Machine Learning Across Network-Connected FPGAs

Dana Diaconu
Northeastern University
Boston, USA
diaconu.d@northeastern.edu

Yanyue Xie
Northeastern University
Boston, USA
xie.yany@northeastern.edu

Mehmet Gungor
Northeastern University
Boston, USA
gungor.m@northeastern.edu

Suranga Handagala
Northeastern University
Boston, USA
s.handagala@northeastern.edu

Xue Lin
Northeastern University
Boston, USA
xue.lin@northeastern.edu

Miriam Leeser
Northeastern University
Boston, USA
mel@coe.northeastern.edu

***Abstract*—FPGAs often cannot implement machine learning inference with high accuracy models due to significant storage and computing requirements. The corresponding hardware accelerators of such models are large designs which cannot be deployed on a single platform. In this research, we implement ResNet-50 with 4 bit precision for weights and 5 bit precision for activations, which has a good trade-off between precision and accuracy. We train ResNet-50 using the quantization-aware training library Brevitas and build a hardware accelerator with the FINN framework from AMD. We map the result to three FPGAs that communicate directly with one another over the network via the User Datagram Protocol (UDP). The multi-FPGA implementation is compared to a single FPGA ResNet-50 design with lower precision of 1 bit weights and 2 bit activations. While the latter can fit on a single FPGA, the former pays for higher accuracy with a three times increase in the required number of BRAM tiles and can only be deployed on multiple FPGAs. We show the difference in accuracy, resource utilization, and throughput for the designs deployed on AMD/Xilinx Alveo U280 data center accelerator cards available in the Open Cloud Testbed (OCT). The final multi-FPGA custom accelerator design for ResNet-50 achieves a 5.3% increase in accuracy and a throughput of 162.3 images/s at a frequency of 200 MHz, comparable to the single FPGA lower precision implementation's throughput of 176.1 images/s at 160 MHz. We further explore a more efficient usage of the available memory on the target platform. By making use of the available Ultra RAM, we are able to fit the accelerator with higher precision on one U280 and achieve a throughput of 165 images/s.**

***Index Terms*—FPGA, Machine Learning, FINN, ResNet-50, Quantization**

I. Introduction

The architectures of Machine Learning (ML) models evolve at a fast pace and their complexity grows significantly with their increase in performance. ML is applied in numerous applications but its deployment on edge devices and in data centers is becoming gradually more difficult. Hardware accelerators need to be adapted frequently to changes in the models and at the same time to be implementable on the available hardware or cloud structure. For this reason, researchers try to find the best balance between accuracy, throughput, power consumption, and hardware footprint based on the application requirements.

FPGAs represent a low power, flexible platform on which low latency and high throughput accelerators can be implemented [1], [2]. However, efficient hardware accelerators for ML, and specifically Deep Neural Networks (DNN), are feasible only through software-hardware co-design. High accuracy can be obtained in software using floating point precision, however floating point precision requires a large amount of hardware computational resources and is not usually employed on FPGA platforms. The accelerator design is constrained by the available resources on the target platform and the throughput can only be increased by reducing the computational complexity. Quantization is a popular technique [3], [4], [5] for reducing the precision and the size of the neural network, thus reducing the computational complexity and the number of resources required.

In this research, we implement a quantized ResNet-50 [6] on multiple network-connected FPGAs. This project builds on top of existing open-source tools, namely FINN [7], Elastic-DF [8] and the AMD/Xilinx VNx UDP/IP stack [9]. ResNet-50 is one of the examples provided with FINN and is also used by others including the Elastic-DF project to illustrate mapping machine learning to multiple FPGAs. In the implementations with FINN and Elastic-DF, ResNet-50 is quantized to 1 bit weights and 2 bit activations. This results in a design size that is small enough to fit on a single AMD/Xilinx Alveo U280, our target accelerator card, and also has a relatively low top-1 accuracy of 67.97%. In contrast, we implement ResNet-50 with 4 bit weights and 5 bit activations. This results in a larger model that requires three Alveo U280s for implementation and also delivers a higher top-1 accuracy of 73.26% which gets closer to a realistic target for image classification.

Our implementation makes use of the Open Cloud Testbed [10], which houses multiple AMD/Xilinx Alveo U280s that are available for public use. Each Alveo U280 is connected directly to a computer host, as well as to a data switch via two 100 Gbps connections per FPGA. The ResNet-50 model is mapped to three Alveo U280s in the flow. The FPGAs are connected using the network infrastructure of the VNx UDP/IP stack provided by AMD/Xilinx which enables direct communication between the FPGAs via the network switch.

In our implementation, the images to be classified are transferred to the first FPGA from its local host, the processing is done on all three nodes and the result is returned to the host from the first node. This is one of two possible configurations and we refer to this as the loop configuration; the other allows the result to be produced on a different FPGA from the first one; we refer to this as the chain configuration.

Both loop and chain configurations illustrate processing in the network. The amount of data to transmit across the network using these approaches is less than if the input data was transmitted to each FPGA separately for processing. This is a consequence of implementing a dataflow architecture for the accelerator.

We differentiate throughout the paper between DataFlow Architectures (DFA) and Matrix of Processing Elements (MPE) architectures. DFAs are fully customized for the ML model, while MPEs implement a general parallel architecture which can accommodate different models but with limited flexibility in terms of, for example, number of processing elements and interconnection patterns.

In the DFA model, as the data to be transmitted between FPGAs is represented by intermediate values in the flow, the result of the last layer computed on one FPGA serves as input to the next layer in the network residing on the next FPGA. Furthermore, the direct FPGA-to-FPGA communication eliminates any overhead that might be incurred when host-to-host communication is involved as data transfer to/from the host and then between hosts is much slower.

The contributions of this paper are:

- Comparison between quantized ResNet-50 with 1 bit weights and 2 bit activations (ResNet-50_W1A2) and ResNet-50 with 4 bit weights and 5 bit activations (ResNet-50_W4A5) in terms of accuracy as well as the resource utilization for their corresponding custom FPGA-based accelerators.
- Multi-FPGA implementation of quantized ResNet-50 model with 4 bit weights and 5 bit activations (ResNet-50_W4A5). The design is generated using the FINN framework and the model is partitioned using the Elastic-DF partitioner. The final custom accelerator is deployed on three network-connected AMD/Xilinx Alveo U280 data center accelerator cards in the Open Cloud Testbed.
- Comparison between FPGA-based custom designs generated through the FINN framework for ResNet-50_W1A2 and ResNet-50_W4A5 in terms of throughput. ResNet-50_W4A5 achieves a comparable throughput compared to ResNet-50_W1A2 which fits on one single FPGA while the former requires three FPGAs.
- Exploration of a more efficient usage of the available memory on the target platform. By making use of the available Ultra RAM blocks on Alveo U280, the resulting ResNet-50_W4A5 accelerator is able to fit on a single FPGA. The design is compared to the original ResNet-50_W4A5 multi-FPGA accelerator with BRAM storage of weights in terms of resource utilization and throughput.

The rest of this paper is organized as follows: Sec. II goes over the main background specifics of the tools used and related work. Sec. III focuses on the multi-FPGA ResNet-50 accelerator implementation. Sec. IV describes experiments and results, Sec. V presents lessons learned and Sec. VI draws conclusions.

II. Background

A. Open Cloud Testbed (OCT)

OCT [11], [12] is a research platform that offers FPGA-enhanced nodes to users via the CloudLab framework. It provides the ability to perform experiments on emerging cloud services and develop cloud-based applications that can leverage the programmable logic resources offered by FPGAs. CloudLab nodes are bare metal, meaning they are provided without an operating system or any pre-installed software or tools [13]. This provides researchers with a blank slate to configure the system as desired. The flexibility offered by OCT enables us to establish repeatable experimental conditions, as we can install all the required runtime tools, components, and dependencies necessary for executing our machine learning accelerators in hardware.

OCT has AMD/Xilinx Alveo U280 accelerator cards that are directly connected to a 100 GbE network via a 100 GbE data center switch. These network ports are exposed to FPGA users. This enables direct FPGA-to-FPGA communication by eliminating the need for processor involvement, thus significantly reducing the latency associated with data transfer and resulting in faster processing times. Additionally, it allows distributing a complex machine learning accelerator across multiple FPGAs allowing direct communication between them. In addition to the network connectivity, the U280s are also connected to a host processor via PCIe. This connection is used to transfer images and weights from the host to the FPGA and to retrieve inference results from the FPGA back to the host by using input and output DMAs (IDMA and ODMA) implemented on the FPGA.

B. FINN

Deep Neural Network inference on FPGAs can be explored using FINN [7], an open-source end-to-end framework that allows building custom accelerators based on a given network topology. The FINN project focuses on Quantized Neural Networks (QNN) and includes the PyTorch library Brevitas [14] for quantization and quantization-aware training of neural networks. The network is exported from Brevitas in the

Open Neural Network Exchange (ONNX) format, a standard that enables interoperability between machine learning tools.

The end goal of FINN is to generate a streaming dataflow hardware accelerator (DFA) for the input quantized ONNX model; however, the framework is highly modular and consists of multiple steps which produce intermediate results. Hence, the flow can be stopped at different stages if intermediate results are needed for a different flow or further analysis. The Brevitas training and export of the model is followed by network preparation which is in turn followed by the hardware build.

The network preparation stage has multiple purposes and sub-steps. One of them is the streamlining step which is in charge of moving operations around and collapsing them into the corresponding nodes and also absorbing floating point scaling factors into integer thresholds [15]. During the same preparation stage, the layers of the model are converted to High Level Synthesis (HLS) layers based on the HLS code library for FINN. One important concept of FINN is folding, which allows changing the number of Processing Elements (PEs) and their SIMD lanes to adjust the throughput and the footprint size of the hardware accelerator. The hardware build stage takes care of generating the bitfile and all the steps that come before that, including IP generation and floorplanning. This is a short overview of FINN; we refer the reader to [7], [16] for more details.

C. Elastic-DF

Elastic-DF is a tool for automatic partitioning and resource balancing for dataflow DNN inference accelerators [8]. The partitioner is based on an Integer Linear Program (ILP) solver and is integrated into FINN as an analysis pass. Based on the resource estimation for each layer and the input constraints (such as resource limits), Elastic-DF aims to find the optimal solution of layer placement across multiple Super Logic Regions (SLRs) for a multi-die FPGA and across multiple FPGAs. Based on the resulting floorplan, FINN splits the ONNX model into partitions where each partition contains a sequence of layers that must be placed on the same SLR. It is important for each partition to fit on one single SLR since die crossings can lead to large propagation delays and therefore to timing closure issues.

Alonso et al. [8] present, along with the partitioner and resource balancer, VNx which is the IP core used for direct FPGA-to-FPGA communication and is covered in Sec. II-D. They demonstrate their tools by deploying MobileNetV1 with 4 bit weights and 4 bit activations and ResNet-50 with 1 bit weights and 2 bit activations on multiple FPGAs.

D. VNx UDP/IP Stack

The VNx UDP/IP stack [9] consists of hardware modules that perform the necessary network and transport layer functions to help send and receive data packets over a network. These modules are designed using hardware description languages and are synthesized to run on an FPGA device such as the Alveo U280. UDP is an unreliable protocol, meaning that it does not provide any guarantees for the delivery of packets, nor does it check for errors in transmission. However, since the FPGAs used in this work are connected to the same switch, there is little risk of packet loss or high latency because the switch acts as a direct link between the FPGAs without the need for additional routing or network hops. As a result, the use of UDP in this context is appropriate, since the reliability and error checking features of other transport protocols, such as TCP, are not necessary and would only add overhead to the communication process.

Applications running on the FPGA use sockets to send and receive data which enable them to establish connections with other applications on different FPGAs or hosts. The VNx stack implements Address Resolution Protocol (ARP) and UDP tables which can be accessed from a host processor via AXI Lite control interfaces. The UDP table manages the state of UDP sockets for sending and receiving data, while the ARP table maps IP addresses to MAC addresses for FPGAs to communicate with each other on the local network. In the context of this work, the machine learning accelerator is the application that we split across multiple FPGAs. Integrating the VNx stack makes it possible to partition a large machine learning model that would not fit on a single FPGA across multiple FPGAs and enable smooth communication among them.

E. Related Work

Multi-FPGA acceleration of DNNs has been previously explored and there is a wide variety of tools and target applications which have been implemented.

Zhang et al. [17] target ResNet-152 on four Virtex Ultrascale FPGAs; however, they transfer data from host to host as opposed to our work where we use direct FPGA-to-FPGA communication. The hosts are connected through 10 GbE Ethernet switches while the FPGAs are PCIe attached.

Fukushima et al. [18] deploy 8-bit quantized ResNet-50 on four MKUBOS boards which are based on AMD/Xilinx Zynq UltraScale+ devices and obtain a throughput of 75.1 images/s. The 8-bit quantization of weights leads to a top-1 accuracy of 74.2%, which is 1% higher than the accuracy obtained by us with 4-bit weights and 5-bit activations. This highlights the fact that comparable accuracy can be achieved with low bit width quantization and this can be a convenient trade-off that results in lower storage demand and smaller accelerated computations. Their communication uses the AMD/Xilinx Aurora IP and each serial link between devices has a bit rate of 8.5 Gbps. The Aurora communication protocol is also used by [19] to interconnect their ZCU102 FPGAs. Jiang et al. [19] propose a framework for accelerating DNN inference across multi-FPGAs; however, they implement a Matrix of Processing Elements (MPE) type of architecture, while we focus on streaming dataflow.

Tarafdar et al. [20] also use ResNet-50 to showcase their AIgean framework for ML deployment on heterogeneous clusters. They deploy the model with 16-bit weights on 10 and 12 AMD/Xilinx ZU19EG FPGAs targeting high throughput.

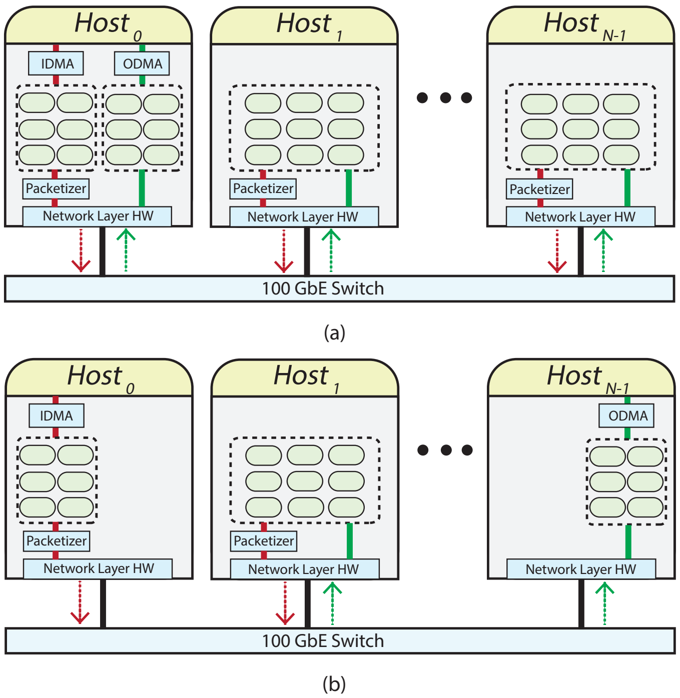

Fig. 1. Multi-FPGA ResNet-50 Accelerator Configurations. (a) Loop configuration: The input images are sent to the first FPGA, the processing is done on all nodes and the result is returned through the first FPGA. (b) Chain configuration: The input images are sent to the first FPGA, the processing is done on all nodes and the result is returned to the host of the last FPGA.

TABLE I
COMPARISON BETWEEN RESNET-50_W1A2 AND RESNET-50_W4A5

Model	Quantization	Dataset	Top-1 Accuracy
ResNet-50	W1A2	ImageNet-1K	67.97%
ResNet-50	W4A5	ImageNet-1K	**73.26%**

They achieve 400 and 660 images/s respectively, by unrolling the multiplications in the convolutional layers at the cost of high resource utilization. In this work, the FPGAs are connected through 100 Gbps Ethernet switches and the UDP communication protocol.

III. MULTI-FPGA RESNET-50 ACCELERATOR

A. Quantization-Aware Training

Using quantization-aware training, we quantize the weights and activations of ResNet-50 [6]. In contrast to post-training quantization, quantization-aware training better preserves the accuracy by simulating the quantization loss and computing the scale factors during the fine-tuning stage of the training process. In post-training quantization, these scale factors are computed after training, which generally leads to a higher drop in accuracy. We chose the bit-width in such a way that the resulting model would have a higher accuracy than ResNet-50_W1A2 and that it would also allow us to demonstrate the multi-FPGA partitioning of an accelerator design that cannot fit on a single FPGA.

We use Brevitas [14], a quantization-aware training library, to train ResNet-50 on the ImageNet-1K [21] dataset, and we reduce the bit-width of weights to 4 bits and the bit-width of activations to 5 bits. We keep the first and the last layer weights at 8 bits to preserve accuracy. We initialize the weights of our quantized model from a pre-trained ResNet-50 model. We use Stochastic Gradient Descent (SGD) with a batch size of 64 for each of the four training GPUs. The learning rate starts from 0.001 and is divided by 10 for every 30 epochs; the model is trained for 90 epochs. We also use a weight decay of 0.00005 and a momentum of 0.875. After quantization-aware training, we export the model weights to integer representation with scale factors in ONNX format.

B. Hardware Implementation

The next implementation step consists of processing the ResNet-50_W4A5 ONNX model with the FINN framework. We go through the same network preparation steps and we keep the same folding configuration as the one used for ResNet-50_W1A2 in [22] because we want to fairly compare the throughput between the two implementations.

We analyze the FPGA resource estimation for ResNet-50_W4A5, and deduce that, with the same folding configuration as ResNet-50_W1A2, it is not possible to fit its corresponding hardware accelerator on a single FPGA. Hence, we use the Elastic-DF analysis pass integrated with FINN to find the optimal placement of layers on the available SLRs. This optimal placement across multiple multi-die FPGAs and across their SLRs later instructs FINN how to split the ONNX model into partitions. A computing kernel is generated for each partition (a contiguous sequence of layers which reside on the same SLR) and the kernels are connected through AXI Stream interfaces. Alonso et al. [8] refer to this type of design as an explicit dataflow design, whereas an embedded dataflow has a single kernel that contains the whole design. We split the final set of kernels across three FPGAs based on their resource utilization making sure that the total BRAM utilization on each device is under 80% as Xilinx advises to exclude the possibility of routing congestion. The network layer adds logic overhead; the resource utilization breakdown can be seen in [8]. We use the VNx UDP stack which is much more lightweight than the TCP stack.

Figure 1 shows the setup of the network-connected FPGAs and how the ResNet-50 accelerator kernels fit into the infrastructure. There are two configurations which we investigate: **(a) Loop configuration** and **(b) Chain configuration**. In [8], from a model parallelism point of view of a multi-FPGA DFA implementation, the chain configuration is referred to as Hardware Model-Parallel (HWMP) and the loop configuration is referred to as Transparent Model-Parallel (TMP).

In the **loop configuration**, Figure 1 (a), the batch of input images to be classified are transferred from the host on the first Alveo U280 through the IDMA and the result is returned by the same node to the host through the ODMA. The difference in the **chain configuration**, Figure 1 (b), is that the result is returned to the host of the last FPGA. Note that, as shown in

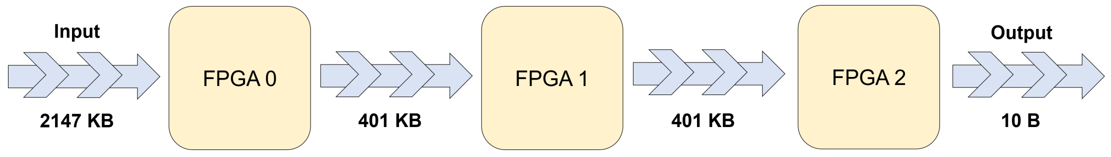

Fig. 2. Sizes of the data transfers for one inference of the multi-FPGA ResNet-50 accelerator

TABLE II
COMPARISON BETWEEN THE RESOURCE UTILIZATION OF RESNET-50_W1A2 AND RESNET-50_W4A5 FPGA-BASED ACCELERATORS

Design	LUT	BRAM tile (36 Kb)	URAM (288 Kb)
Alveo U280 (available resources)	1903200	2016	960
ResNet-50_W1A2	285520	**1392.5**	20
Utilization (%)	15%	**69.07%**	2.08%
ResNet-50_W4A5	830906	**4320**	40
Utilization (%)	43%	**214%**	4.16%

TABLE III
COMPARISON BETWEEN THE THROUGHPUT OF RESNET-50_W1A2 AND RESNET-50_W4A5 FPGA-BASED ACCELERATORS

Model	Quantization	No. of FPGAs	Frequency [MHz]	Throughput [images/s]
ResNet-50	W1A2	1	160	176.10
ResNet-50	W4A5	3	200	162.32

Figure 1 (a), only the first host on the first node is involved, whereas in Figure 1 (b), there is communication with the hosts on both the first and the last accelerator card. The other hosts of the intermediate nodes in each topology are not involved, and there is no data transfer needed between the hosts and the reconfigurable fabric on these nodes. All intermediate results are transferred between nodes in the FPGA cluster through direct FPGA-to-FPGA communication. An advantage of the loop configuration is that from the software perspective, it looks like a single FPGA accelerator.

A user may find it useful for the output to be returned to the same host and the processing to be done on the FPGA cluster; however, it may also be useful to feed the images to one node and to return the classification label(s) from a different node. The configuration choice depends on the application and on the node where the user would like to receive the output. If there were additional hardware connected to, for example, the last FPGA in the pipeline or if its host needs the classification results for further processing, the chain configuration would be preferred.

IV. EXPERIMENTS AND RESULTS

The experiments have been done using Vitis 2020.1 and FINN v0.7, and the tests have been run on the Alveo U280 data center accelerator cards hosted in Open Cloud Testbed. The hardware accelerator for ResNet-50_W1A2 has been generated from the pre-trained model provided by FINN examples [22]. ResNet-50_W4A5 has been trained with Brevitas from scratch on the ImageNet-1K dataset. Both training and inference images are RGB images resized to 224x224x3 pixels.

In the ResNet-50 architecture, the last fully-connected layer is the layer with the highest storage demand. Therefore, it is more convenient to store the weights externally and then feed them to the accelerator than to store them on chip. The external weight file is fed from the host to the first FPGA, as well as the input batch of images to be classified, but through separate IDMAs.

Table I shows the accuracy comparison between ResNet-50_W1A2 and ResNet-50_W4A5, and Table II presents the FPGA resource demand of the accelerator designs. It can be observed that for an around 5% increase in accuracy, the latter model needs triple the number of LUTs and BRAMs. However, the most critical resource is the BRAM tile. The number of used LUTs is notably below the number of available LUTs, while the required number of BRAM tiles significantly exceeds the available resources on one Alveo U280. It is also recommended by Xilinx to keep the total BRAM utilization under 80% because exceeding this limit might lead to routing congestion and inability to meet timing. This makes it impossible to fit the accelerator for ResNet-50_W4A5 on one single FPGA and the only solution in this case is to partition the dataflow accelerator and deploy it on three FPGAs.

Table III compares the two ResNet-50 designs with different precisions in terms of throughput. ResNet-50_W4A5 achieves a comparable throughput compared to ResNet-50_W1A2 even though the former is split across three different FPGAs. The single FPGA implementation runs at a frequency of 160 MHz, while the multi-FPGA accelerator reaches 200 MHz. The slower clock on the single FPGA is likely due to routing complexity. There is more flexibility for routing on the three FPGA design due to a lower usage of resources.

In Figure 2 we show the sizes of the data transfers between the nodes for one inference. The input consists of one input image and the external weights. The output consists of the top 5 labels. It can be observed that the intermediate results have a much lower size than the input. If copies of the input frame or external weights would need to be sent to each node for parallel processing, as in the case of a split MPE type of accelerator, that would increase the latency. In this case, for a DFA, only the intermediate results are transferred between the FPGAs. Furthermore, the direct FPGA-to-FPGA communication and the streaming dataflow architecture of the accelerator enable fast data transfers between nodes without the involvement of the host. Communication through the host would ultimately affect the throughput since it can be significantly slower. The main target and advantage of DFAs is high

TABLE IV
COMPARISON BETWEEN DIFFERENT FPGA-BASED IMPLEMENTATIONS OF RESNET-50_W4A5

Implementation	LUT	BRAM tile (36 Kb)	URAM (288 Kb)	Number of FPGAs	Frequency [MHz]	Throughput [images/s]
Alveo U280 (available resources)	1903200	2016	960	1	-	-
Predominant BRAM storage of weights	830906	4320	40	3	200	162.32
Utilization % (with respect to one U280)	43%	214%	4.16%	-	-	-
Utilization % (with respect to three U280)	14.55%	71.42%	1.3%	-	-	-
Predominant URAM storage of weights	561797	981.5	587	1	150	165.09
Utilization % (with respect to one U280)	29.51%	48.68%	61.14%	-	-	

throughput compared to MPE architectures. A multi-FPGA implementation of the architecture with directly connected FPGAs is more efficient from a throughput point of view since it significantly reduces the latency for communication between nodes.

The baseline FINN configuration for ResNet-50_W1A2 in [22] includes, besides the folding factors, the type of RAM the weights of the model should be stored in. Because we work with Alveo U280 which has an Ultrascale architecture, we also explore a more memory-efficient implementation. The initial requirement is for the weights of ResNet-50_W4A5 to be stored in Block RAM, however, as shown in Table II, the BRAM tiles represent the most critical FPGA resource. The other resources are underutilized; thus we chose to take advantage of the available Ultra RAM (URAM) and to try to rescale the size of the whole accelerator to fit on a single FPGA. This means that part of the storage that was previously implemented through BRAM is now implemented with URAM; we do not change the folding configuration or any other parameters. Table IV shows that by storing the weights of the fully-connected layers in URAM, the number of BRAM tiles required decreases by a factor of four. The URAM utilization increases by 14x, but is still below the number of available tiles on the accelerator card. Ultimately, the entire ResNet-50_W4A5 design fits on one FPGA and the throughput is maintained even though the maximum frequency achieved is lower.

V. DISCUSSION

The main purpose of FPGAs in the cloud is to accelerate different time-consuming tasks as they can achieve better performance for lower power consumption compared to CPUs and GPUs [23]. In general, only critical parts of applications are offloaded onto the FPGAs since they have limited resources which are often not enough to host an entire cloud application. Network-connected FPGA clusters support deployment of larger designs. The direct FPGA-to-FPGA communication through 100 Gb Ethernet networking and the UDP/IP protocol also provide a faster solution in comparison to host-to-host communication. In the machine learning context, accurate models demand a significant amount of storage and they also require a large amount of computation. Even when using aggressive quantization and quantization-aware training to reduce the size of the accelerator, the final hardware design might still be too large to fit on a single FPGA; an example of this being ResNet-50_W4A5 implemented in this work.

This work focuses on dataflow inference accelerators that target high throughput. DFAs are architectures customized for a specific model where each layer has resources allocated as needed, compared to MPEs which are more generic having a fixed set of processing elements. The generality of MPEs comes with several drawbacks such as limited flexibility and intensive communication between on-chip and off-chip memory as they need to fetch weights and activations from the external memory when layers are scheduled to be executed on the available processing units. DFAs minimize the amount of data transfers between on-chip and off-chip memories by storing the parameters of most layers on-chip, hence reducing latency and power consumption. While MPEs can pipeline computations on a fixed set of processing elements, DFAs need to allocate resources for each layer, hence their challenge is frequently that resources are limited. In this situation, being able to partition a design onto multiple network-connected FPGAs allows for the accommodation of larger ML dataflow accelerators with higher accuracy and higher throughput.

ResNet-50 represents a basic architecture that has enabled accurate image classification. Recently, other ML models have taken the lead. After all, the accuracy of ResNet50_W4A5 implemented in this work (73.26%) is much lower than the state-of-the-art (around 90%). Currently, on the ImageNet-1K dataset, the highest accuracy is achieved by vision transformers [24]; [25] provides an example of a transformer acceleration framework. However, transformers demand substantially more storage. The vision transformer used in [24] has 1.88 billion parameters, while ResNet-50 has 23 million parameters. Even if we stick with a CNN architecture, the most accurate model has almost 100x more parameters (2158 million parameters). In the end, a designer needs to determine the best trade-off between accuracy and the feasibility of deploying the corresponding accelerator on FPGAs. Network-connected FPGAs give a designer more options.

VI. CONCLUSION

We showcase the implementation of a multi-FPGA dataflow accelerator for a higher precision version of ResNet-50 which cannot be deployed on a single FPGA. Network-connected FPGAs enlarge the design space and options for implementation. In this research, ResNet-50 is trained and quantized with 4 bit weights and 5 bit activations. A custom accelerator is generated using the FINN framework and is partitioned using the Elastic-DF partitioner and resource balancer. The design is deployed in the Open Cloud Testbed on three network-connected Alveo U280 data center accelerator cards which communicate through the VNx UDP/IP stack and 100 Gbps Ethernet. We compare this work in terms of accuracy, resource utilization and throughput with the accelerator of ResNet-50_W1A2 which is generated and deployed using the same tools. An alternative implementation is explored where we make use of the available Ultra RAM and we reduce the size of the accelerator, the final design being able to fit on one FPGA.

In the future, we plan to explore different ML architectures that achieve or are close to state-of-the-art accuracy on image classification tasks, such as vision transformers. We aim to implement and optimize accelerators for such models which can successfully be implemented on FPGA clusters. Moreover, we intend to extend FINN to support 3D CNNs for video classification and investigate how the performance of DFA architectures compares to MPE accelerators for such applications.

Acknowledgment

This work is funded in part by the National Science Foundation (NSF) under Grant CCF-1901378. The Open Cloud Testbed is funded by NSF grants CNS-1925464, CNS-1925504, and CNS-1925658. The authors would also like to thank AMD/Xilinx for their donations.

References

[1] R. Wu, X. Guo, J. Du, and J. Li, "Accelerating neural network inference on fpga-based platforms—a survey," *Electronics*, vol. 10, no. 9, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/9/1025

[2] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, "A survey of FPGA based neural network accelerator," *CoRR*, vol. abs/1712.08934, 2017. [Online]. Available: http://arxiv.org/abs/1712.08934

[3] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," 2021.

[4] M. Sun, Z. Li, A. Lu, Y. Li, S.-E. Chang, X. Ma, X. Lin, and Z. Fang, "Film-qnn: Efficient fpga acceleration of deep neural networks with intra-layer, mixed-precision quantization," in *Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 134–145. [Online]. Available: https://doi.org/10.1145/3490422.3502364

[5] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," 2021.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[7] M. Blott, T. B. Preußer, N. J. Fraser, G. Gambardella, K. O'brien, Y. Umuroglu, M. Leeser, and K. Vissers, "FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, dec 2018. [Online]. Available: https://doi.org/10.1145/3242897

[8] T. Alonso, L. Petrica, M. Ruiz, J. Petri-Koenig, Y. Umuroglu, I. Stamelos, E. Koromilas, M. Blott, and K. Vissers, "Elastic-DF: Scaling Performance of DNN Inference in FPGA Clouds through Automatic Partitioning," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 15, no. 2, dec 2021. [Online]. Available: https://doi.org/10.1145/3470567

[9] M. Ruiz. (2022) XUP Vitis Network Example (VNx). AMD. [Online]. Available: https://github.com/Xilinx/xup_vitis_network_example

[10] M. Zink, D. Irwin, E. Cecchet, H. Saplakoglu, O. Krieger, M. Herbordt, M. Daitzman, P. Desnoyers, M. Leeser, and S. Handagala, "The Open Cloud Testbed (OCT): A Platform for Research into new Cloud Technologies," in *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, 2021, pp. 140–147.

[11] M. Leeser, S. Handagala, and M. Zink, "FPGAs in the Cloud," *Computing in Science & Engineering*, vol. 23, no. 6, pp. 72–76, 2021.

[12] S. Handagala, M. Leeser, K. Patle, and M. Zink, "Network Attached FPGAs in the Open Cloud Testbed (OCT)," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–6.

[13] D. Duplyakin, R. Ricci, A. Maricq, G. Wong, J. Duerig, E. Eide, L. Stoller, M. Hibler, D. Johnson, K. Webb, A. Akella, K. Wang, G. Ricart, L. Landweber, C. Elliott, M. Zink, E. Cecchet, S. Kar, and P. Mishra, "The design and operation of CloudLab," in *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. Renton, WA: USENIX Association, Jul. 2019, pp. 1–14. [Online]. Available: https://www.usenix.org/conference/atc19/presentation/duplyakin

[14] A. Pappalardo. (2022) Xilinx/brevitas. Xilinx. [Online]. Available: https://doi.org/10.5281/zenodo.3333552

[15] Y. Umuroglu and M. Jahre, "Streamlined Deployment for Quantized Neural Networks," *ArXiv*, vol. abs/1709.04060, 2017.

[16] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 65–74. [Online]. Available: https://doi.org/10.1145/3020078.3021744

[17] W. Zhang, J. Zhang, M. Shen, G. Luo, and N. Xiao, "An Efficient Mapping Approach to Large-Scale DNNs on Multi-FPGA Architectures," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 1241–1244.

[18] Y. Fukushima, K. Iizuka, and H. Amano, "Parallel Implementation of CNN on Multi-FPGA Cluster," in *2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, 2021, pp. 77–83.

[19] W. Jiang, E. H.-M. Sha, X. Zhang, L. Yang, Q. Zhuge, Y. Shi, and J. Hu, "Achieving Super-Linear Speedup across Multi-FPGA for Real-Time DNN Inference," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, oct 2019. [Online]. Available: https://doi.org/10.1145/3358192

[20] N. Tarafdar, G. Di Guglielmo, P. C. Harris, J. D. Krupa, V. Loncar, D. S. Rankin, N. Tran, Z. Wu, Q. Shen, and P. Chow, "AIgean: An Open Framework for Deploying Machine Learning on Heterogeneous Clusters," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 15, no. 3, dec 2022. [Online]. Available: https://doi.org/10.1145/3482854

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[22] Xilinx Research Labs. (2020) FINN Examples. AMD/Xilinx. [Online]. Available: https://github.com/Xilinx/finn-examples

[23] C. Bobda, J. M. Mbongue, P. Chow, M. Ewais, N. Tarafdar, J. C. Vega, K. Eguro, D. Koch, S. Handagala, M. Leeser, M. Herbordt, H. Shahzad, P. Hofste, B. Ringlein, J. Szefer, A. Sanaullah, and R. Tessier, "The future of fpga acceleration in datacenters and the cloud," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 15, no. 3, feb 2022. [Online]. Available: https://doi.org/10.1145/3506713

[24] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le, "Symbolic Discovery of Optimization Algorithms," 2023.

[25] Z. Li, M. Sun, A. Lu, H. Ma, G. Yuan, Y. Xie, H. Tang, Y. Li, M. Leeser, Z. Wang, X. Lin, and Z. Fang, "Auto-vit-acc: An fpga-aware automatic acceleration framework for vision transformer with mixed-scheme quantization," in *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)*, 2022, pp. 109–116.