ESSPER
Elastic and Scalable System for High-Performance Reconfigurable Computing

RIKEN R-CCS PROCT

System and Applications of FPGA Cluster "ESSPER" for Research on Reconfigurable HPC

Kentaro Sano

RIKEN Center for Computational Science

Mar 18, 2022

Introduce Myself : Kentaro Sano

Hiring researchers:
R-CCS2105 or R-CCS2022

RIKEN Center for Computational Science

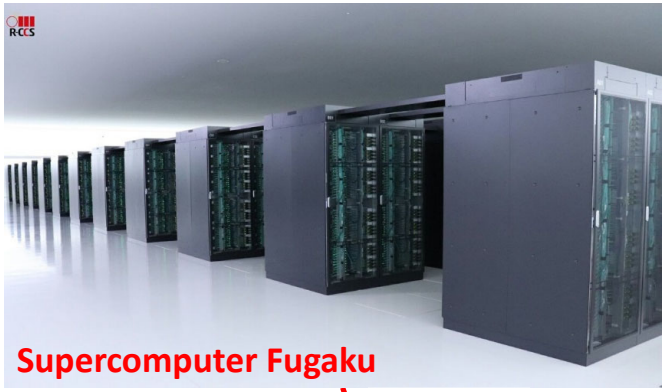
- ✓ Develop and operate Supercomputer Fugaku
- ✓ Facilitate leading edge infrastructures for research based on supercomputers
- ✓ Conduct cutting-edge research on HPC

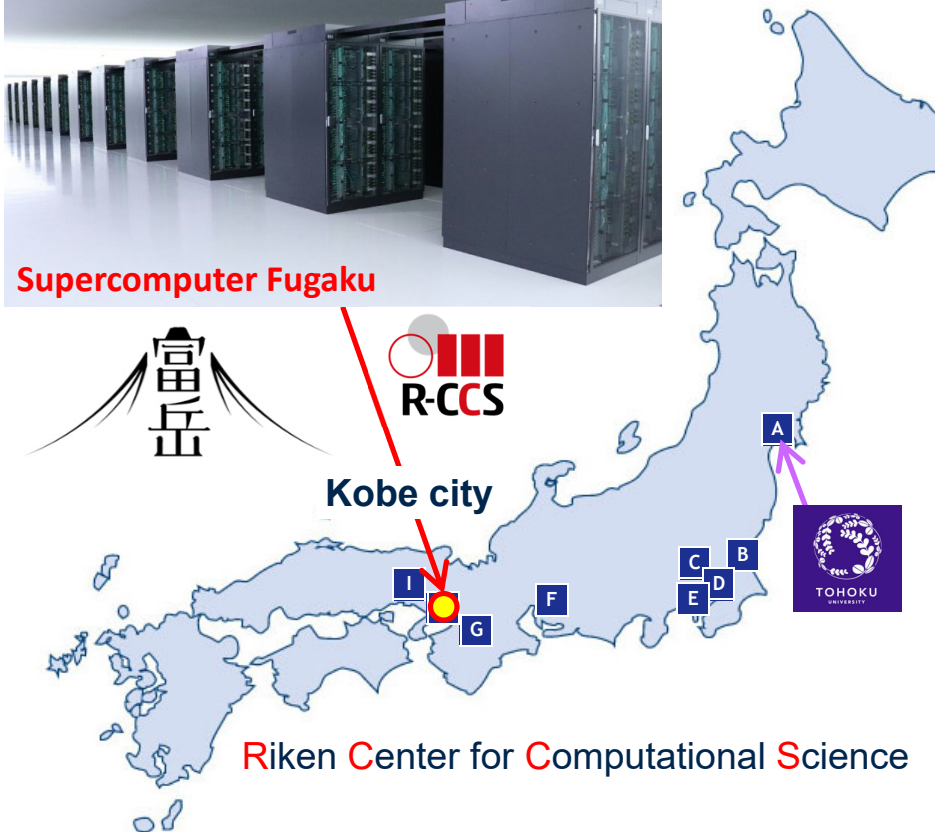
Leader, Processor Research Team

- ✓ Exploration of future HPC architectures
- ✓ Advanced use of present HPC systems

Joint Laboratory at Tohoku University

- ✓ Visiting Professor
  "Advanced Computing Systems Lab"





Goal and Roadmap of the Team

Establish HPC architectures suitable for Post-Moore Era

2020 — 2021 — 2022 — 2023 — 2024 — 2025 — 2026 — 2027

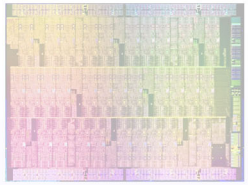
Advancement of Supercomputers

- ✓ Functional extension with FPGAs and system stacks
- ✓ System software and apps of task-flow computing

This talk

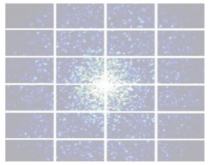
Exploration of HPC Architectures

- ✓ Novel processors based on CGRA / Data-flow architectures
- ✓ New organization of computing nodes with CPU and novel processors



Near-sensor / Near-storage Processing

- ✓ FPGA-based processing for X-ray imaging detector

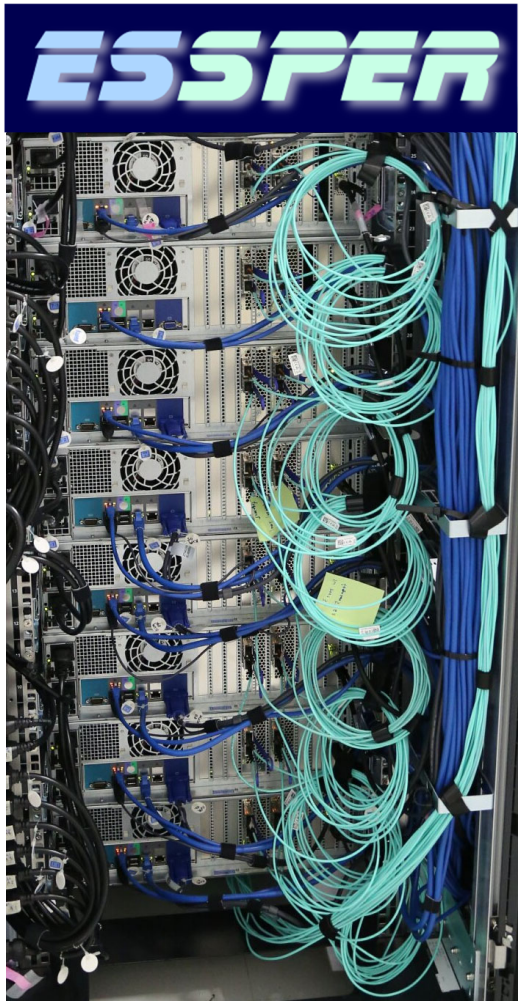


Exploration of Novel Computing Principle and Model

- ✓ Neuromorphic computing and its applications to CPS

Outline

- Introduction
- Motivation and Challenges of HPC with FPGAs
- **ESSPER :** Proof-of-Concept FPGA Cluster System
- Summary



PoC FPGA Cluster System

Introduction

[https://top500.org]

World Ranking of Supercomputers

Ranking of HPL performance

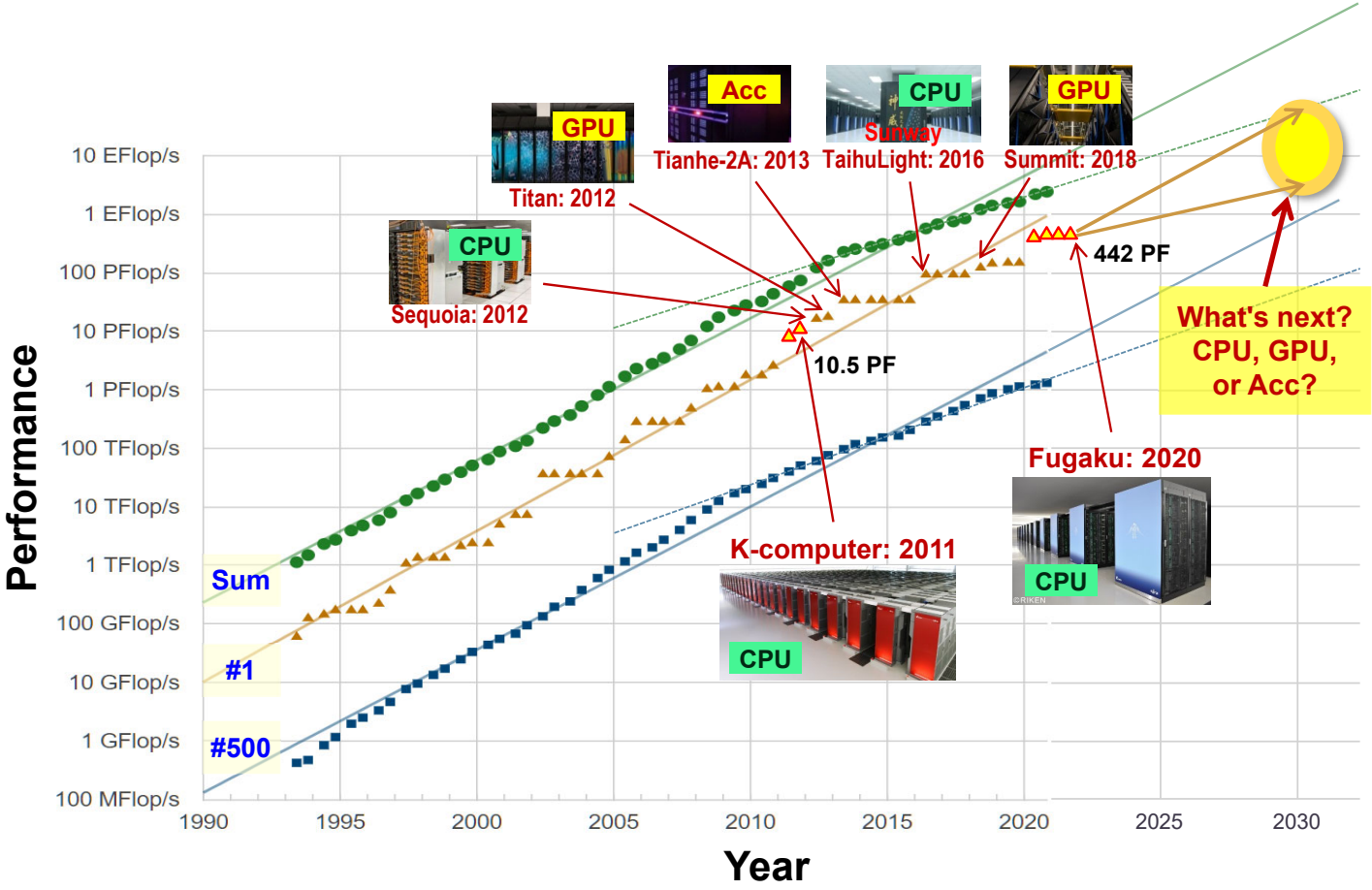
- ✓ Linear algebra (LINPACK)
- ✓ Distributed-memory parallel computers

Supercomputer Fugaku

- ✓ #1 in TOP500
- ✓ #1 in HPCG, HPL-AI, Graph500
- ✓ #26 in Green500

Trend of HPC Technology

- ✓ Advancement slowing down?
- ✓ Difficulties in Moore's law?
- ✓ CPU vs. GPU/Accelerator?



Constraint : System Power Consumption

Average power consumption

- ✓ in TOP10, TOP50, TOP500

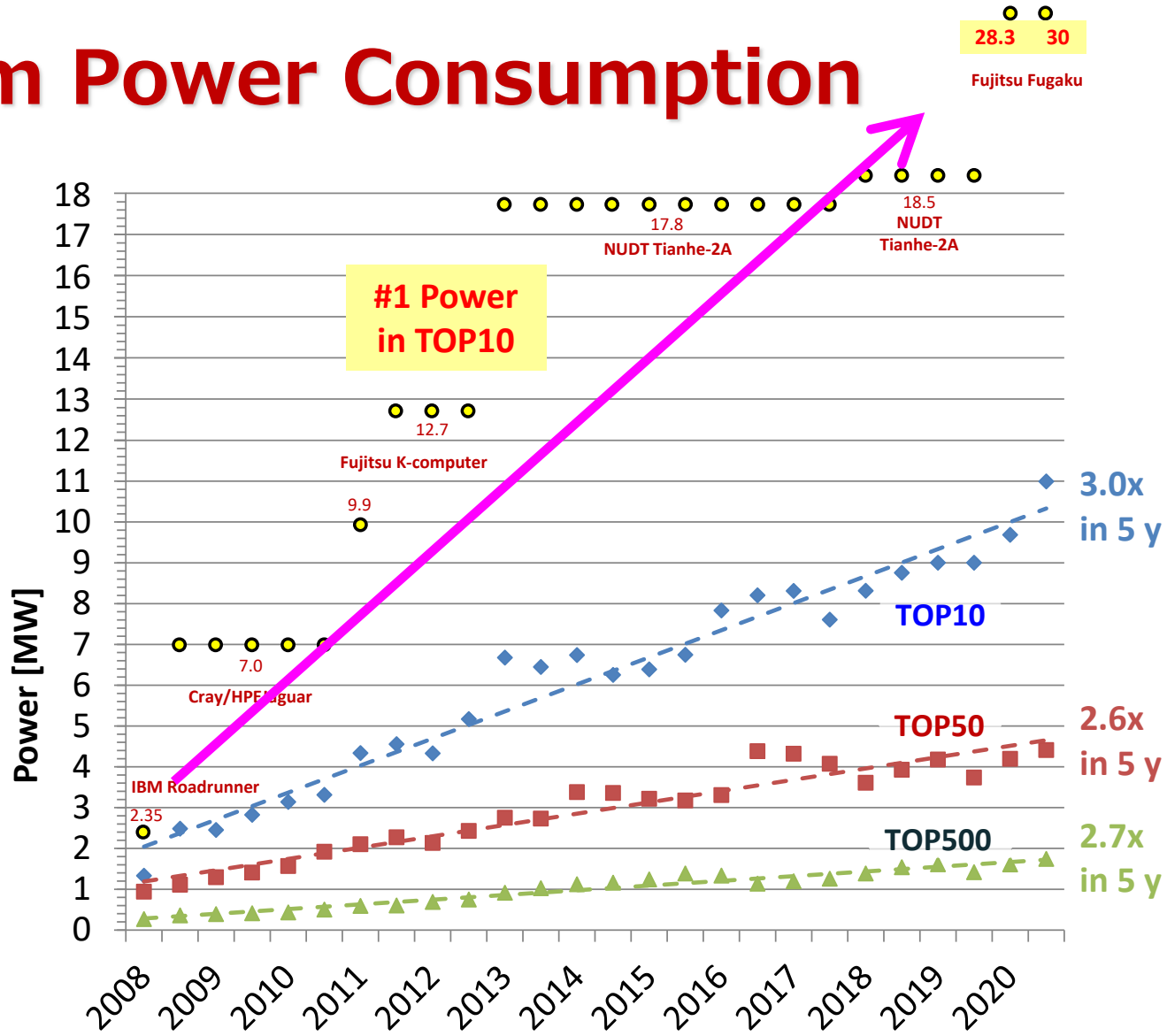
Needed to increase for higher system performance

- ✓ Limited improvement of performance per power

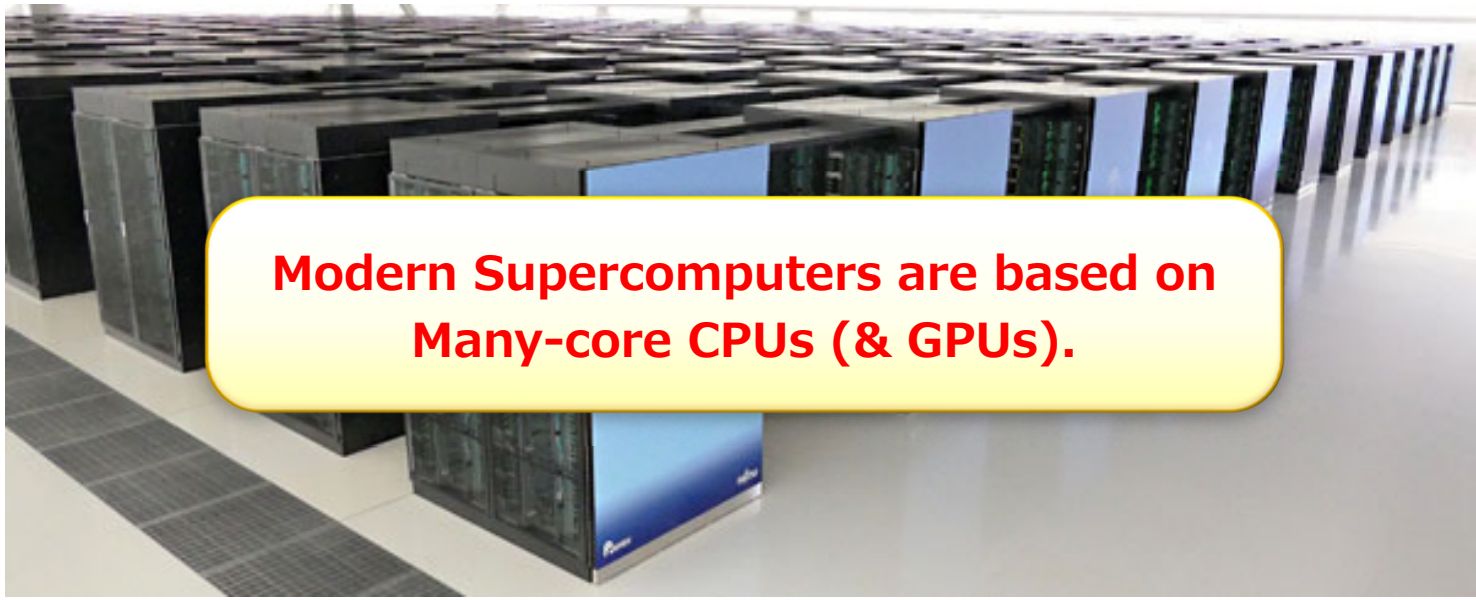
10s of MW for #1 systems

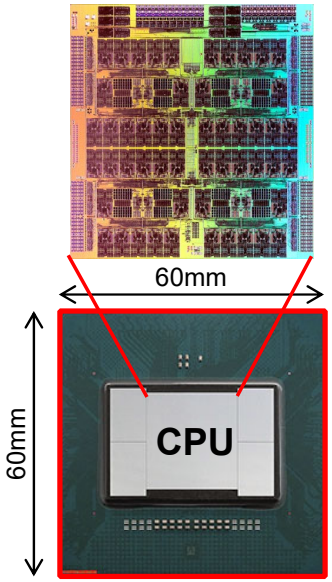
- ✓ **30MW** for 442PF (HPL) with 7,630,848 cores in Fugaku

System power budget
= **Critical constraint** of system performance

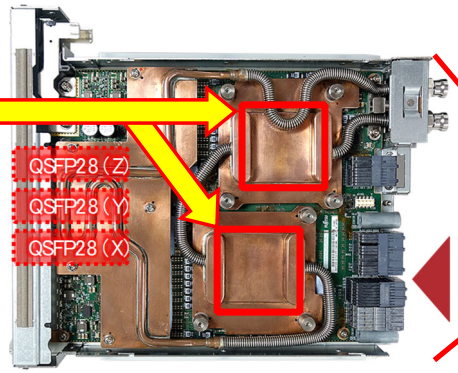


System Configuration of Fugaku

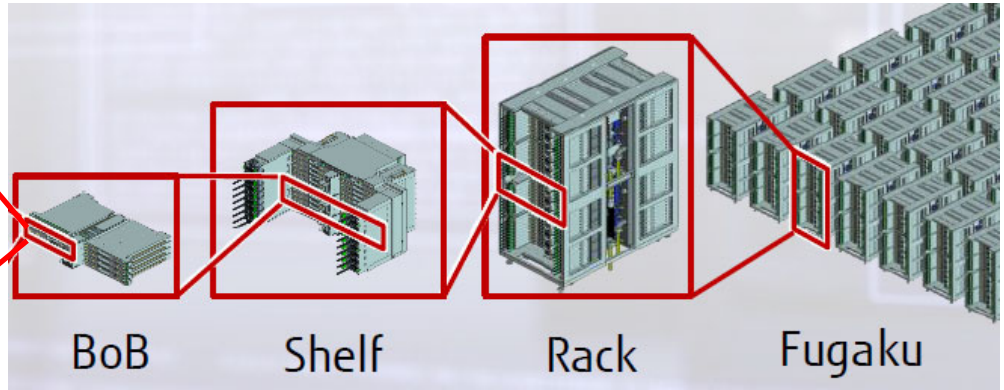




48+ cores / 1 node
2.7+ TF



CPU-Memory Unit (CMU) 2 nodes
5.4+ TF



BoB	Shelf	Rack	Fugaku
16 nodes 43+ TF	48 nodes 129+ TF	384 nodes 1+ PF	158,976 nodes 537 PF @ FP64 (414 racks)

Photos & figs by Fujitsu

Does Many-core Scale?

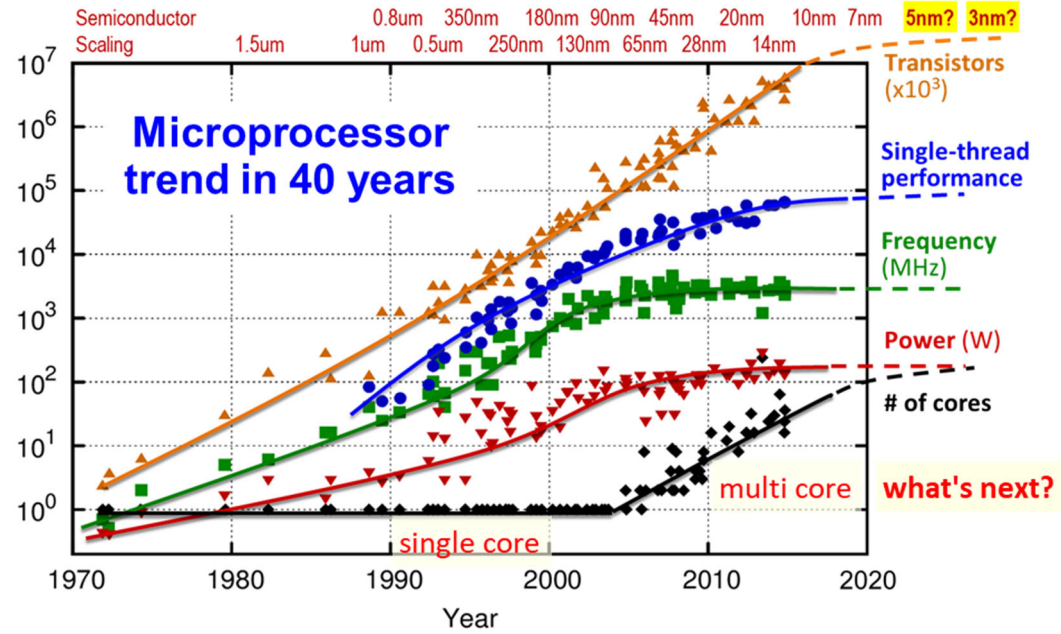
Present mainstream : many-core

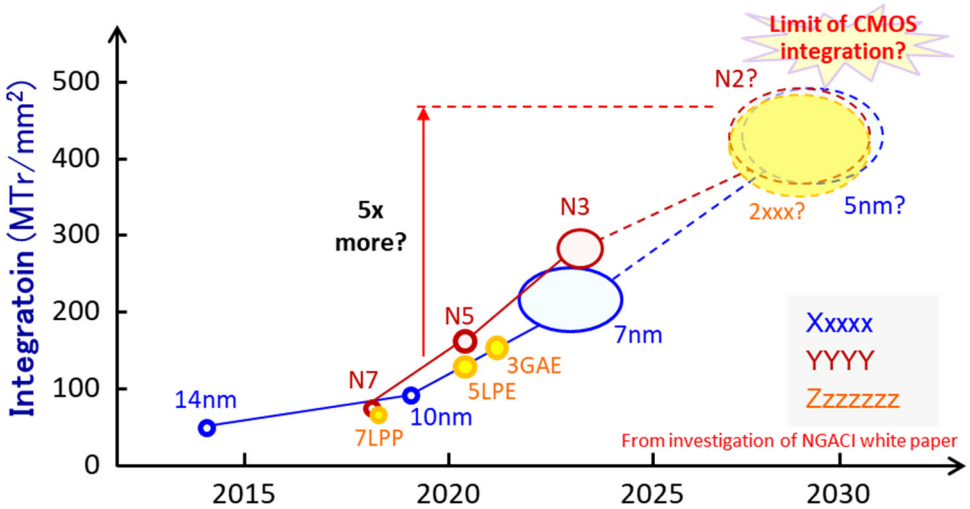
(perf) = (# cores) x (freq) x (utilization)

Technology trends

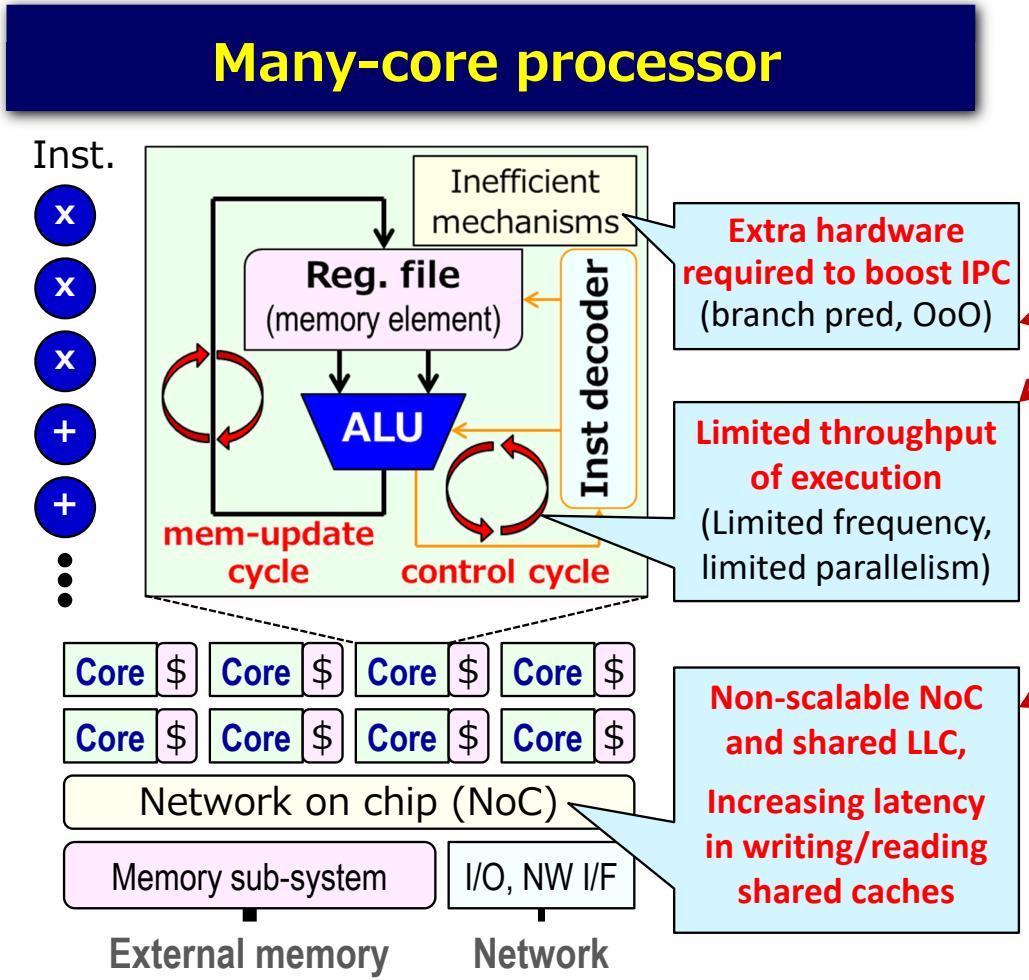
- ✓ **# of cores** ↗ Fin FET -> GAA FET
- ✓ **Frequency** → End of Dennard scaling
- ✓ **Utilization ?**
- ✓ **Performance per power ?**

More cores, higher performance?





Many-Core is Difficult to Scale!



Core performance is already difficult to scale.

No big improvement of
single-core performance and performance per power.

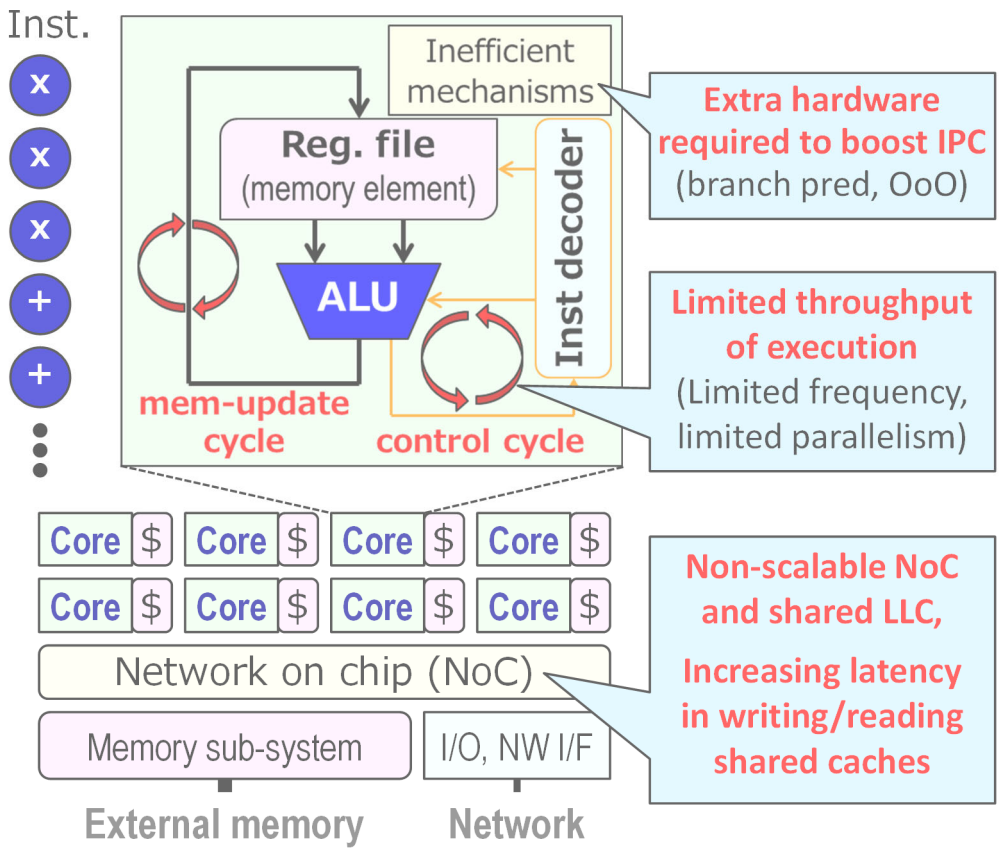
Entire performance doesn't scale even if we could have more cores.

Inter-core data movement is getting less efficient and becoming a bottleneck in parallel computing with many core.

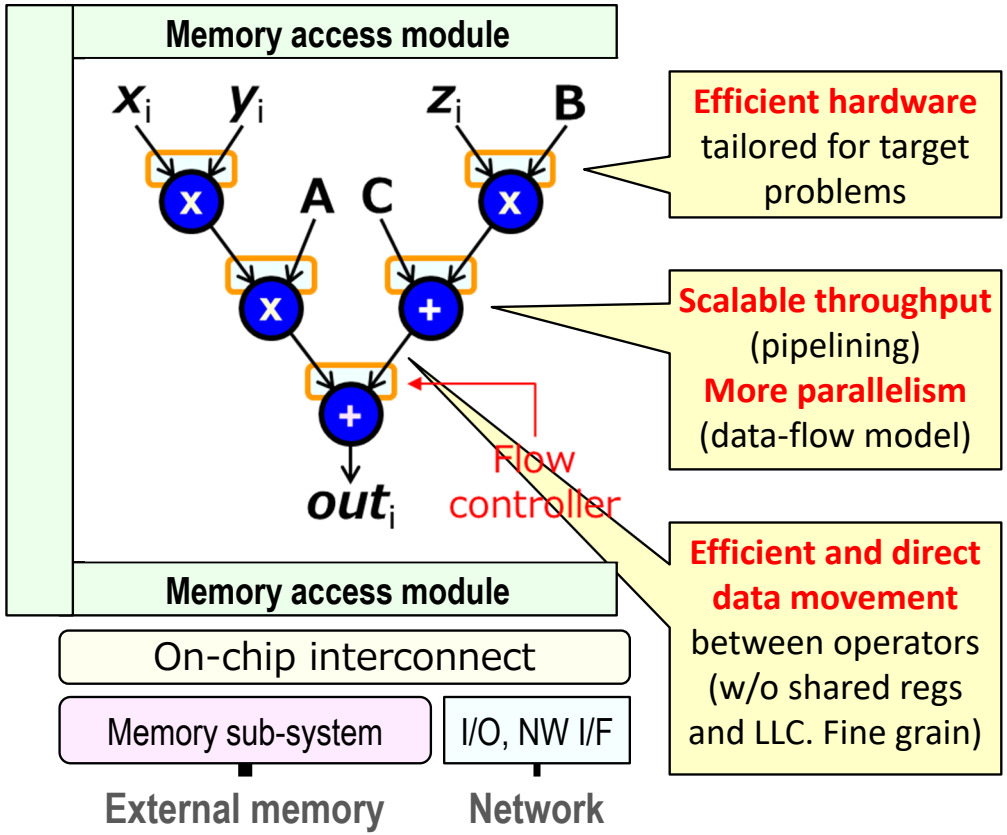
No more improvement in performance/power.

Solution : Custom Data-Flow Computing

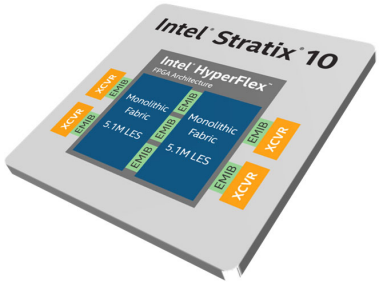
Many-core processor



Custom data-flow processor



FPGA as Platform for Custom DFC.



The state-of-the-art FPGA

- ✓ High-performance operation
- ✓ High-bandwidth external memories
- ✓ Ultra high-bandwidth on-chip memories
- ✓ Fast inter-device communication

Intel Stratix 10 FPGA (14nm)

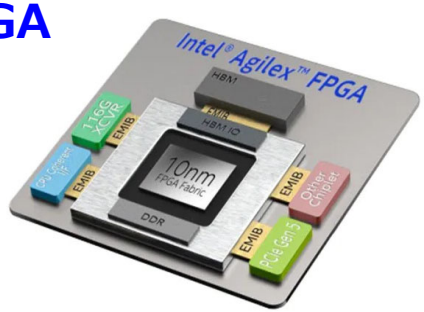
- ➢ 5760 floating-point DSPs
- ➢ comparative to CPU, GPU (DDR4, HBM2)
- ➢ aggregate ~1000 TB/s
- ➢ multiple tx / rx of 100 Gbps

Use cases in data-center, cloud, or HPC systems

- ✓ *Microsoft Catapult, AWS EC2 F1, Alibaba Cloud, Tencent Cloud, Huawei Cloud*
- ✓ *Tsukub U Cygnus, Paderborn U Noctua*

Intel Agilex FPGA

More advanced next-generation 10nm, or 7nm



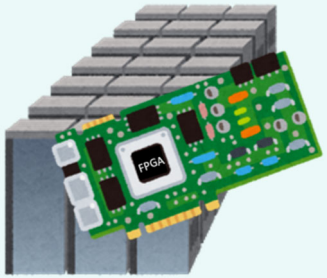
Promising not only for computing, but also for data movement

Challenges and requirements for HPC with FPGAs

Requirements for FPGA-based HPC System

Req. 1 Interoperability w/ various HPC systems

- ✓ Able to easily extend existing systems with FPGAs
- ✓ Can we extend Supercomputer Fugaku?

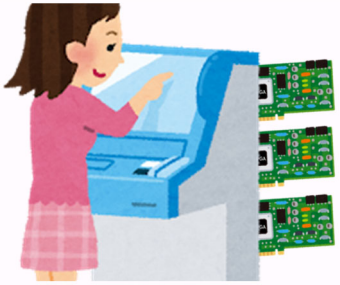


Req.2 Flexibility in using FPGAs in a system

- ✓ Allow any CPUs to flexibly utilize FPGAs in a system
- ✓ Appropriate for a machine shared with multiple users
- ✓ High utilization of FPGAs

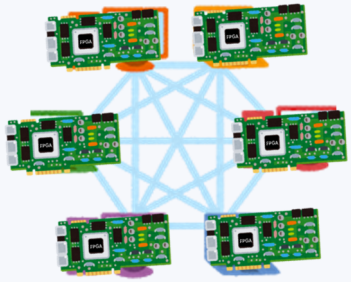
Req.3 Platform with sufficient customizability

- ✓ Able to implement various hardware (algorithms) on FPGA
- ✓ Give a high productivity by providing common SoC and its software abstraction



Req.4 Techniques for performance scalability

- ✓ Allow low-latency and high-throughput communication among FPGAs
- ✓ Allow users to easily try multi-FPGA applications



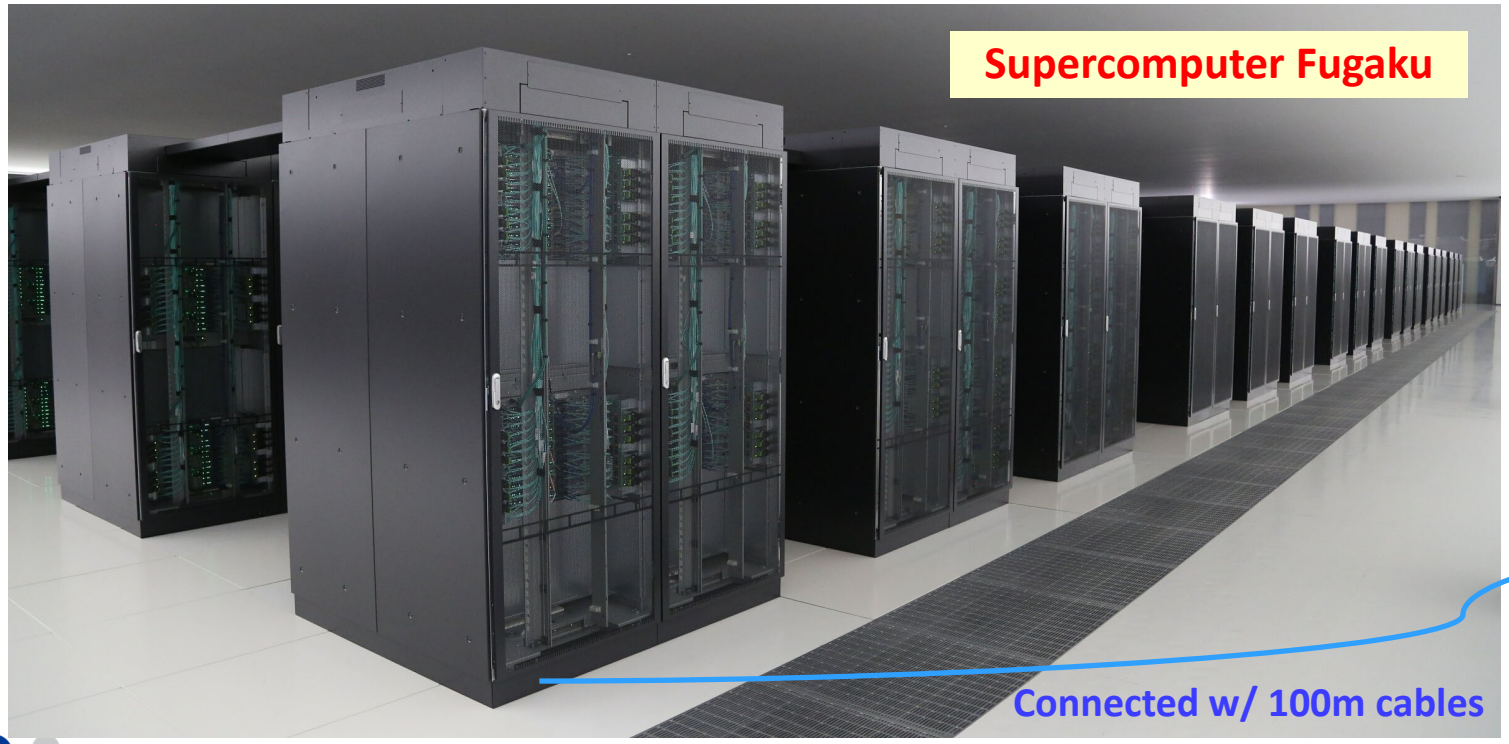
ESSPER

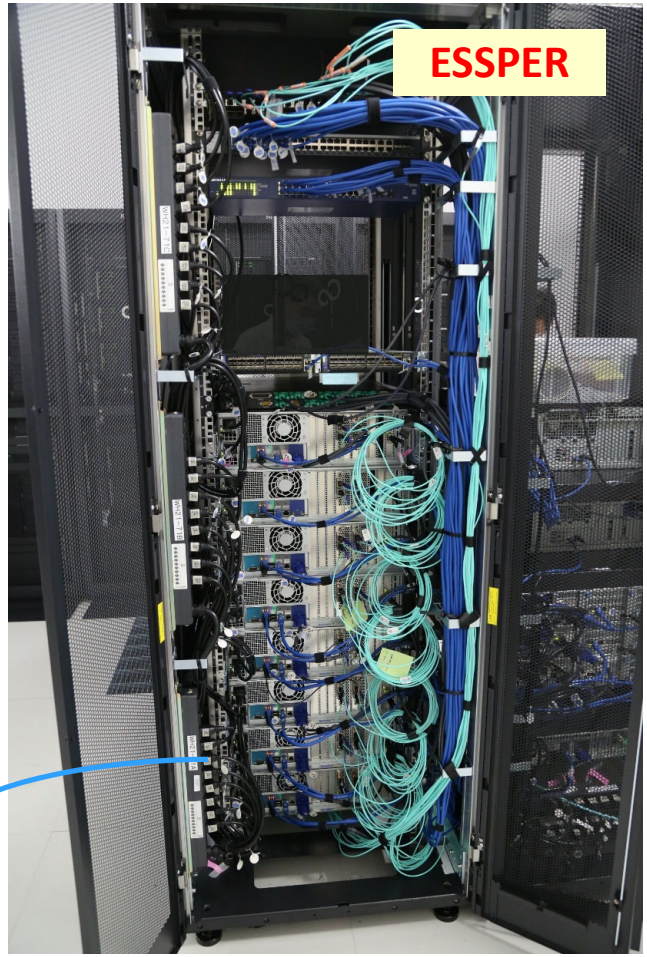
Elastic and Scalable System for High-Performance Re-configurable Computing

ESSPER : Proof-of-Concept FPGA Cluster System

Elastic and Scalable System for High-PErformance Reconfigurable Computing

Experimental prototype
for research on functional extension with FPGAs

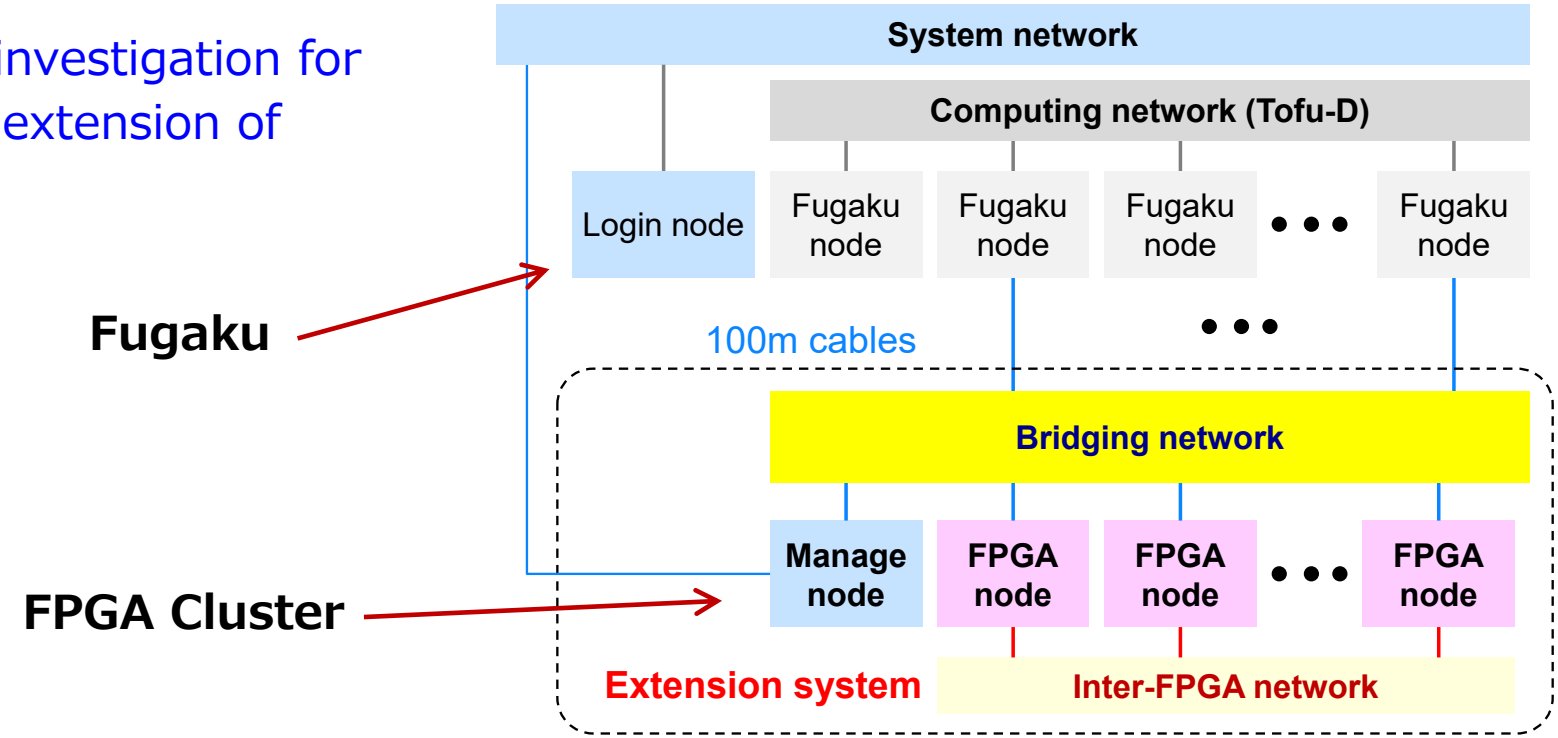




Architecture of ESSPER

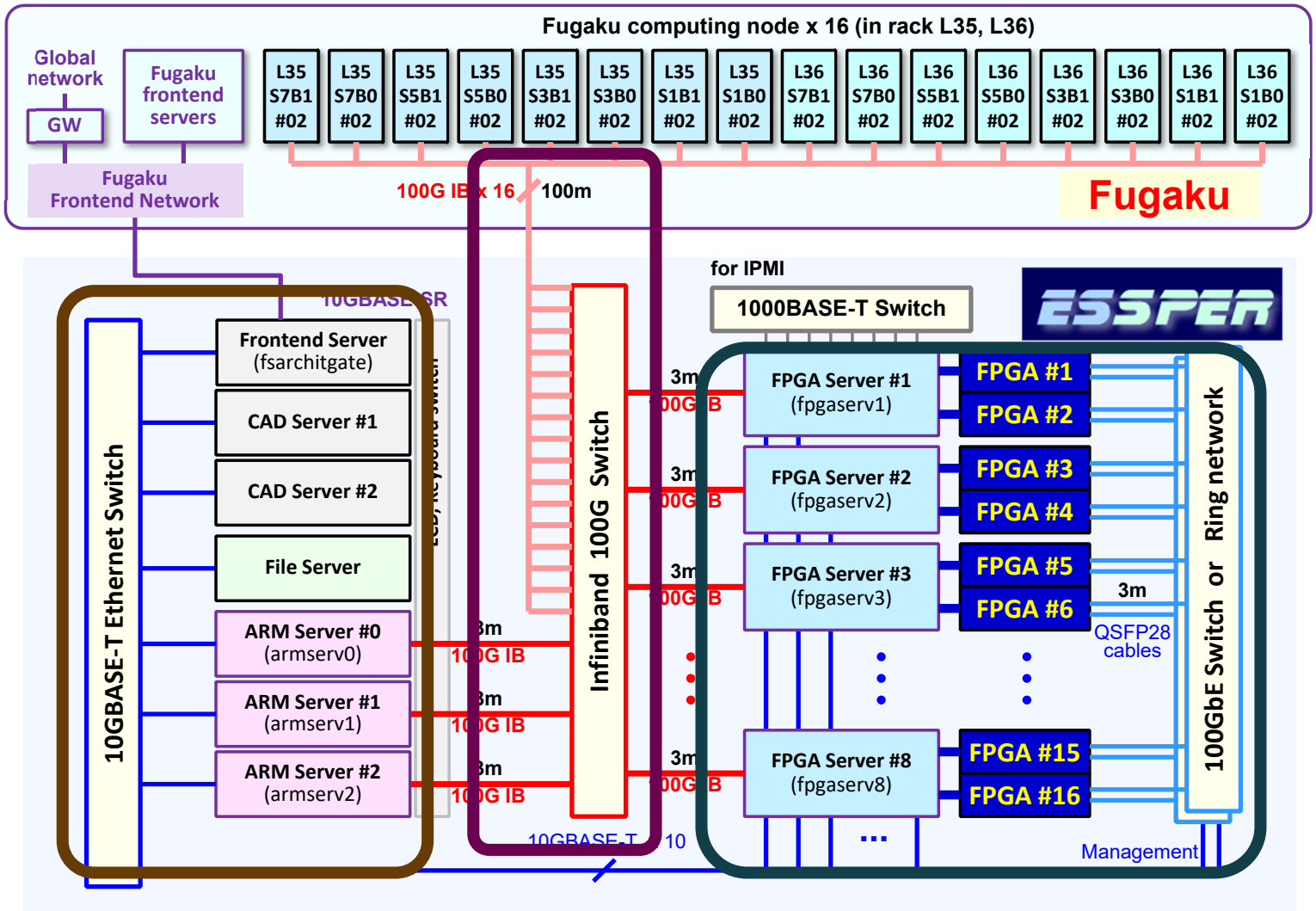
Goal

- ✓ Technical investigation for functional extension of Fugaku.



System Organization

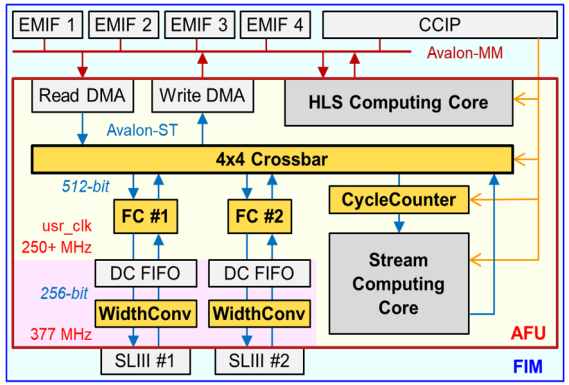
- FPGA Cluster
- CPU-FPGA bridging network
- Other servers

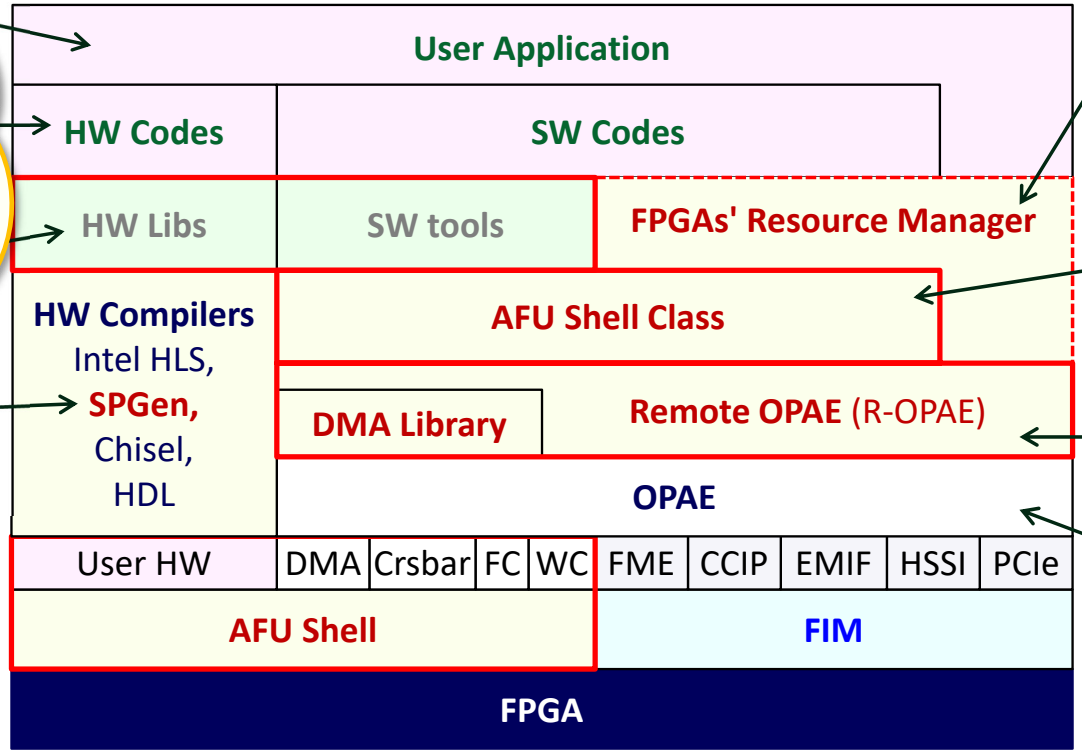


System Stack of ESSPER

Call for Joint researches:

- Applications
- Libraries for HW and SW
- Tools / system software
- Parallelization techniques with multi FPGAs





Resource manager

- Search and allocate resources of multiple FPGAs
- FPGA network management / control

AFU Shell class

- Object of AFU shell
- Abstraction of HW

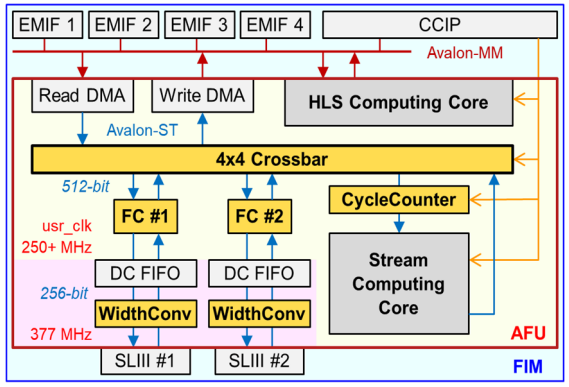
Remote OPAE

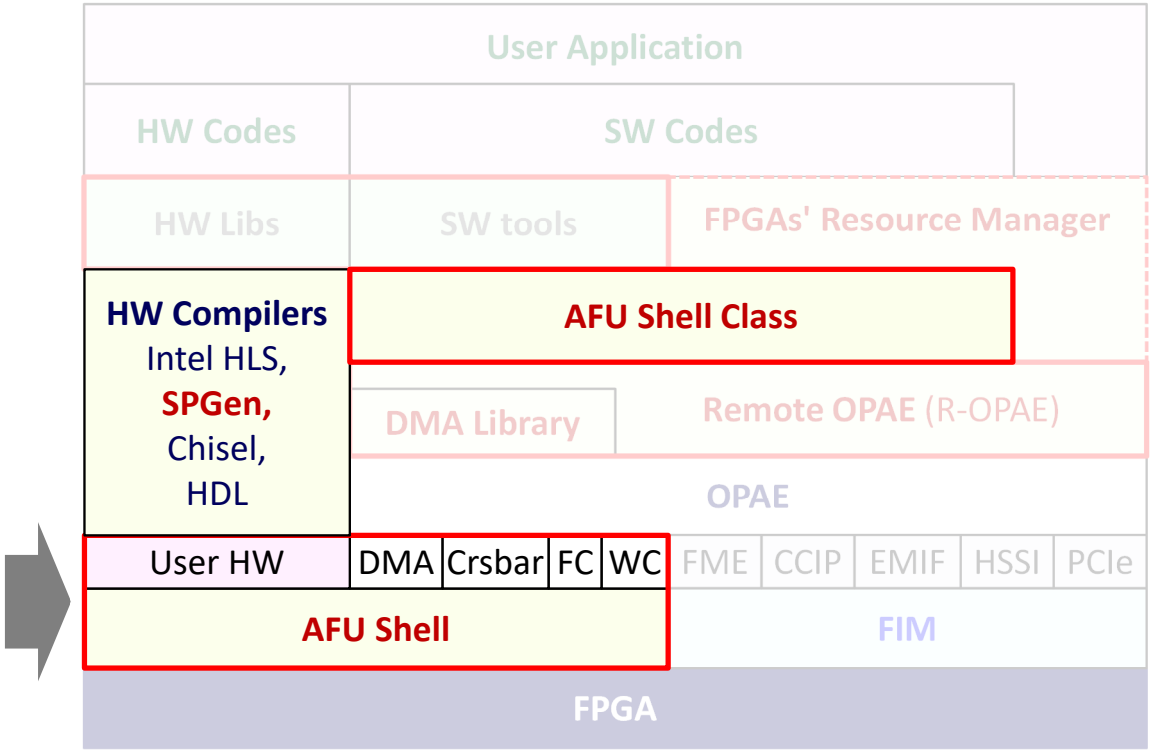
- Software bridge using Infiniband Verbs

OPAE

- Low-level driver

System Stack of ESSPER





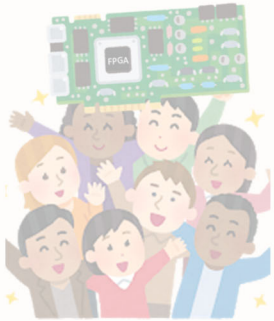
Approaches for Proof of Concept

Req. 1 Interoperability w/ various HPC systems

- ✓ Able to easily extend existing systems with FPGAs
- ✓ Can we extend Supercomputer Fugaku?

Req.2 Flexibility in using FPGAs in a system

to flexibly
stem
chine
e users
ion of FPGAs



Software-bridged FPGA driver

FPGA resource manager

Req.3 Platform with sufficient customizability

✓ Able

inte

Req.4 Techniques for performance scalability

✓ All

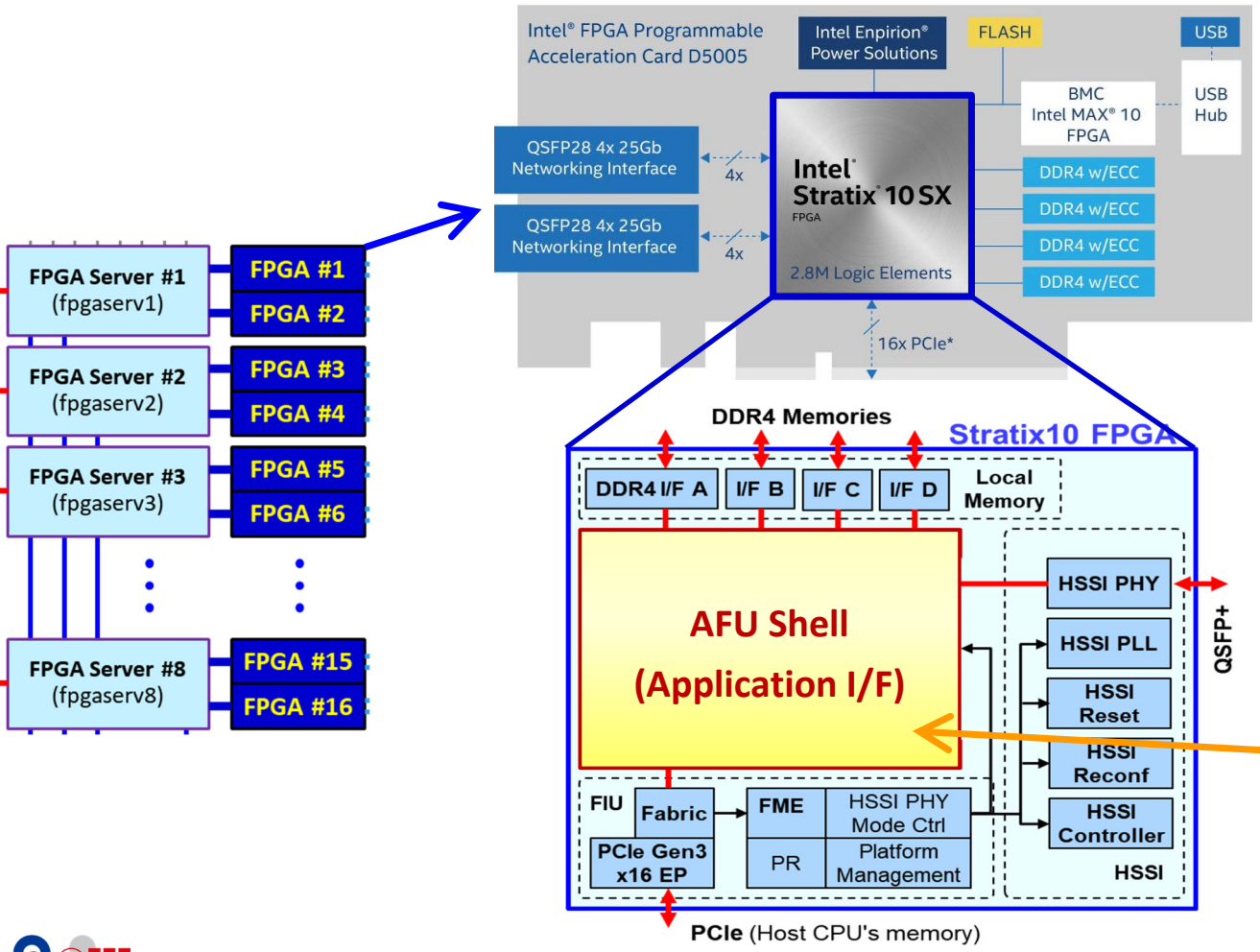
FPGA Shell (SoC)

HLS-based programming flow, & FPGA object lib

Inter-FPGA network

Virtual circuit switching over Ethernet

Design of FPGA System-on-Chip



Intel FPGA PAC D5005

- ✓ *Intel Stratix 10 FPGA (14nm)*
- ✓ *2753K LEs, 229 Mb BRAMs*
- ✓ *5760 FP DSPs (7TF @ 600MHz)*
- ✓ 8GB DDR4 x 4ch
- ✓ PCIe Gen3 x16
- ✓ 2x QSFP28 (100Gb/s)

FIM (FPGA Interface Manager)

- ✓ Fixed region including I/F

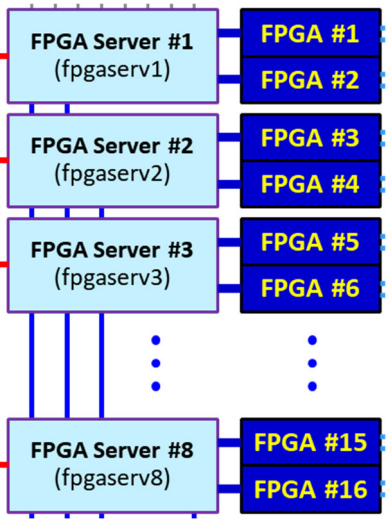
AFU (Acceleration Function Unit)

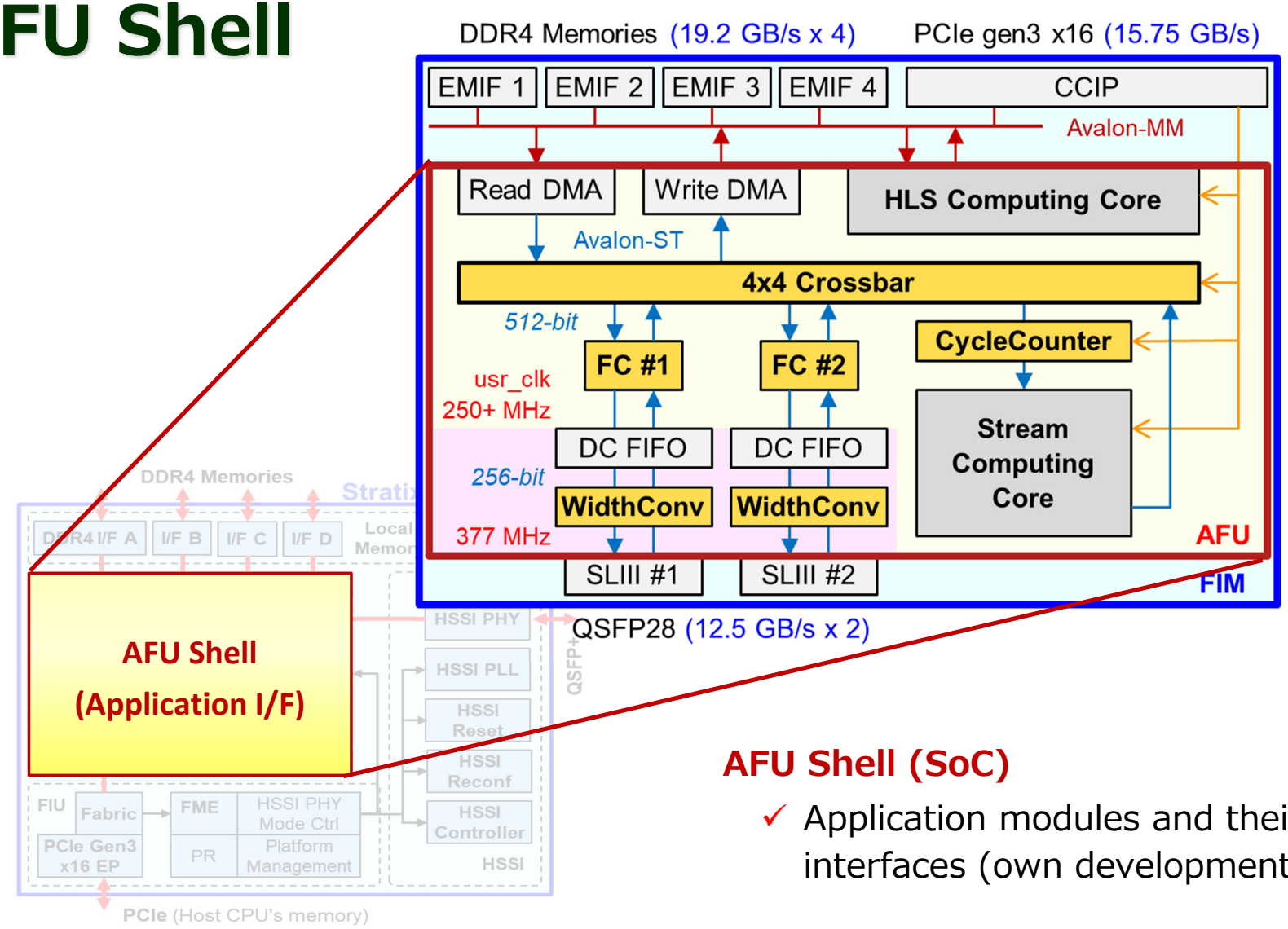
- ✓ Reconfigurable region

AFU Shell (SoC)

- ✓ Application modules and their interfaces (own development)

Design of AFU Shell





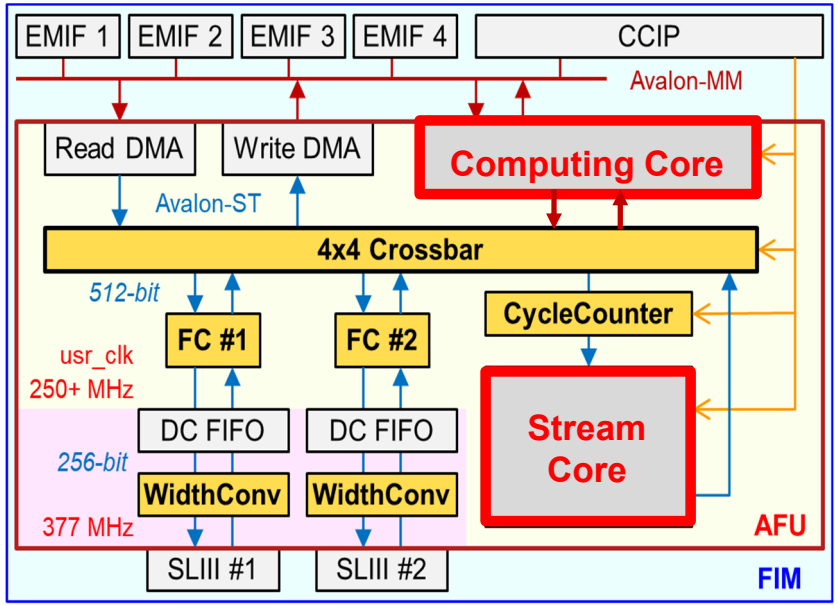
AFU Shell (SoC)

- ✓ Application modules and their interfaces (own development)

Programming Computation

Implement your own core

- ✓ **Computing core** — Connected to DDR4 memories. read and write data by itself.
- ✓ **Stream core** — Connected to crossbar. compute with data stream.



How to program cores

- ✓ **Software-oriented** — **HLS** — **Describe algorithms in C/C++** (Intel HLS)
- ✓ **Hardware oriented** — **HDL** — **Describe hardware structure & FSM** (Verilog-HDL, VHDL, Chisel, etc.)
- ✓ **Others** — **DSL** — **Domain-specific langs for HW generation** (Stream processor generator : SPGen)

Low-level, but **more flexible than OpenCL and its BSP.** Mem IF and network are customizable.

HLS: High-level synthesis, **Chisel**: Scala-based language for RTL, **SPGen** : Stream processor generator

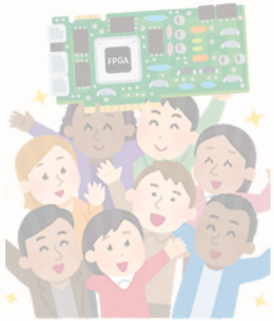
Approaches for Proof of Concept

Req. 1 Interoperability w/ various HPC systems

- ✓ Able to easily extend existing systems with FPGAs
- ✓ Can we extend Supercomputer Fugaku?

Req.2 Flexibility in using FPGAs in a system

to flexibly
stem
chine
e users
ion of FPGAs



Software-bridged FPGA driver

FPGA resource manager

Req.3 Platform with sufficient customizability

✓ Able

inte

FPGA Shell (SoC)

HLS-based programming flow, & FPGA object lib

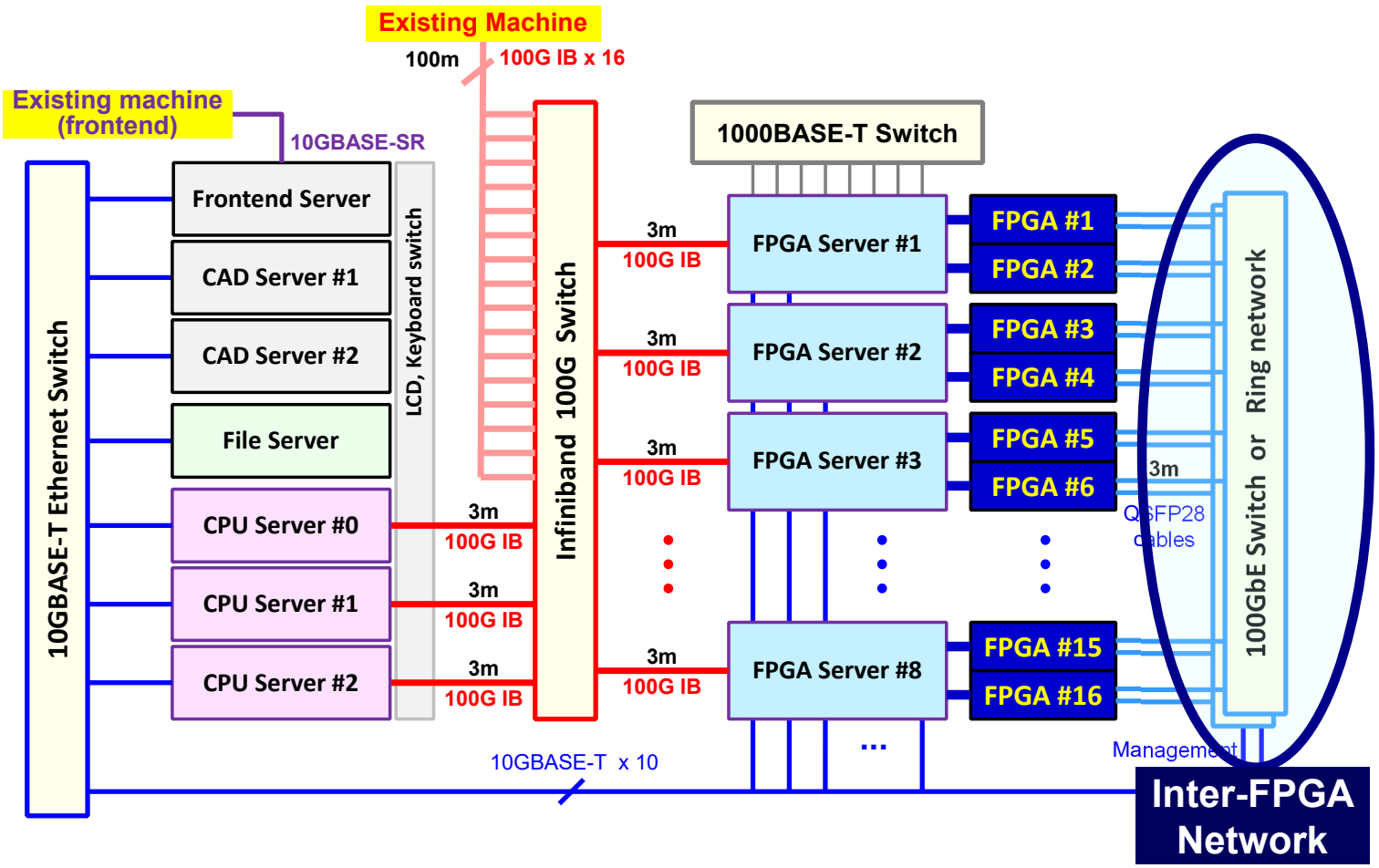
Req.4 Techniques for performance scalability

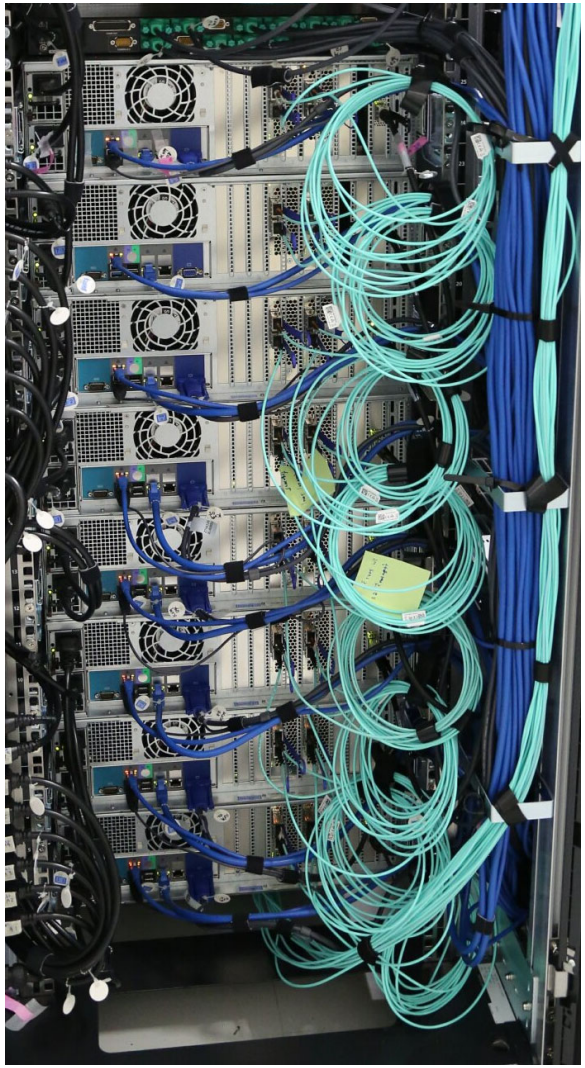
✓ All

Inter-FPGA network

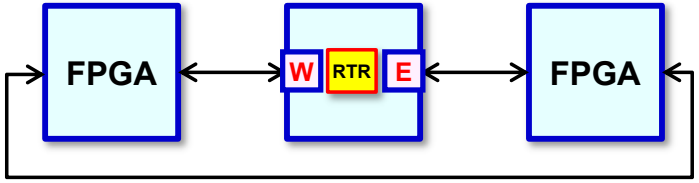
Virtual circuit switching over Ethernet

Inter-FPGA Networks

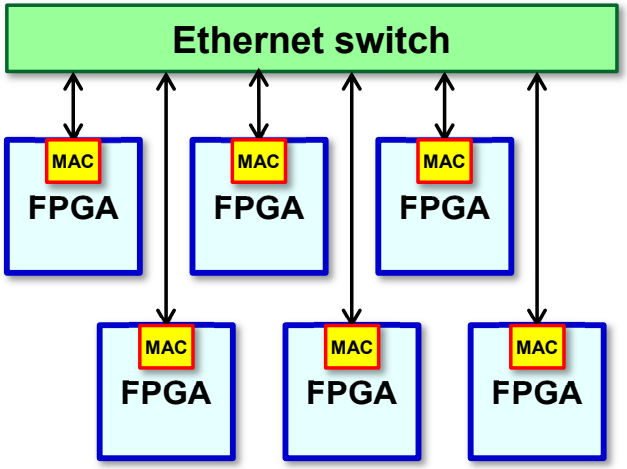




Two Types of Networks



(Arbitrary topology virtualized)



Direct network : 1D torus

Pros) Smaller overhead (lower/fixed latency), easy to use

Cons) Inflexibility of resource allocation, more consumption of HW resources, difficulty to catch up

Indirect network : 100G Ethernet

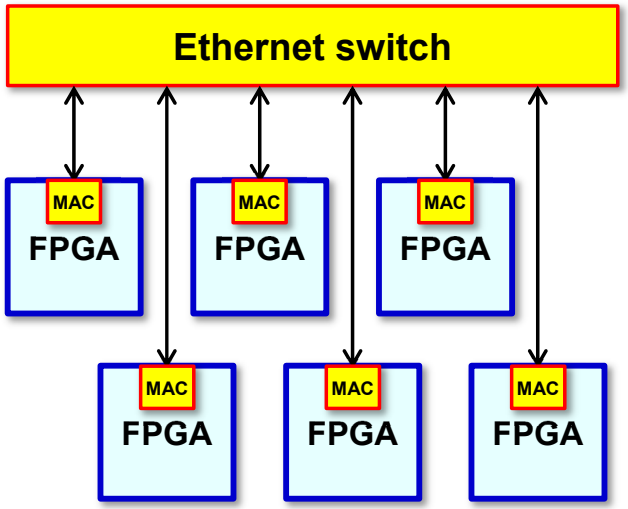
Pros) Flexibility of resource allocation, easy adoption of cutting-edge tech

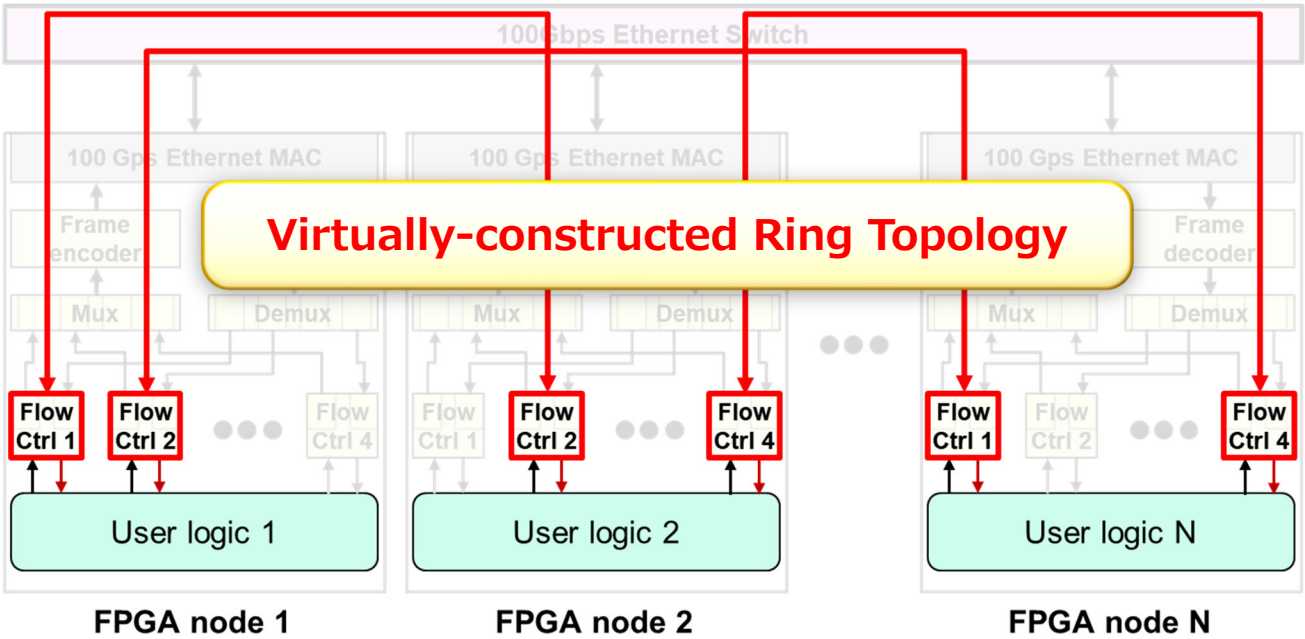
Cons) Overhead of ethernet frames (higher and variable latency), difficulty in flow-control and use, cost of expensive switches

Virtual Circuit Switching Network (VCSN)

Arbitrary topology with virtual links between FPGAs over Ethernet

✓ User logic can simply send and receive data streams through virtual links.





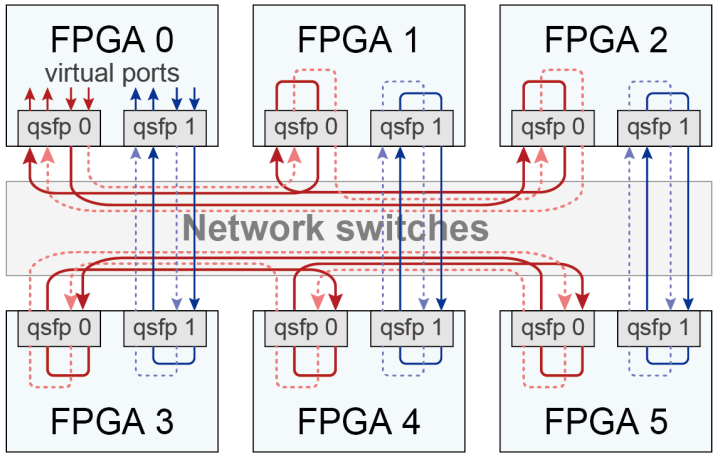
100G Ethernet switches

Pros) Flexibility, cutting-edge technology

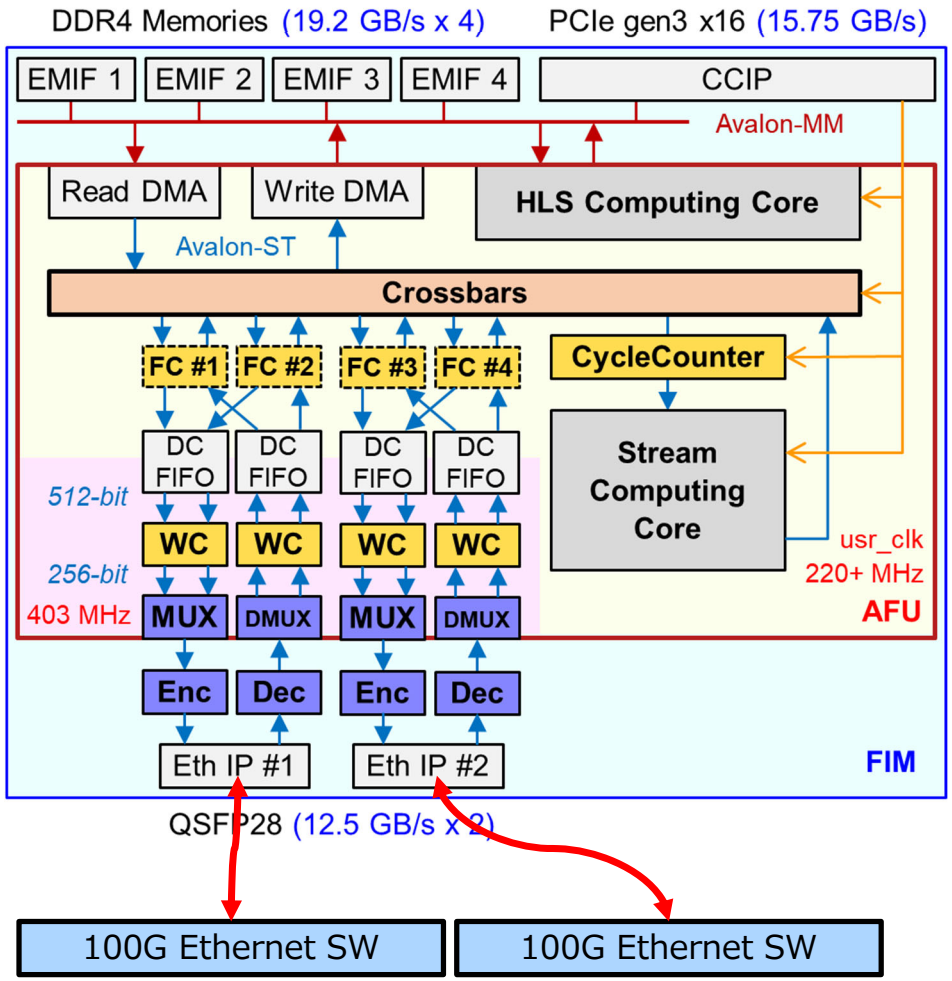
Cons) Overhead of ethernet frames, higher and variable latency, difficulty in flow-control and use

Implementation of Indirect Network

- **Indirect network : 100G Ethernet**
  - ✓ Implementation completed, under verification (2, 4, and 8 virtual ports per Eth MAC)
  - ✓ Higher throughput than Direct network
  - ✓ Developing system software to manage VCSN



b. 2D torus (bi-directional)



Preliminary Evaluation : Throughput

DCN vs.. VCSN

- ✓ **DCN** Direct Connection Network
- ✓ **VCSN** Virtual Circuit Switching Network

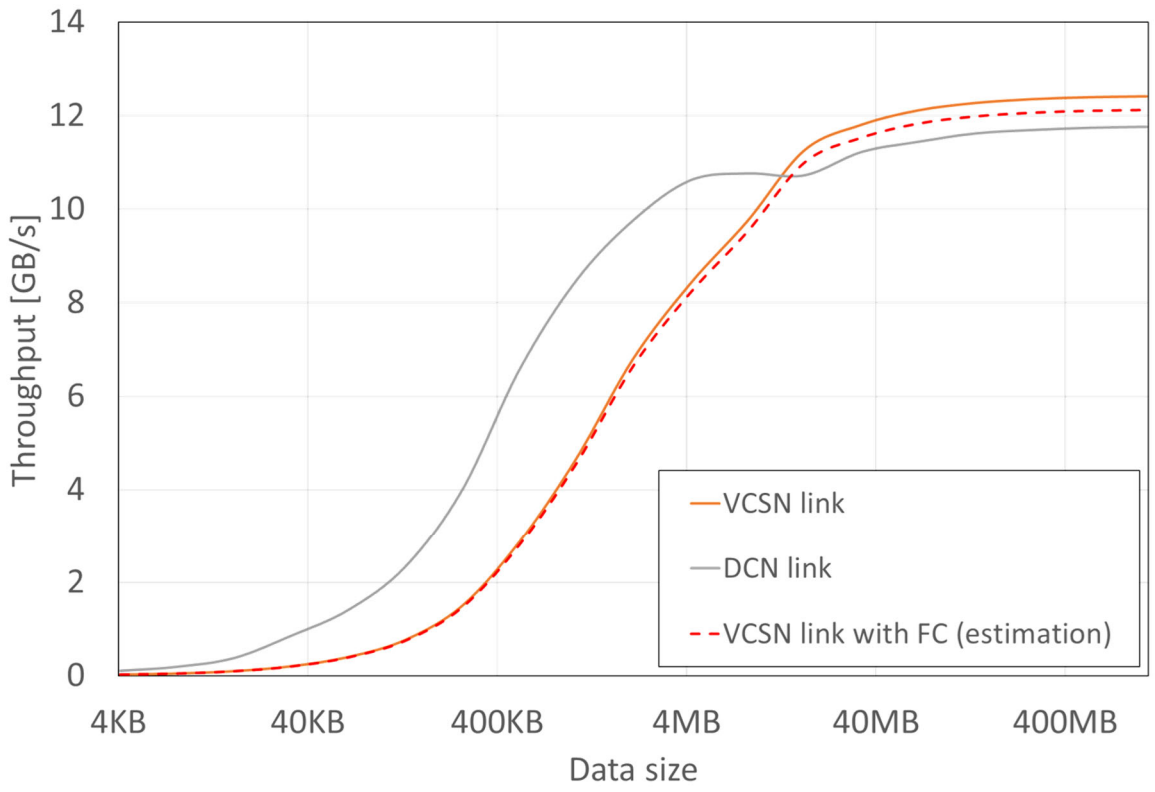
VCSN rises slowly due to higher latency.

- ✓ P2P latency of VCSN 851 ns
- ✓ P2P latency of DCN 490 ns

VCSN has higher Max throughput.

- ✓ Jumbo frame of Ethernet is more efficient.

Anyway, it works!



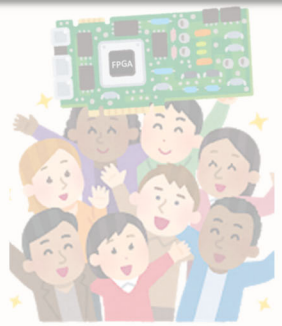
Approaches for Proof of Concept

Req. 1 Interoperability w/ various HPC systems

- ✓ Able to easily extend existing systems with FPGAs
- ✓ Can we extend Supercomputer Fugaku?

Req.2 Flexibility in using FPGAs in a system

to flexibly
stem
chine
e users
ion of FPGAs



Software-bridged FPGA driver

FPGA resource manager

Req.3 Platform with sufficient customizability

✓ Able

inte

FPGA Shell (SoC)

HLS-based programming flow, & FPGA object lib

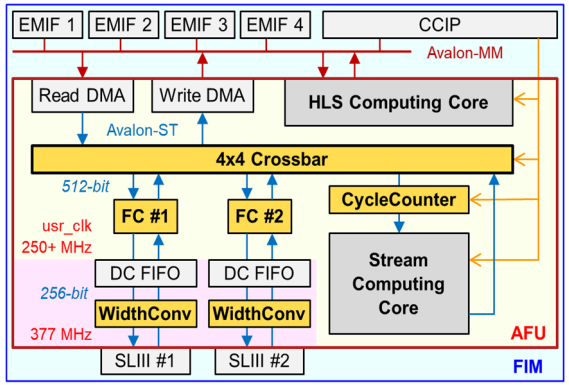
Req.4 Techniques for performance scalability

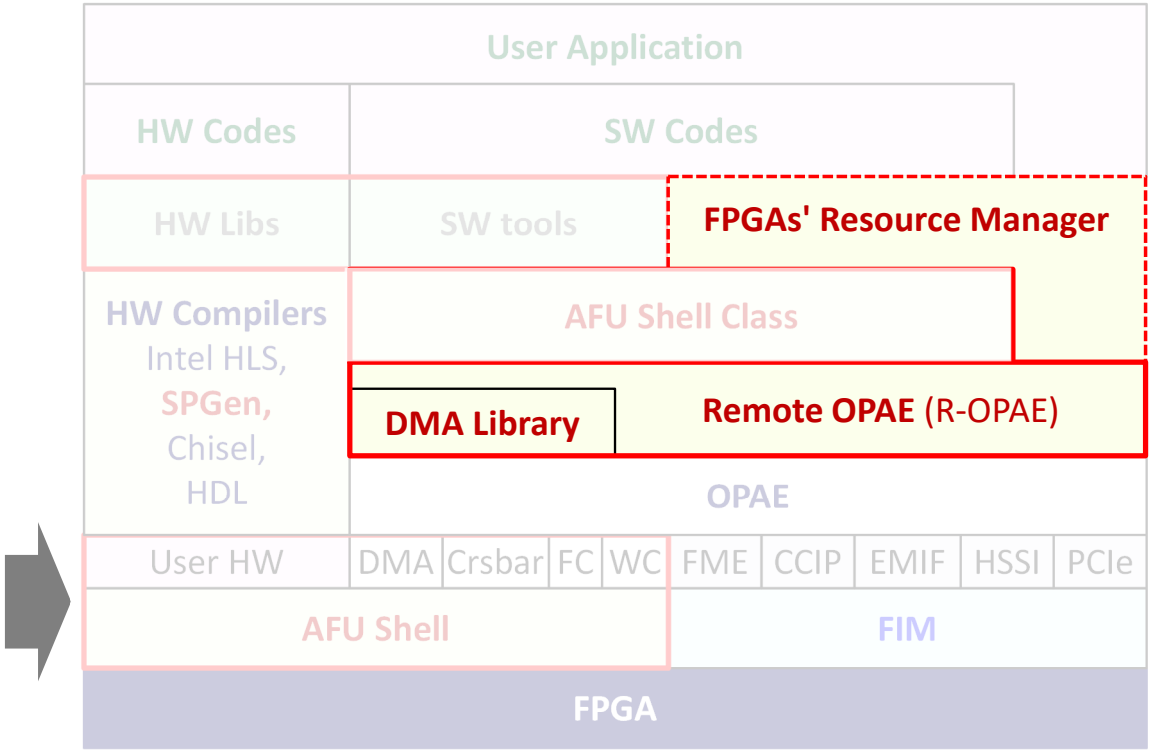
✓ All

Inter-FPGA network

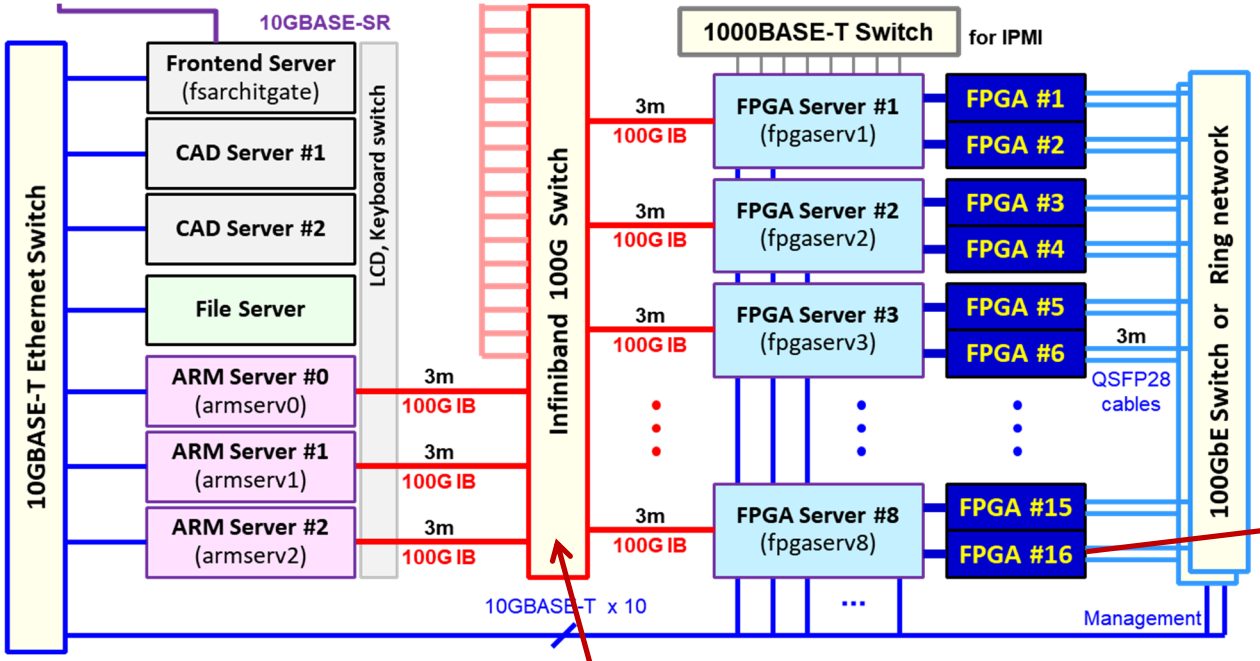
Virtual circuit switching over Ethernet

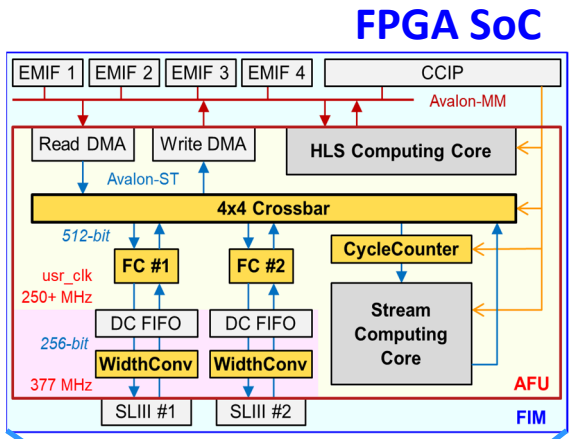
System Stack of ESSPER

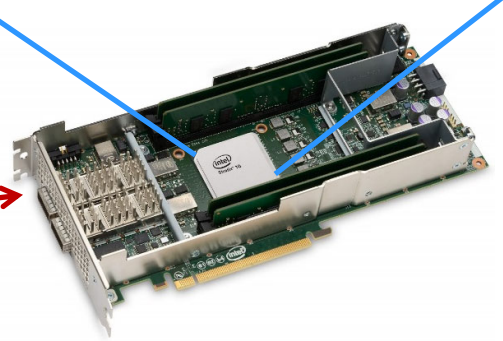




Software-bridged Driver to Utilize Remote FPGAs







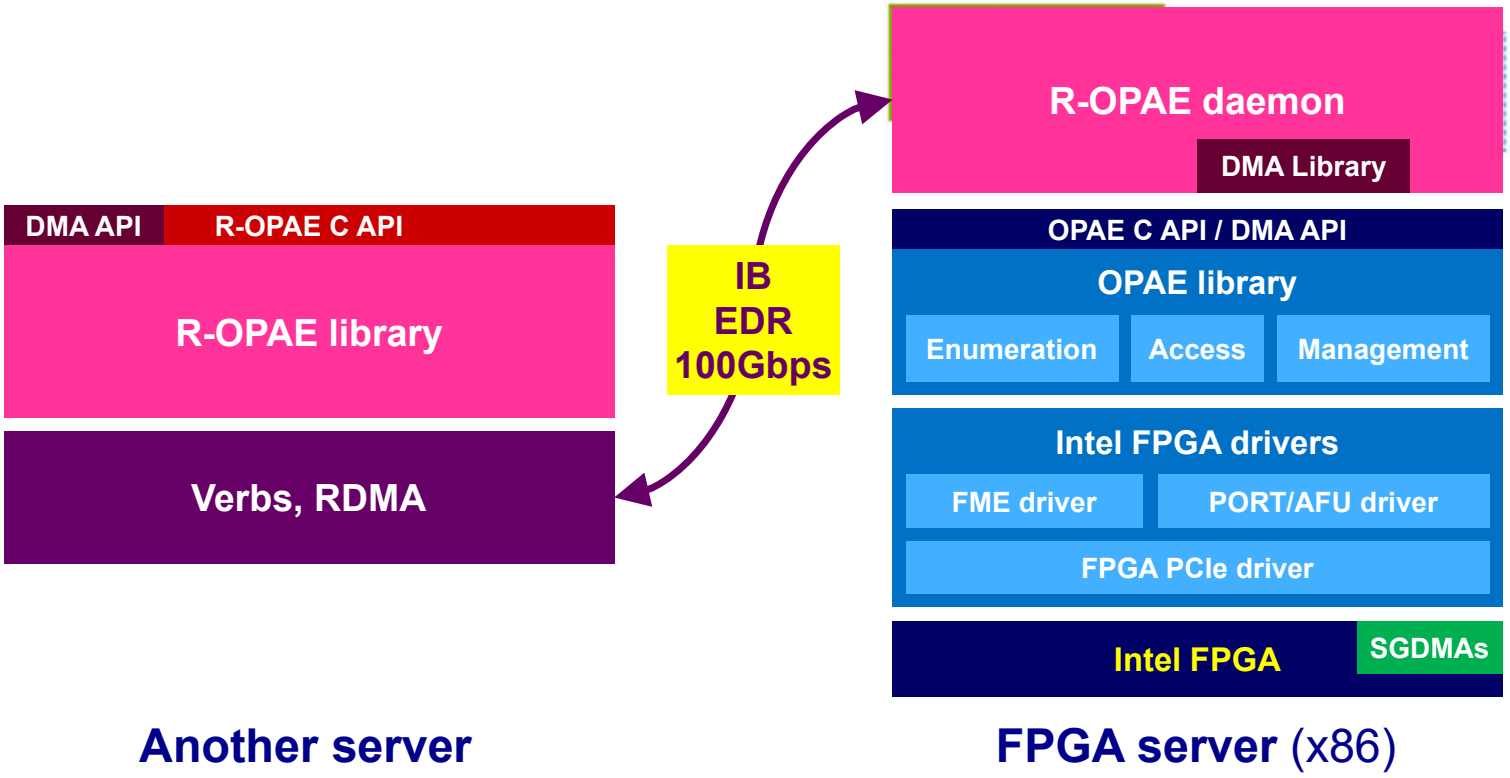
CPU - FPGA network

- 100G Infiniband
- Software-bridged driver (R-OPAE)

Remote-OPAE (for remote FPGA Access)

Software bridge for FPGAs over Infiniband

- ✓ **OPAE**: Open Programmable Acceleration Engine (PCIe FPGA driver)
- ✓ 99% of OPAE APIs are supported.
- ✓ We can use any FPGAs in a system via IB as if they were locally installed.



R-OPAE as Software-based Resource Disaggregation

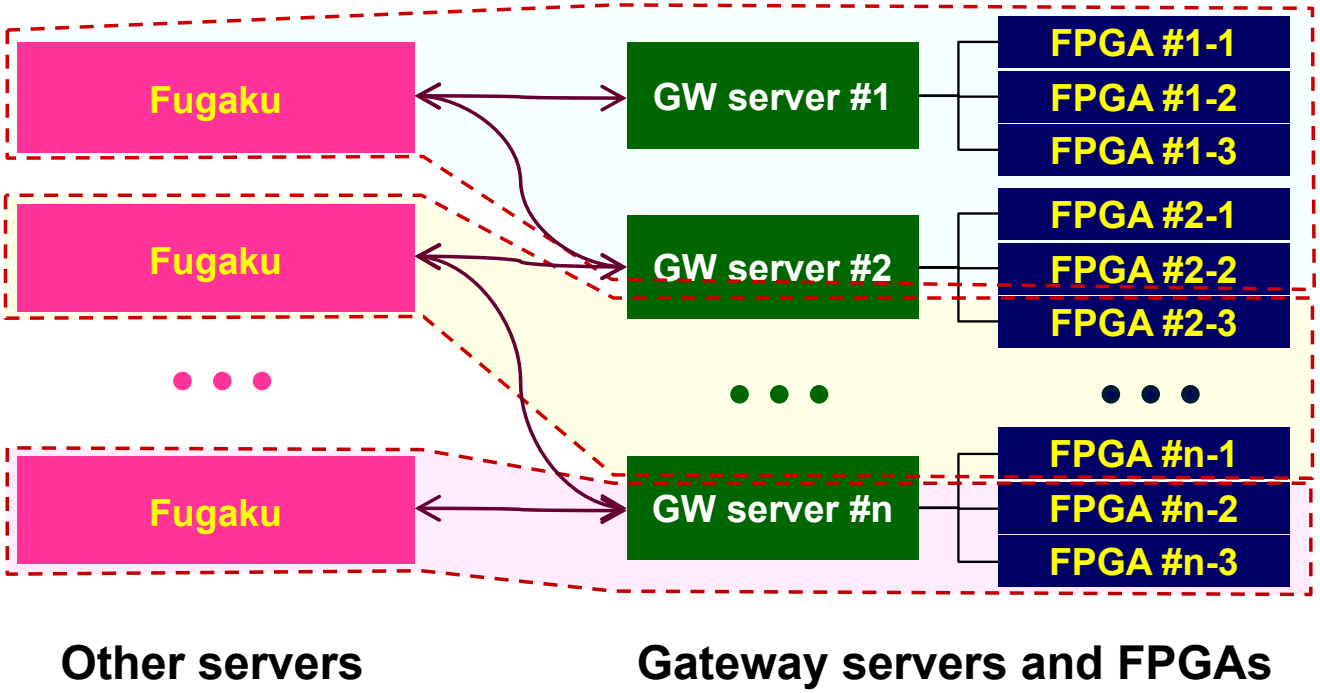
Transparent access to remote FPGAs

Flexible utilization:

- ✓ Can use any available FPGA resources

Inter-operability and extensibility:

- ✓ Vendor/ISA-independent
- ✓ Operable with various architectures such as Fugaku (ARM)



Applications, and Joint Research Projects

On-going (Joint) Research Projects

Hardware

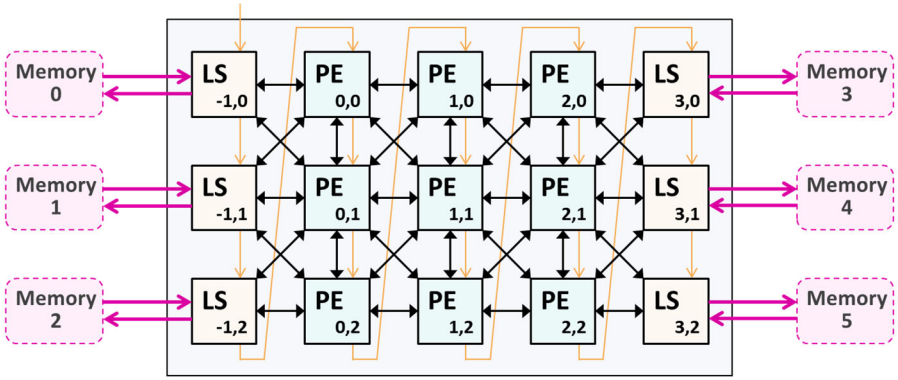
- ✓ Processor Team — **CGRA**
- ✓ Kumamoto Univ — **AI Engine** (ReNA)

System Software

- ✓ RIKEN — RPC for FPGAs
- ✓ Tohoku Univ — neoSYCL (on Fugaku)

Applications

- ✓ Univ of Tokyo — Bayesian network analysis
- ✓ Meiji Univ — 3D FFT (presented later)
- ✓ Processor Team — Fluid simulation
- ✓ Nagasaki Univ — Convex method
- ✓ Hiroshima City U — Breadth First Search of Graph
- ✓ Processor Team — Hardwired MNIST
- ✓ JAIST — Sound rendering



Riken CGRA (coarse-grained reconfigurable array)

AI Engine, ReNA

Future Prospects for (Reconfigurable) HPC

Reconfigurable and Data-Flow Architectures are Promising (CGRA).

Backward compatibility is also required.

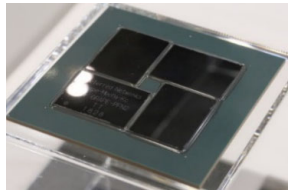
- ✓ Programming language & eco-system

System design supported by

- ✓ Archi. coupling CPU & accelerators for various types of workloads
- ✓ Compiler and system software for seamless usage

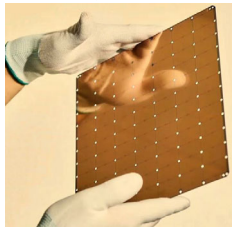
XILINX VERSAL

Graphcore



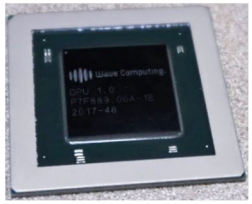
PFN MN-Core

Groq



Cerebras

SambaNova



Wave computing

DOI:10.1145/3282307

Innovations like domain-specific hardware, enhanced security, open instruction sets, and agile chip development will lead the way.

BY JOHN L. HENNESSY AND DAVID A. PATTERSON

A New Golden Age for Computer Architecture

WE BEGAN OUR Turing Lecture June 4, 2018[11] with a review of computer architecture since the 1960s. In addition to that review, here, we highlight current challenges and identify future opportunities, projecting another golden age for the field of computer architecture in the next decade, much like the 1980s when we did the research that led to our award, delivering gains in cost, energy, and security, as well as performance.

"Those who cannot remember the past are condemned to repeat it." —George Santayana, 1905

Software talks to hardware through a vocabulary called an instruction set architecture (ISA). By the early 1960s, IBM had four incompatible lines of computers, each with its own ISA, software stack, I/O system, and market niche—targeting small business, large business, scientific, and real time, respectively. IBM engineers, including ACM A.M. Turing Award laureate Fred Brooks, Jr., thought they could create a single ISA that would efficiently unify all four of these ISA bases.

They needed a technical solution for how computers as inexpensive as

» key insights

- Software advances can inspire architecture innovation.
- Elevating the hardware/software interface creates opportunities for architecture innovation.
- The marketplace ultimately settles architecture debates.

John L. Hennessy, David A. Patterson, Communications of the ACM, Feb 2019.

Summary

Hiring researchers:
R-CCS2105 or R-CCS2022

Goal	Explore new system architectures for reconfigurable HPC
This project	**ESSPER**: Elastic and Scalable System for High-Performance Reconfigurable Computing
PoC :	***Interoperability and flexibility, Platform for customizability, and scalability***

Research subjects

- ✓ More applications with multi-FPGAs, Shell with HBM2, Resource management & Task scheduling for Fugaku

Expecting Collaborations with You!