improvement
Autoencoder: Takeaway

- Using a larger fabric allows us to implement larger circuits
- The difficulty of communication between multi-FPGA is abstracted away
ResNet-50

• Currently IP cores implemented at 6600 images/second (slightly better than Brainwave)
• Prototype in software working
• Bridges working at line rate
• 12 FPGA bitstreams currently being synthesized and tested
• In the pipeline: 30000 images/second
SUMMARY AND CONCLUSION
Summary

• Multi-FPGA/CPU neural net framework by leveraging and combining HLS4ML and Galapagos frameworks
• Tunable IP cores, flexible communication
• ML HLS IP cores deployed onto cluster of network connected FPGAs and CPUs
• Communication abstracted away from user

Model Training
Keras, PyTorch

AlGean Automated Flow

Tuning

CPU/FPGA Cluster

November 13, 2020

H2RC 2020
Conclusions

- Network connected FPGAs/CPUs are more scalable than traditional PCIe
- Creation of larger fabrics with network connected FPGAs opens door for more complex algorithms
- Many opportunities to explore in multi-FPGA ML
- Galapagos provides a good foundation for multi-FPGA applications
Acknowledgments

[Logos of Xilinx, CMC Microsystems, NSF, NSERC, Huawei, and iris hep]
Thank You

- Email: pc@eecg.toronto.edu