An FPGA platform for Reconfigurable Heterogeneous HPC and Cloud Computing

Francois Abel, Burkhard Ringlein, Beat Weiss, Christoph Hagleitner and Bernard Metzler

IBM Research - Zurich
The advent of accelerators

The cloudFPGA platform from 10’000 feet

Architecture and design choices
- Hardware: Boards, SLEDs, chassis
- Software: Shell, Role, Management Core
- Data Center: Resource Manager

Deployment @ ZYC2

Network Stack
- Data path
- RDMA/Fabric choices
- NVM integration

Summary & Outlook & Call for contributions
Memory capacities are scaling directly with Moore’s law.

So did the clock speeds until the very early 2000s.

Then physical effects limited the clock speeds to ~ 4Ghz.

To take profit from a still increasing number of transistors, specialization seems to be a promising path.

System specialization using accelerators: Architectures designed with a specific class of computations in mind.
Silicon Alternatives for rapid enterprise-ready Specialization

- A GPU is effective at processing the same set of operations in parallel – single instruction, multiple data (SIMD).
- A GPU has a well-defined instruction-set, and fixed word sizes – for example single, double, or half-precision integer and floating-point values.

- An FPGA is effective at processing the same or different operations in parallel – multiple instructions, multiple data (MIMD).
- An FPGA does not have a predefined instruction-set, or a fixed data width.
cloudFPGA Goals

Goal → Deploy FPGAs at **large scale** in hyperscale Data Centers

1-10s of thousands per DC

- Cloud driven requirements
  - ✓ Server commodity & homogeneity
  - ✓ Decrease in cost and power
  - ✓ Easy to manage and to deploy
  - ✓ On-demand acceleration
  - ✓ High utilization + workload migration
  - ✓ Security, virtualization, orchestration
  - ✓ Hybrid → public & private
  - ✓ Flexible → IaaS, PaaS, FaaS
  - ✓ Clusters → #accelerators per server
  - ✓ Community → # of APPs, # of developers
cloudFPGA in a few Words

- **End of CPU slavery**
  - FPGA becomes the compute node

- **Standalone Operation**
  - Disaggregate from CPU servers
  - Independent scaling of compute
  - Fast, independent operation (power on/off)

- **Network attached**
  - TCP/UDP/IP/Ethernet (today 10 .. 40GbE)
  - Leaf-spine topology

- **Hyperscale infrastructure**
  - Focus on cost, energy, density, scalability
  - Promotes usage of mid-range FPGAs
FPGAs to become 1st class citizens in DC Cloud

CPU-Centric Deployment

FPGA as a Co-Processor

FPGA as a Peer-Processor

FPGA-Centric Deployment

This work (cloudFPGA)
DC Vision = Hyperscale Infrastructure

The FPGA platform

64/chassis

1024/rack

Standalone
Network-attached
FPGAs over
TCP/IP/Ethernet

10 Tb/s full-duplex

Plentiful/DC

© OpenFabrics Alliance
Cloud Vision = IaaS, PaaS, FaaS

IaaS
- Source Code
- VM
- 1 FPGA

PaaS
- GENOME.COM
- IBM Container
- 100+ FPGAs

FaaS
- IBM Cloud Services (e.g. Watson-VR, -NLU, -D)
- IBM Cloud Functions
- Blockchain
- FPGA-based Serverless Functions and Microservices
- 1000+ FPGAs

© OpenFabrics Alliance
Architecture & Design choices
HW: Boards, SLEDs, chassis
Standalone → The FPGA becomes the Node

Disaggregate FPGA from the server

Bare Metal VM Container

IP Address: 10.10.1.50
DRAM: 32GB, Cores: 4

FPGA

IP Address: 10.10.1.9
DRAM: 8GB, BRAM: 38MB
CLBs: 660,000. DSPs: 2760
1. Replace PCIe I/F with integrated NIC (iNIC)

2. Turn FPGA card into a standalone resource

3. Replace transceivers with backplane connectivity
One carrier SLED (a.k.a PoD) = 32 FPGA modules
The cloudFPGA Platform (19”x2U w/64 FPGAs)
Architecture & Design choices
SW: Shell, Role, Management core
Hardware Abstraction → Shell Role Architecture (SRA)

ROLE (non-privileged)
Embeds user’s application logic. Partially reconfigured over the network. (typically HLS)

SHELL (privileged logic)
Abstracts hardware components of FPGA and exposes standard AXI(S) interface to user
cloudFPGA Development Kit (cFDK)

Typical HLS flow

- **C/C++**
  - HLS Flow
  - User Design
  - DDR4 I/F
  - Memory I/F
  - Network I/F
  - Integrated NIC (iNIC)

- **Verilog/VHDL**
  - HDL Flow
  - TOP (v, vhdl)
  - SHELL (cpp)

- **ROLE (v, vhdl)**
  - Synthesis, Place and Route
  - Partial Bit Stream
  - Static Bit Stream

- **SHELL (dcp, xci)**
  - Vendor Provided
FPGA Management Core

There is one management core per FPGA (FMC):

- The FMC contains a simplified HTTP server which provides support for the REST API calls issued by the Data Center Resource Manager (DCRM).

The FMC understands REST API calls:

- **POST /configure**: Submits a partial bitfile and triggers the PR of the Role region.
- **GET /status**: Returns some application-specific status information.
- **PUT /node_id**: Sets the node-id register of the Role.
- **POST /routing**: Sends the routing information of a cluster to the FPGA.
Architecture & Design choices
DC: Resource manager
Cloud Service Architecture for FPGAs (1/2)

- **Instance** = CPU + Image
- **Cluster** = $N \times Instance$

A typical cloud service hosting VMs has three components:
- A pool of compute resources
- A database of VM images
- A management service
- Instance = FPGA + Bitstream
- Cluster = $N \times \text{Instance}$
RESTful Web API Based
cloudFPGA Deployment @ ZYC2
IaaS - “Hello, World!” with a single FPGA

- Download the cFDK to work remotely on your desktop or use a VM @ ZYC2
- Setup a VPN client, create an OpenStack project and a private network for it
- Develop and simulate
- Place and route
- Upload your bitstream
- You’ll receive an *image-id*
- Request an instance to be launched with your *image-id*
- You’ll get back an *image-IP* and an *instance-id*
- Ping the *image-IP*
- You are ready to communicate with your FPGA via network sockets with TCP or UDP protocol!
**MPI is the de-facto standard for HPC**

- ZRLPMI → Bring MPI to Reconfigurable Heterogeneous HPC clusters

- ZRLMPIrun → One-click deployment

```bash
$ ZRLMPIrun new udp 10.0.47.11 0ddb12b2-8459-4843-b339-236b2b92b59f 8 ./stencil_SW 0
```

*host IP*  
*partial bitstream id*  
*# of FPGAs*  
*software binary*  
*software rank*
cloudFPGA Networking
Network topology per chassis = 64 FPGAs + 2 Switches

Legend (per slice):
- [==] x8 40GbE up links
- [—] x32 10GbE FPGA-to-Switch links
- [——] x32 10GbE redundant links
- [——] x32 10GbE FPGA-to-FPGA links
- [██] x16 PCIe x8 Gen3
- SP x1 Service Processor

Balanced (i.e. no over-subscription) between north and south links of Ethernet switch
- Ethernet 10 Gb/s
- TCP/IP and UDP/IP stack (+ ICMP, ARP…)
- 10k simultaneous connections
- Active and passive connection establishment
- Network stack: 15% of FPGA logic
cloudFPGA Networking: RX/TX path

- **Application interface**
  - Socket API
  - Asynchronous RX:
    - TOE receives
    - TOE signals app reception
    - App reads/copies data
  - Asynchronous TX:
    - App signals buffer
    - TOE copies data

- **Data path (example RX)**
  - IP receive, TOE places into memory
  - TOE signals data reception and buffer location
  - Socket receive copies data
  - Path-through optimization for small # connections and immediate consume by application

- **Architecture ready for RDMA operations**
  - RoCEv2 or iWarp implementation needed
  - libfabrics or libibverbs application library needed
  - Feel free to contribute! 😊
cloudFPGA Networking: Performance

Comparison with bare-metal servers, VMs and Linux containers @ 10 Gb/s Ethernet

Latency (RTT)
- FPGA/FPGA
  - UDP: 2 μs
  - TCP: 7 μs
- FPGA/Host
  - UDP: 20 μs

Throughput
- FPGA/FPGA
  - UDP: max
  - TCP: 80%
Non-volatile Memory Integration

- **2 options for NVM integration:**
  - Replacing FPGA with NVMeF target possible
  - Adding NVMe resource to FPGA preferred
- **NVMe-oF target (TCP based)**
- Remote (peer FPGA or CPU) + local access
- Very dense NVM integration
- Flexible ‘near storage compute’
FPGAs are eligible to become 1\textsuperscript{st} class citizens
   • Standalone approach sets the FPGA free from the CPU
     • Large scale deployment of FPGAs independent of #servers
     • Significantly lowers the entry barrier
     • Promotes the use of medium and low-cost FPGAs

The network-attachment model
   • Makes FPGAs IP-addressable and scalable in DCs
     • Users can rent and link them in any type of topology
     • Opens the path to use FPGAs in large scale applications
     • Serverless computing, HPC, DNN inference, Signal Processing, ...

The hyperscale infrastructure
   • Integrates FPGAs at the chassis (aka drawer) level
   • Combines passive and active water cooling
   • Key enabler for FPGAs to become plentiful in DCs
Future Work

- **Open-source the cloudFPGA Development Kit (cFDK)**
  - Give the research community access to cloudFPGA platform

- **Walking up the application stack**
  - Lower-precision inference and autoML
  - Support for Vitis accelerated libraries
  - Large-scale distributed applications
  - Support popular programming languages and frameworks

- **Walking up the systems stack**
  - Integration with Function-as-a-Service (aka Serverless computing)
  - Add composable and disaggregated storage (NVMe-oF)
  - Lighter and faster data center network protocols
  - Adding RDMA protocols and API’s

- **Expand the numbers of Xilinx-based modules & support other FPGA vendors**

- **Share the cloudFPGA platform design (e.g. à la OCP)**
THANK YOU
Bernard Metzler
IBM Research - Zurich
BACKUP
From top-of-rack down to SLED/PoD switch

From 7938 cm³ ... (41x44x4.4cm)

32 x 10GbE + 8 x 40GbE

...to 378 cm³ (14x6x4.5cm)

Intel SeaCliffTrail – ToR reference system

Switch Module SM6000

48 x 10GbE + 4 x 40GbE

© OpenFabrics Alliance
How does it compare w/ PCIe cards?

- For comparison: ALPHA DATA ADM-PCIE-9H3, 1/2 Length, low profile, x16 PCIe form Factor
How to disaggregate 4PB per rack with NVMe-over-TCP