Enabling FPGAs for Heterogeneous Cloud Computing

Paul Chow

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

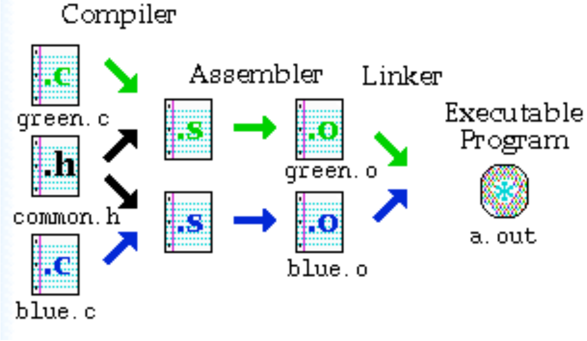
April 16, 2018

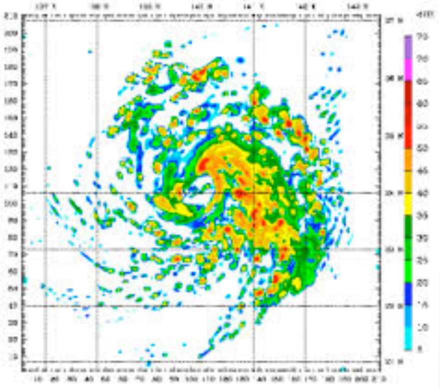
How do we put FPGAs in the Cloud?

- What makes a computer?
- Current state of FPGA computing
- What's needed to compute with FPGAs
- What's needed to compute with FPGAs in a cloud
- FPGA computing at UofT

What’s a Computer?

- Provide a baseline for what we are talking about





FPGA Computer Programming

+

PCIe
MIG
DMA
Ethernet
Drivers

+

```
always @(posedge clock)
{
if(!reset_b)
        q <= D;
}
```

But, HLS fixes this!

Not really…

Consider

+

```
08048918     pushl  %ebp
08048919     movl   %esp,%ebp
0804891b     subl   $0x4,%esp
0804891e     movl   $0x0,0xfffffffc(%ebp)
08048925     cmpl   $0x63,0xfffffffc(%ebp)
08048929     jle    08048930
0804892b     jmp    08048948
0804892d     nop
0804892e     nop
0804892f     nop
08048930     movl   0xfffffffc(%ebp),%eax
08048933     pushl  %eax
08048934     pushl  $0x8049418
08048939     call   080487c0 <printf>
0804893e     addl   $0x8,%esp
08048941     incl   0xfffffffc(%ebp)
08048944     jmp    08048925
08048946     nop
08048947     nop
08048948     xorl   %eax,%eax
0804894a     jmp    0804894c
0804894c     leave
0804894d     ret
```

x86 motherboard + x86 assembler

Is like...

+

```
always @(posedge clock)
{
if(!reset_b)
        q <= D;
}
```

FPGA board + Verilog

Adding HLS is just

+

```
main()
int A[10],sum;
{
sum = 0;
for(i = 0; i <10; i++)
#pragma HLS PIPELINE II=1
        sum =+ A[i];
}
```

Still need to build your own I/O services

Which is like

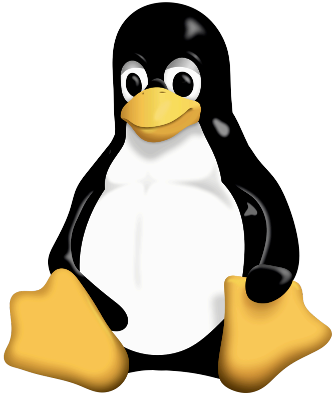
+

```
main()
int A[10],sum;
{
sum = 0;
for(i = 0; i <10; i++)
        sum =+ A[i];
}
```

x86 motherboard + gcc

What makes a computer?

+



+

```
main()
int A[10],sum;
{
sum = 0;
for(i = 0; i <10; i++)
        sum =+ A[i];
}
```

Linux provides services and an abstraction from the physical hardware

Recent FPGA Computing Platforms

SDAccel™ Environment

MAXELER Technologies

MAXIMUM PERFORMANCE COMPUTING

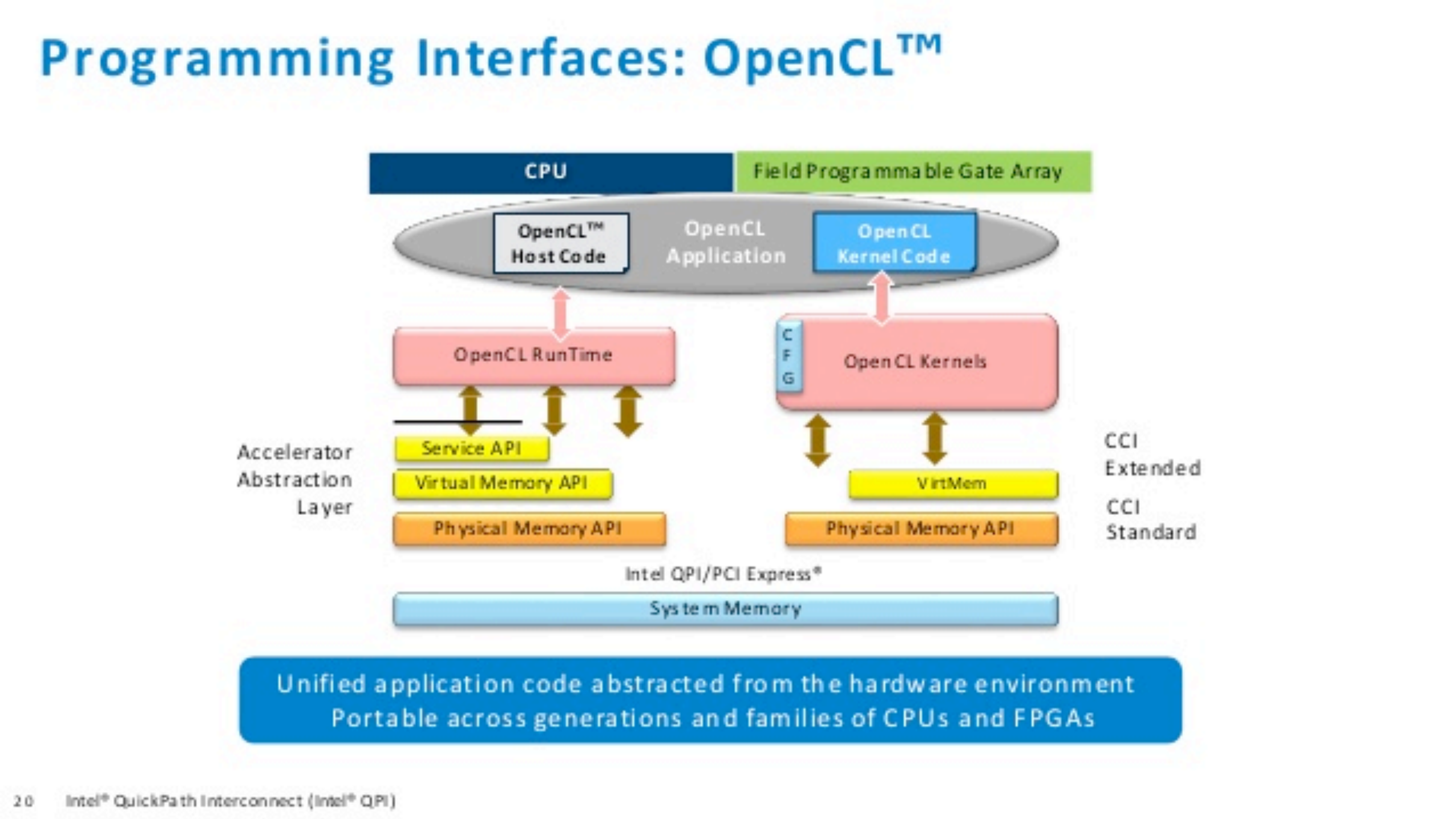
SDSoC™ Environment

INTEL® FPGA SDK FOR OPENCL™

Enabling FPGAs into heterogeneous platforms.

What makes them computing platforms?

- Hardware is abstracted
- Memory is managed
- Runtimes that manage configuration, data transfers, accelerator hardware
- Like what you expect when you run your C programs



Programming Interfaces: OpenCL™

CPU

Field Programmable Gate Array

OpenCL™ Host Code

OpenCL Application

OpenCL Kernel Code

OpenCL RunTime

C F G

OpenCL Kernels

Accelerator Abstraction Layer

Service API

Virtual Memory API

Physical Memory API

VirtMem

Physical Memory API

CCI Extended

CCI Standard

Intel QPI/PCI Express®

System Memory

Unified application code abstracted from the hardware environment
Portable across generations and families of CPUs and FPGAs

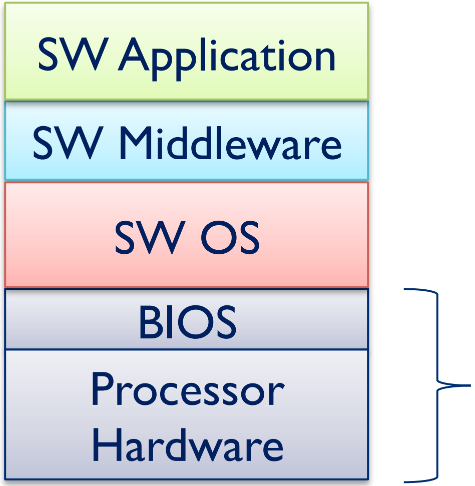
20 Intel® QuickPath Interconnect (Intel® QPI)

Commercial FPGA Clouds

- Microsoft – internal infrastructure
  - – Baidu too?
- Public – Amazon F1, Huawei, Tencent, Alibaba
  - – tools + some cores
  - – Maybe OpenCL
  - – Commercial services using FPGAs

WHAT DO WE NEED?

The Parts (Before the Cloud)



SW Application

SW Middleware

SW OS

BIOS

Processor Hardware

MPI Rank 0 and Software Ranks

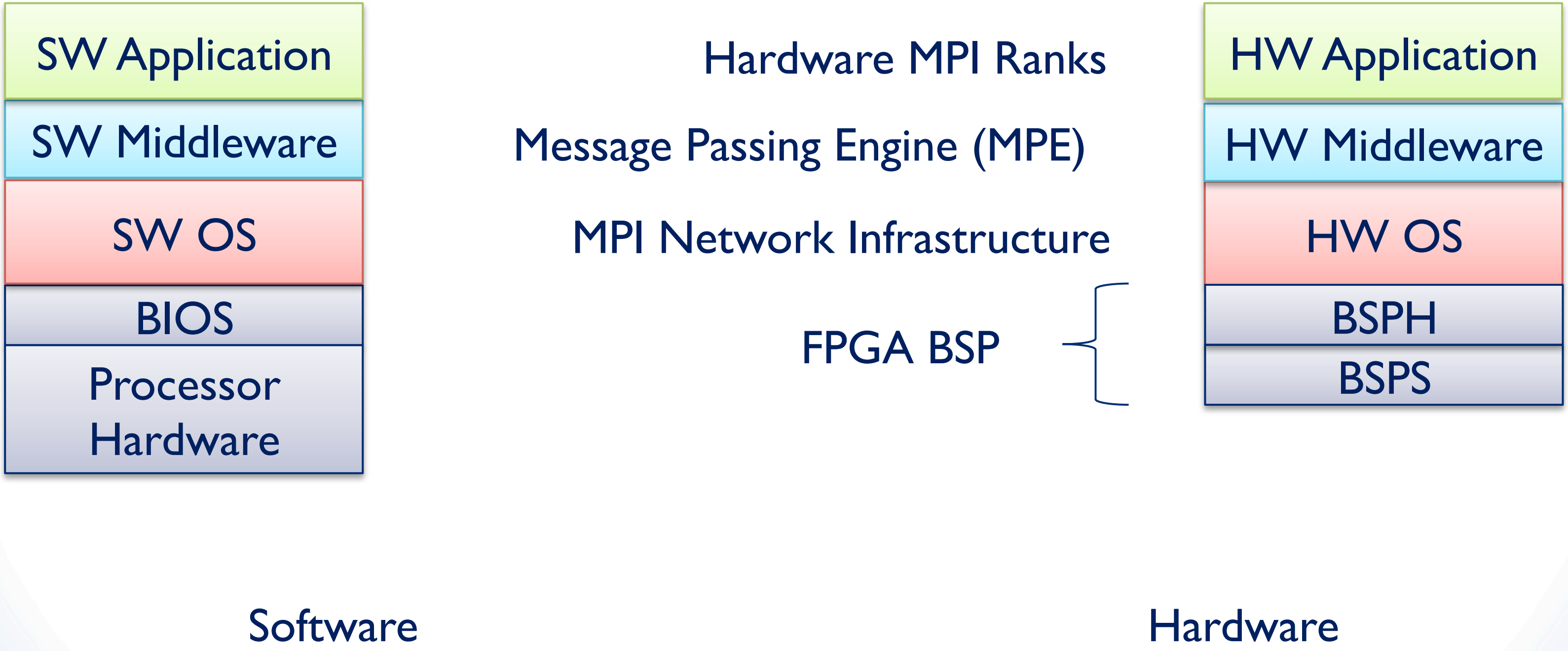
MPI Library

Linux

Xeon Processor Motherboard

Software

The Parts (Before the Cloud)



SW Application

SW Middleware

SW OS

BIOS

Processor Hardware

Hardware MPI Ranks

Message Passing Engine (MPE)

MPI Network Infrastructure

FPGA BSP

HW Application

HW Middleware

HW OS

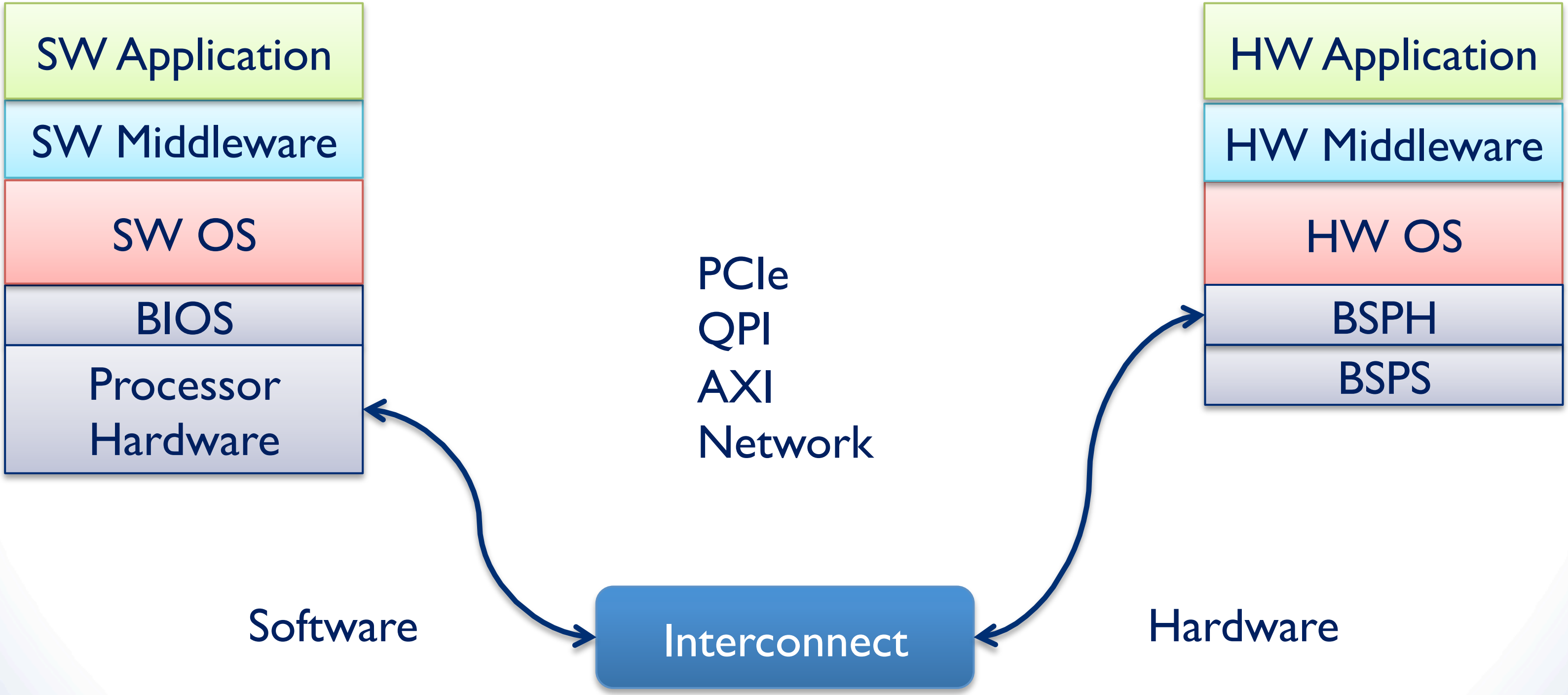
BSPH

BSPS

Software

Hardware

The Parts (Before the Cloud)


SW Application

SW Middleware

SW OS

BIOS

Processor Hardware

HW Application

HW Middleware

HW OS

BSPH

BSPS

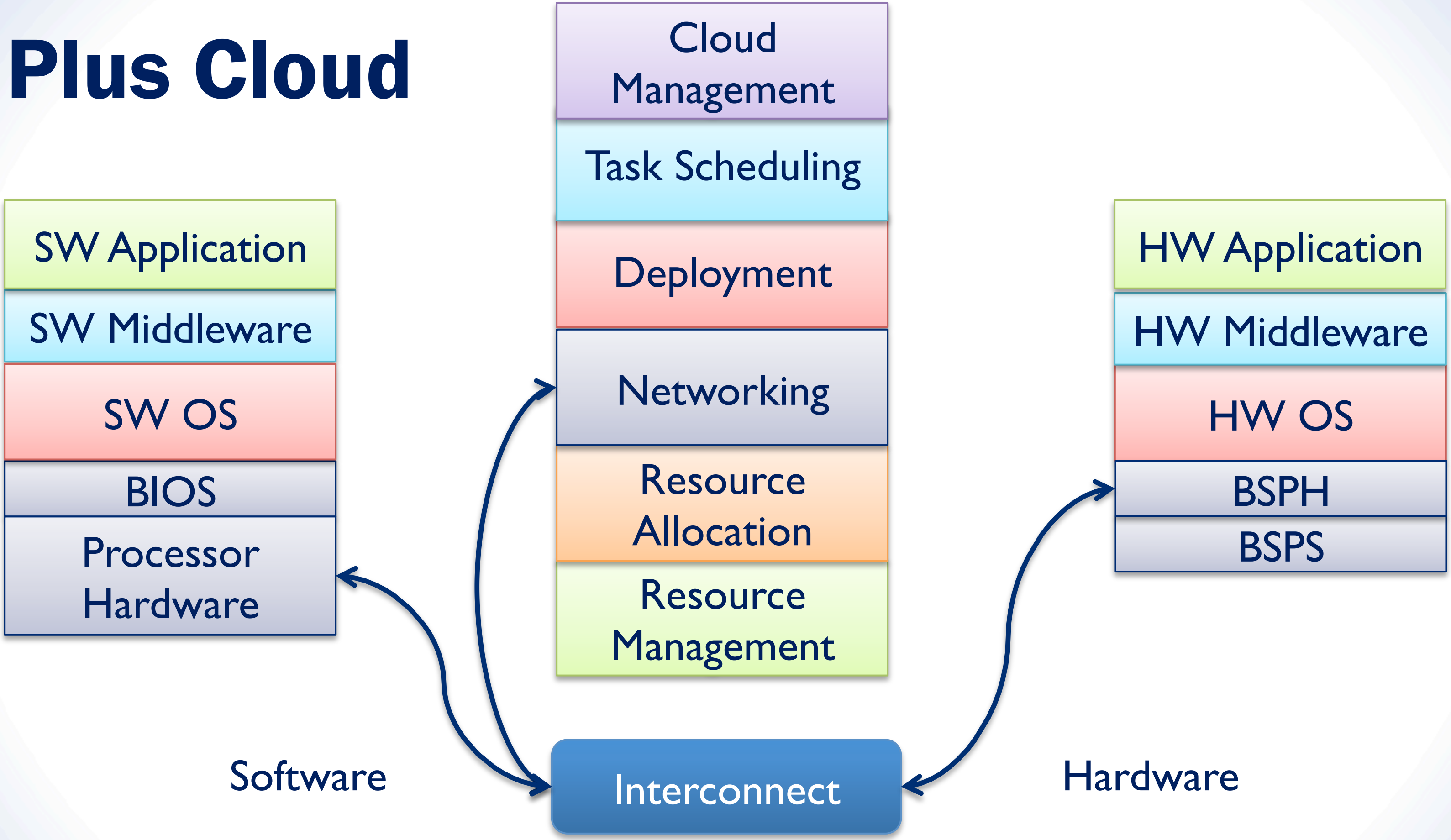
PCIe
QPI
AXI
Network

Software

Interconnect

Hardware

Plus Cloud


Cloud Management
Task Scheduling
Deployment
Networking
Resource Allocation
Resource Management
SW Application
SW Middleware
SW OS
BIOS
Processor Hardware
HW Application
HW Middleware
HW OS
BSPH
BSPS
Interconnect
Software
Hardware

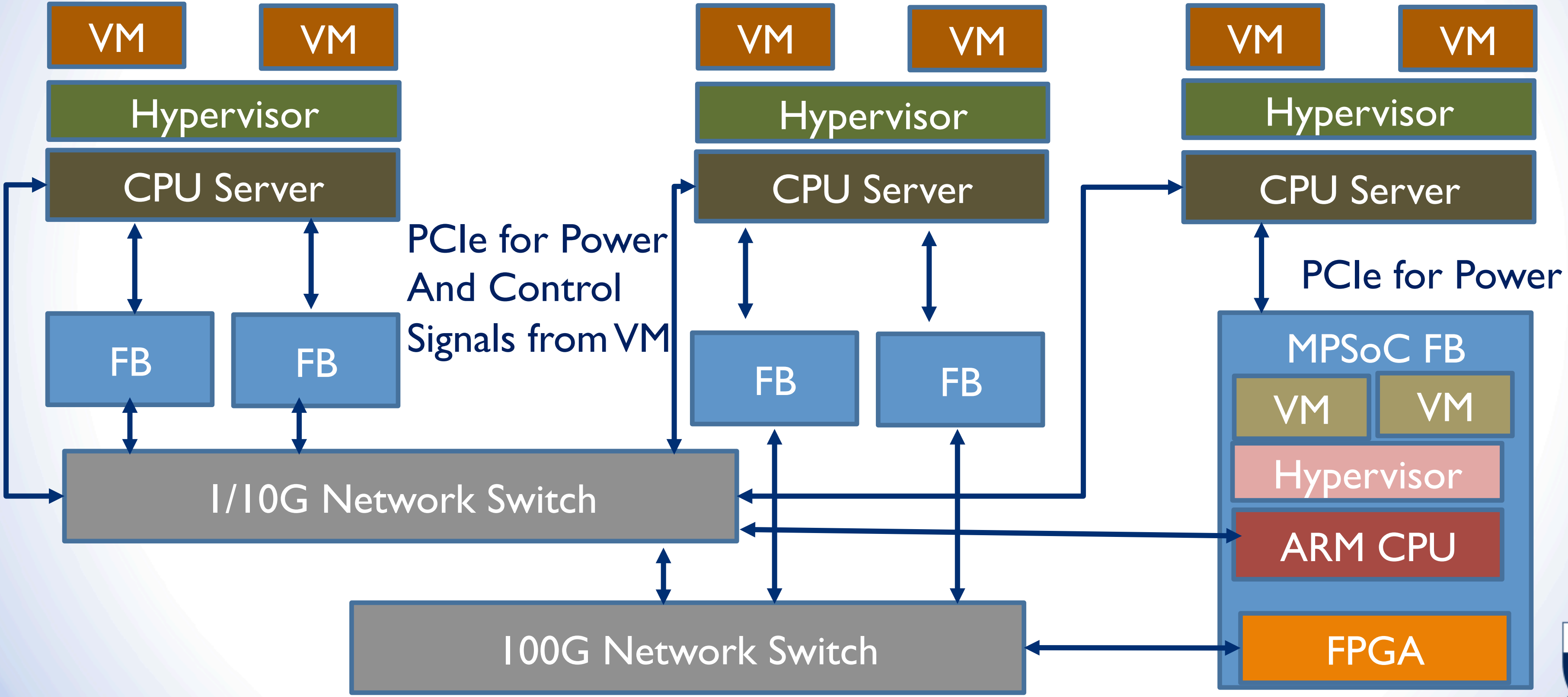
HOW DO WE GET THERE?

Start with Software Ecosystem

- Build from software as much as possible
  - Already lots of knowledge and infrastructure
- OpenStack is starting point for several groups
  - Cloud resource management
  - IBM, Huawei, UofT
- Virtualization
  - Means many things!
  - Sharing, abstraction

U OF T WORK

Our Physical Architecture

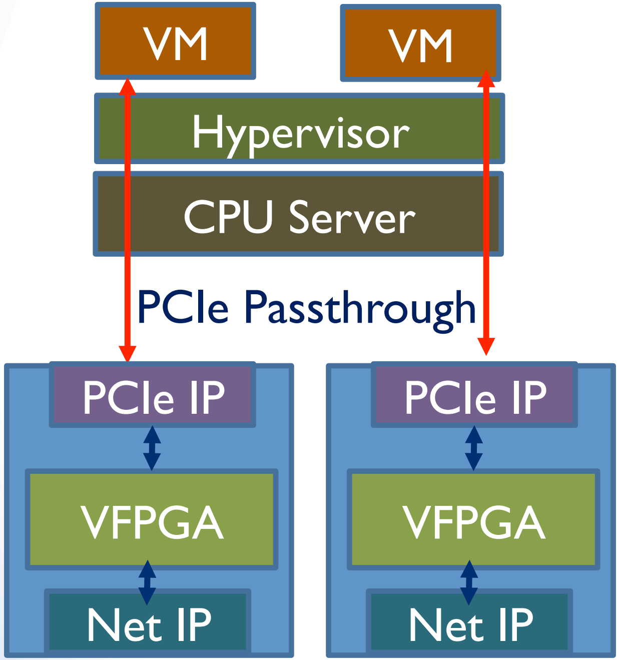


FPGA Boards in Cluster

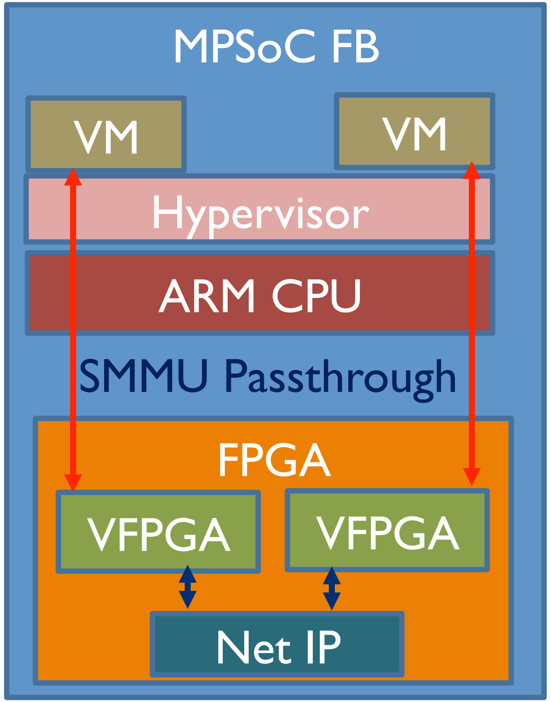
- Alphadata 7v3 * 4 – Virtex 7 -690T
  - ~690K Logic cells, 3600 DSP Slices , 52.9 MB BRAM
  - 2 * 10G SFP Networking ports
  - 16 GB off-chip memory
- Alphadata 8v3 * 8 – Virtex Ultrascale – XCV095
  - ~1176K Logic cells, 768 DSP Slices, 60.8 MB BRAM
  - 2 * 100G QSFP Networking Ports
  - 16 GB off-chip memory
- Alphadata 8k5 *8 –Virtex Ultrascale – KU115
  - ~1450K Logic cells, 5520 DSP Slices,
  - 2 * 10G SFP Networking ports
  - 16 GB off-chip memory
- Fidus Sidewinder * 20 – Zynq Ultrascale + MPSoC
  - ARM A53 (4 core) + ARM R5 (2 core)
  - ~1140 Logic Cells, 1910 DSP Slices, 128 MB BRAM
  - 2 * 100G QSFP Networking Ports
  - 16 GB off-chip DRAM for ARM, 16 GB off-chip DRAM for FPGA

FPGA Hypervisors

Non-MPSoC



MPSoC (In Progress)



ENABLING FLEXIBLE NETWORK FPGA CLUSTERS IN A HETEROGENEOUS CLOUD DATA CENTER

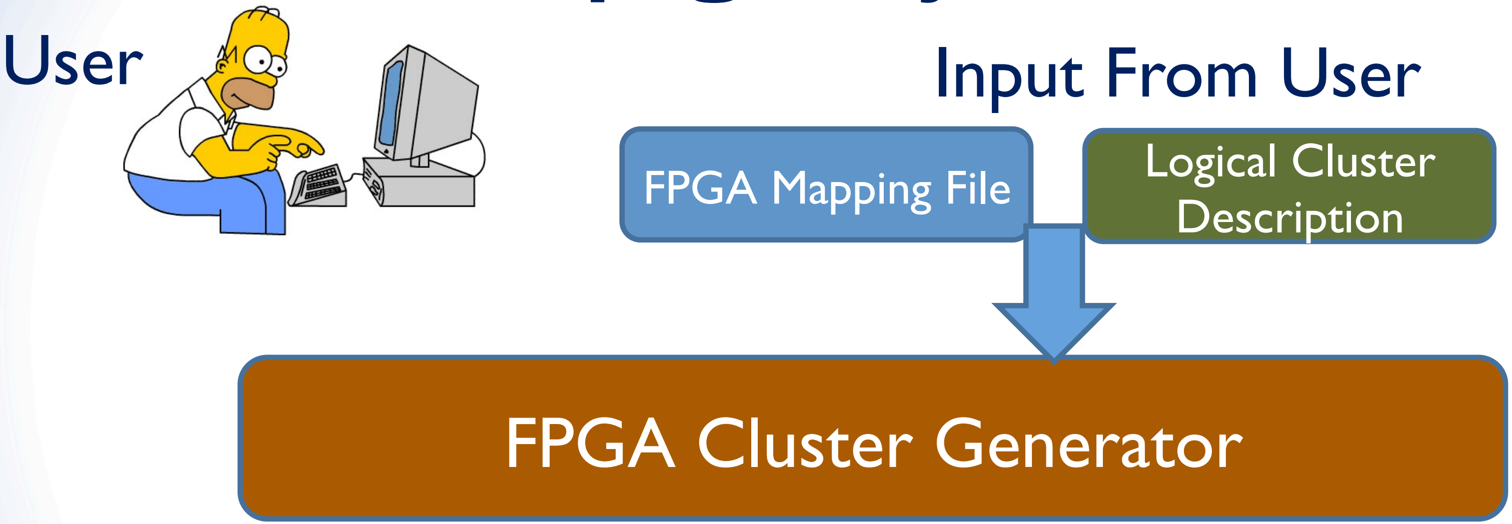
Naif Tarafdar, Thomas Lin, Eric Fukuda,
Hadi Bannazadeh, Alberto Leon-Garcia, Paul Chow
University of Toronto

FPGA 2017

Problems We Target

- Large multi-FPGA systems
- Abstractions for building applications on FPGA clusters
  - User provides application
  - We build the cluster and deploy it with the application
- Easy scalability of system

Galapagos System View



User

Input From User

FPGA Mapping File

Logical Cluster Description

FPGA Cluster Generator

Galapagos System View

User

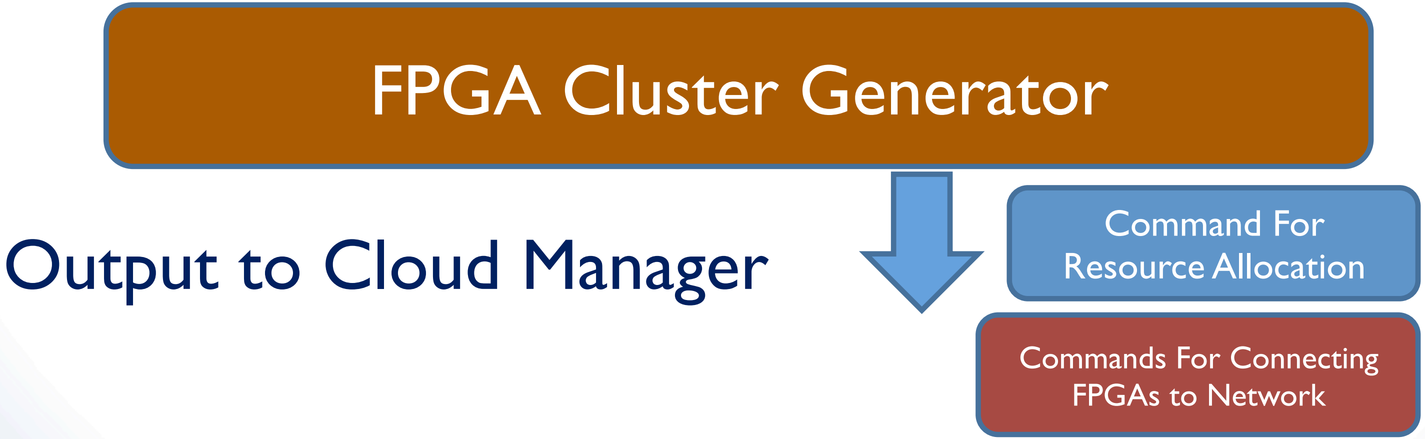
FPGA Cluster Generator

Output to VM with FPGA Tools

Individual FPGA Projects

Galapagos System View

User



FPGA Cluster Generator

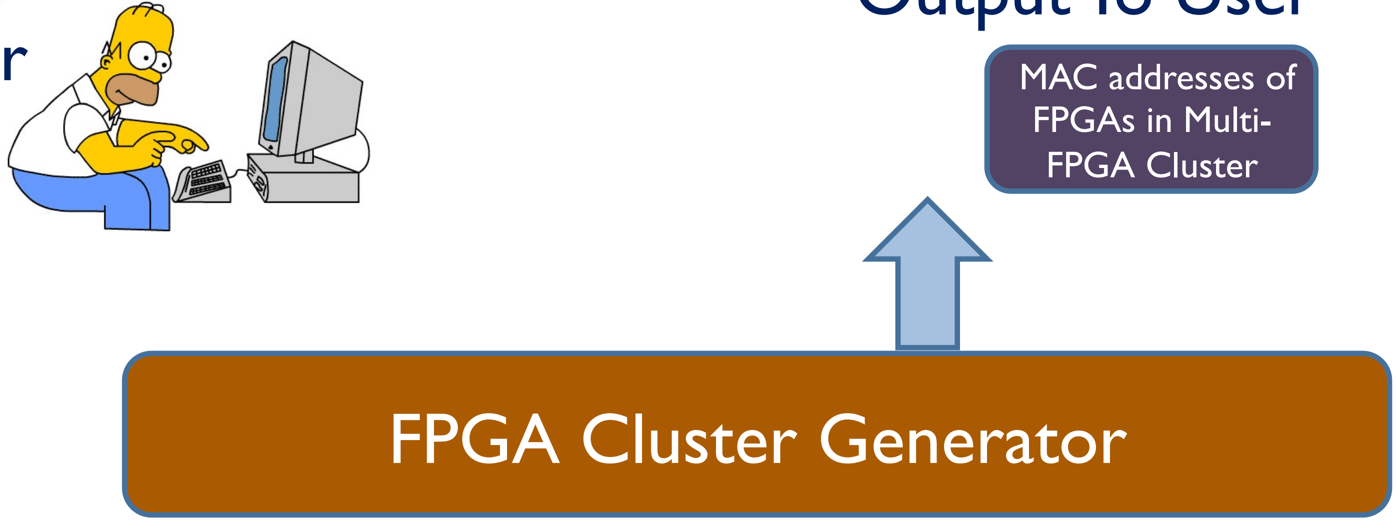
Output to Cloud Manager

Command For Resource Allocation

Commands For Connecting FPGAs to Network

Galapagos System View

User



Output To User

MAC addresses of FPGAs in Multi-FPGA Cluster

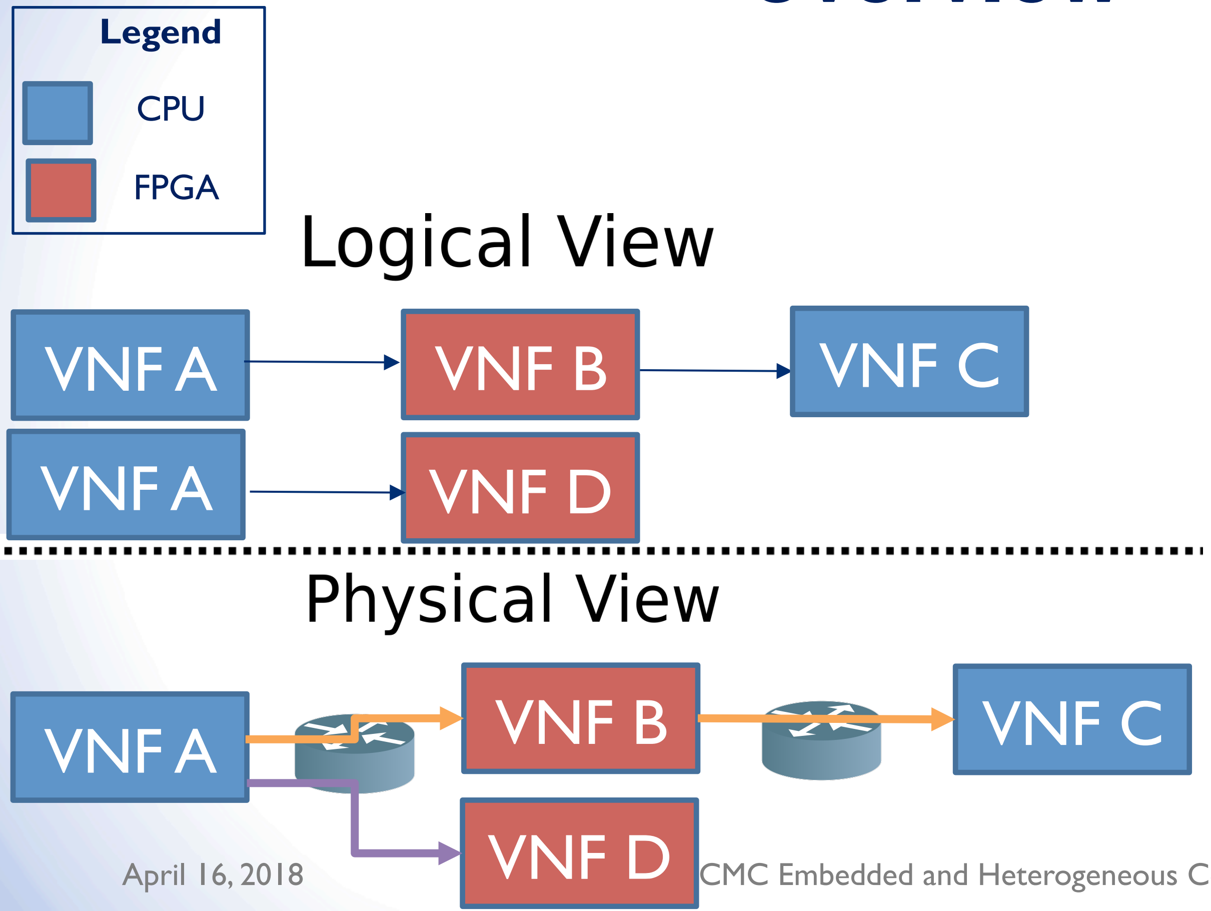
FPGA Cluster Generator

HETEROGENEOUS VIRTUALIZED NETWORK FUNCTION FRAMEWORK FOR THE DATA CENTER

Naif Tarafdar, Thomas Lin, Nariman Eskandari,
David Lion, Alberto Leon-Garcia, Paul Chow
University of Toronto

FPL 2017

Overview



-Circuit switched network

-Circuit includes CPU and FPGA

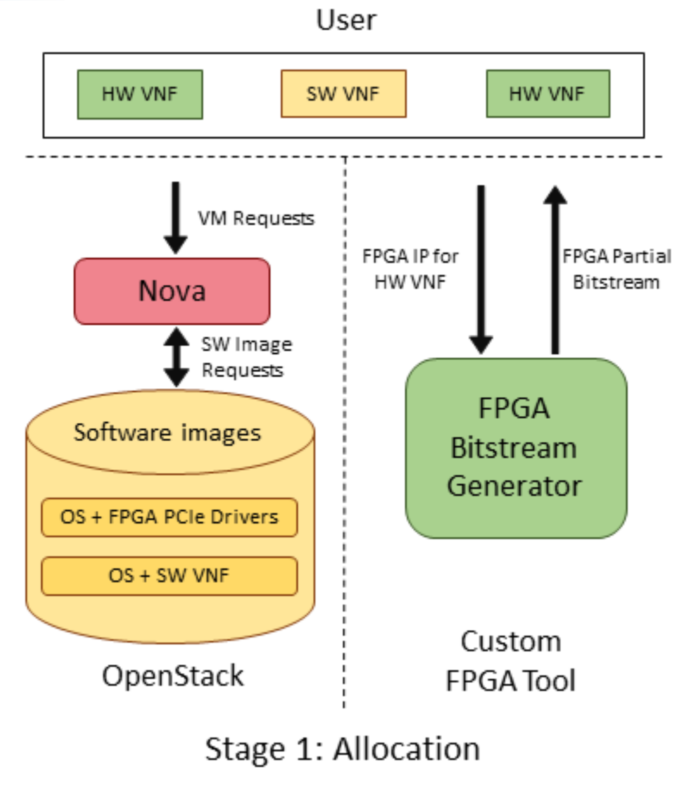
-Kernels are physically distributed

-Individual FPGAs provisioned with Openstack, along with network

-Shell abstracting network (10G) and PCIe

-Partial Reconfiguration flow

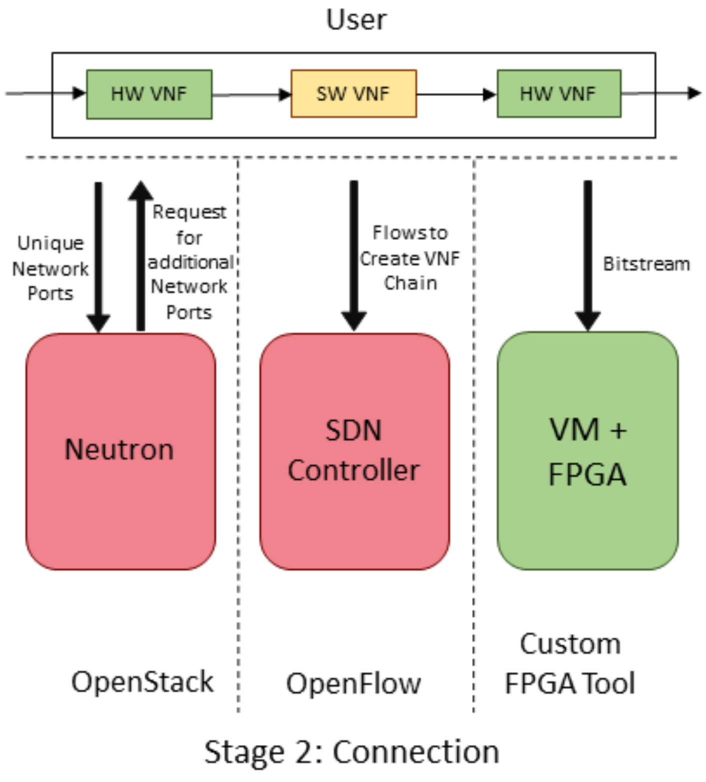
Service Chain Scheduler



Stage 1: Allocation

- Resource allocation
  - OS image
  - Parameters: cores, PCIe devices, NIC ports
- Bitstream generator

Service Chain Scheduler



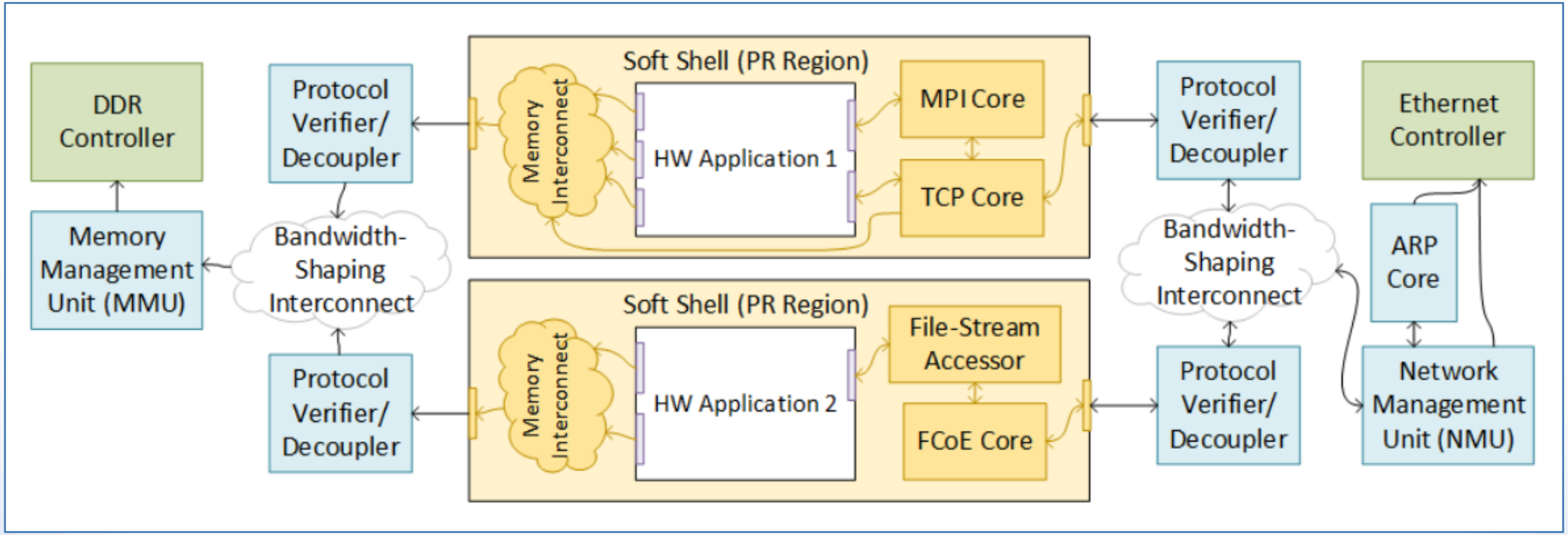
Stage 2: Connection

- For each FPGA, assign and register virtual port
- Create chain between source and sink of network and intermediate VNFs

Daniel Rozhko

MULTI-TENANT HYPERVISOR (SHELL)

High-Level (Long-term Plan)


DDR Controller
Protocol Verifier/ Decoupler
Soft Shell (PR Region)
Memory Interconnect
HW Application 1
MPI Core
TCP Core
Protocol Verifier/ Decoupler
Ethernet Controller
Memory Management Unit (MMU)
Bandwidth-Shaping Interconnect
Bandwidth-Shaping Interconnect
ARP Core
Soft Shell (PR Region)
Memory Interconnect
HW Application 2
File-Stream Accessor
FCoE Core
Protocol Verifier/ Decoupler
Protocol Verifier/ Decoupler
Network Management Unit (NMU)

Key Components

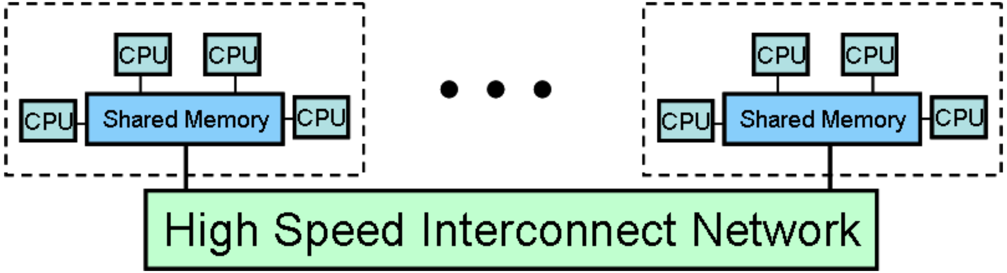
- Virtualized access to external I/O (i.e. abstracted, shared, and secured)
- Soft vs. Hard shell distinction

Nariman Eskandari

A HETEROGENEOUS MPI

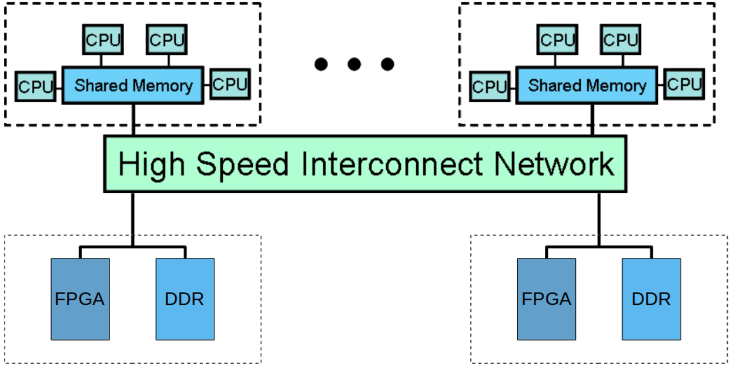
MPI

- Message Passing Interface
- Used as a programming model for HPC



MPI

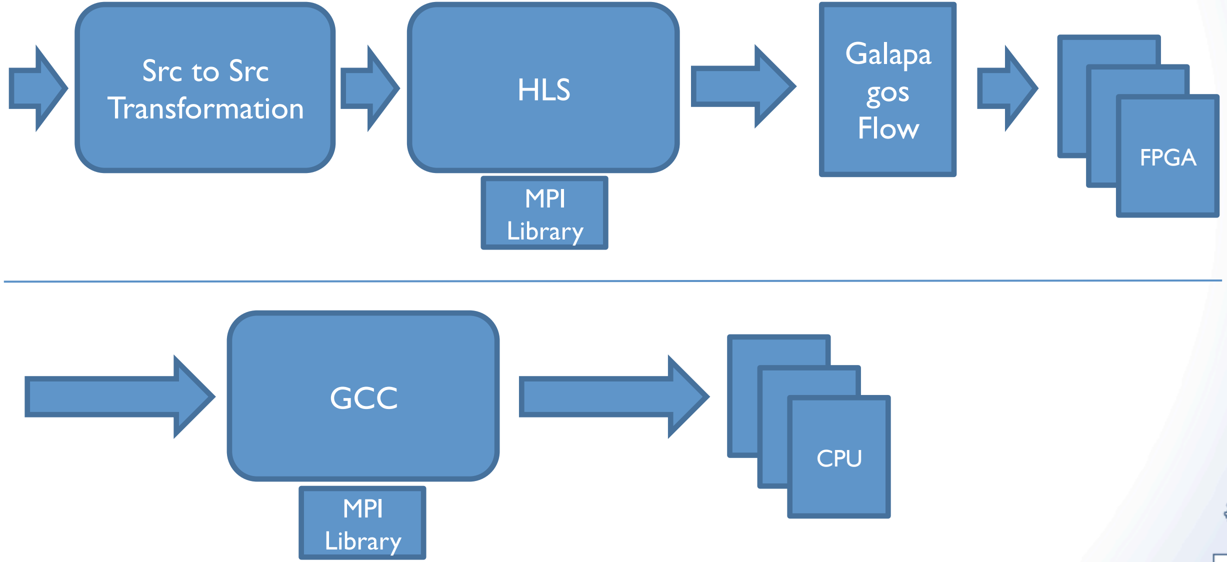
- Another abstraction layer on Galapagos
- MPI in this work is programming model for heterogenous platform (CPUs and FPGAs)



MPI

- The code for FPGAs and CPUs are the same.

```
Main(){
    MPI_Init();
    If(rank == 0){
        MPI_Send();
    }else{
        MPI_Recv();
    }
}
```



MPI

- First applications – Jacobi, MD, K-means
  - 1 to 90 ranks tested on 6 FPGAs

Conclusions

- Lots of focus on HLS today – it's needed, not sufficient
- Some working now on other layers – need identified
- To achieve a cloud ecosystem for using FPGAs, much more is needed – it's a big stack
- Need a coordinated effort to enable cloud computing with FPGAs – cannot be haphazard → need a plan
  - Open source is only way to harness enough resources
  - How do we do this?

Acknowledgements

XILINX

HUAWEI

NSERC CRSNG

CMC MICROSYSTEMS

Ontario Centres of Excellence
Where Next Happens

emSYSCAN

Stuart Byma, Naif Tarafdar, Eric Fukuda, Daniel Ly-Ma, Daniel Rozhko, Roberto DiCecco, Nariman Eskandari,

Qiang Liu, Clark Shen, Charles Lo, Varun Sharma, Sean Nijjar, Camilo Vega

SAVI – Prof. Alberto Leon-Garcia, Hadi Bannazadeh, Thomas Lin

Questions?