EUROEXA
FUNDED BY THE EUROPEAN UNION

The Euroexa system architecture for exascale

Prof. John Goodacre
The University of Manchester
30 June 2020

Project ID: 754337

MANCHESTER 1824
The University of Manchester

EURO EXA

Key Challenges to Investigate

To control the ALU costs over 70% of the required power

- Need to flow data between ops rather than control which op on data
- Need to stop storing data between ops to reduce the bandwidth disparity between compute/dram

Movement of data cost orders of magnitude more power than the ALU itself

- Need introduce data locality to reduce the movement of data
- Need to increase compute-density to further reduce distances

Scaling the sequential execution of ALU is limited by latency

- Need to pipeline ops while ensuring low latency for remaining sequential parts

The speed of the network is approaching the speed of memory

- The network needs to nativity attach to compute, just as memory does today

Storage devices are castrated by the way they are accessed

- Need to directly attach to the network while providing improved local access

Memory capacity within a node won't scale

- Need to extend the current "shared memory" and "device" paradigm with "remote memory" paradigm

Application specific chips are economically unviable for most markets

- Need to modularise device manufacturing using scalable units compute, use of FPGA prior to ASICs

Without breaking existing software

While improving on existing performance metric

and utilizing the best of class compute/memory and acceleration technologies

Driving High-level Concepts

Hardware Story

- Accelerator centric architecture
  - Accelerator own application RAM
  - Hostless link to IO (storage/interconnect)
- Compress distances,
  - Leverage locality
- High-thermal/compute densities
- Hybrid interconnect topologies
- Converged distribute storage
  - Local access to shared data

Software Story

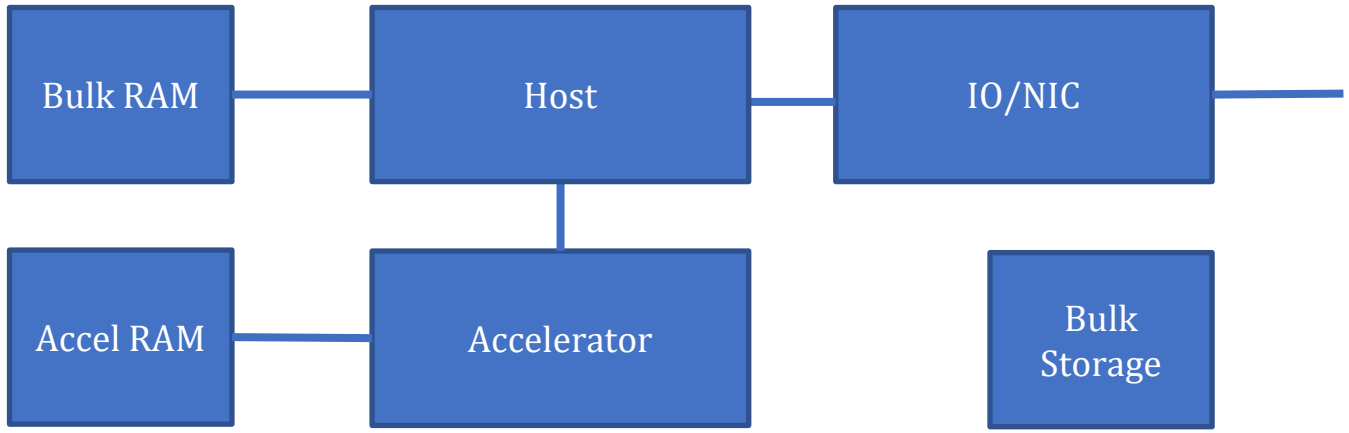
- Mix-granular Dataflow
  - Use of FPGA to investigate non von-Neumann application acceleration at fine grain
  - Data-aware task scheduling at course grain
- Using Compute Unit scalability model
  - Direct network link-layer memory transaction (Unimem model)
  - Global shared memory
    - Asymmetric cache coherence

EUROEXA

FUNDED BY THE EUROPEAN UNION

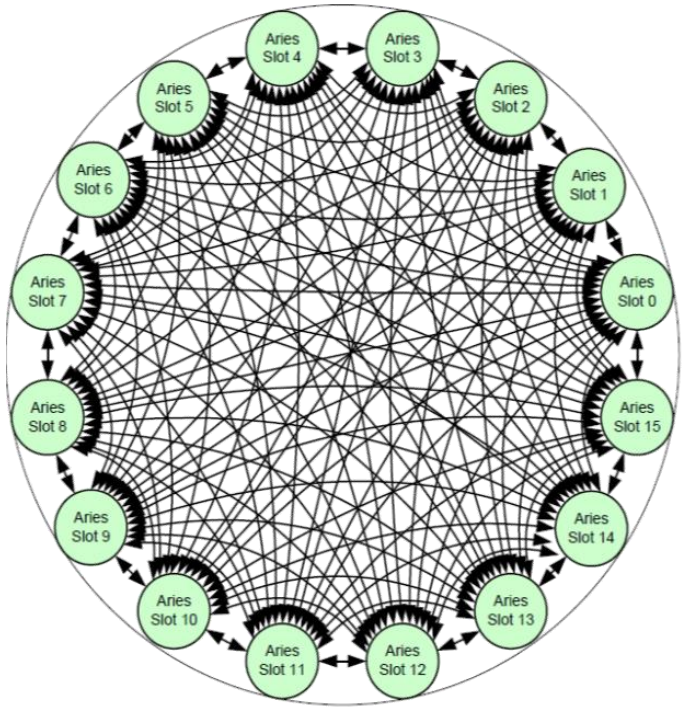
Hardware Story

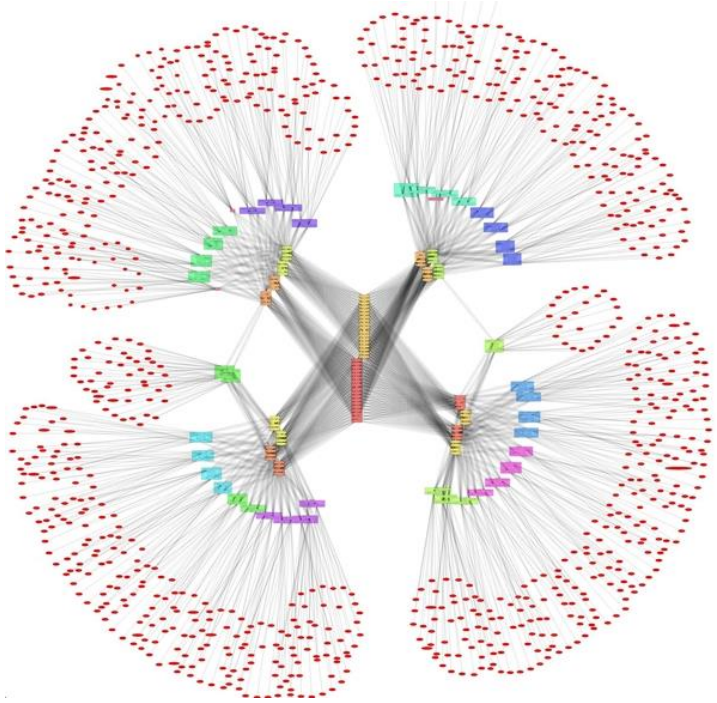
Typical Node Architecture



- Host owns the IO and the Accelerator
  - Struggling to bypass bottlenecks of the Host
- Storage is remote to node
- Host manages all application RAM

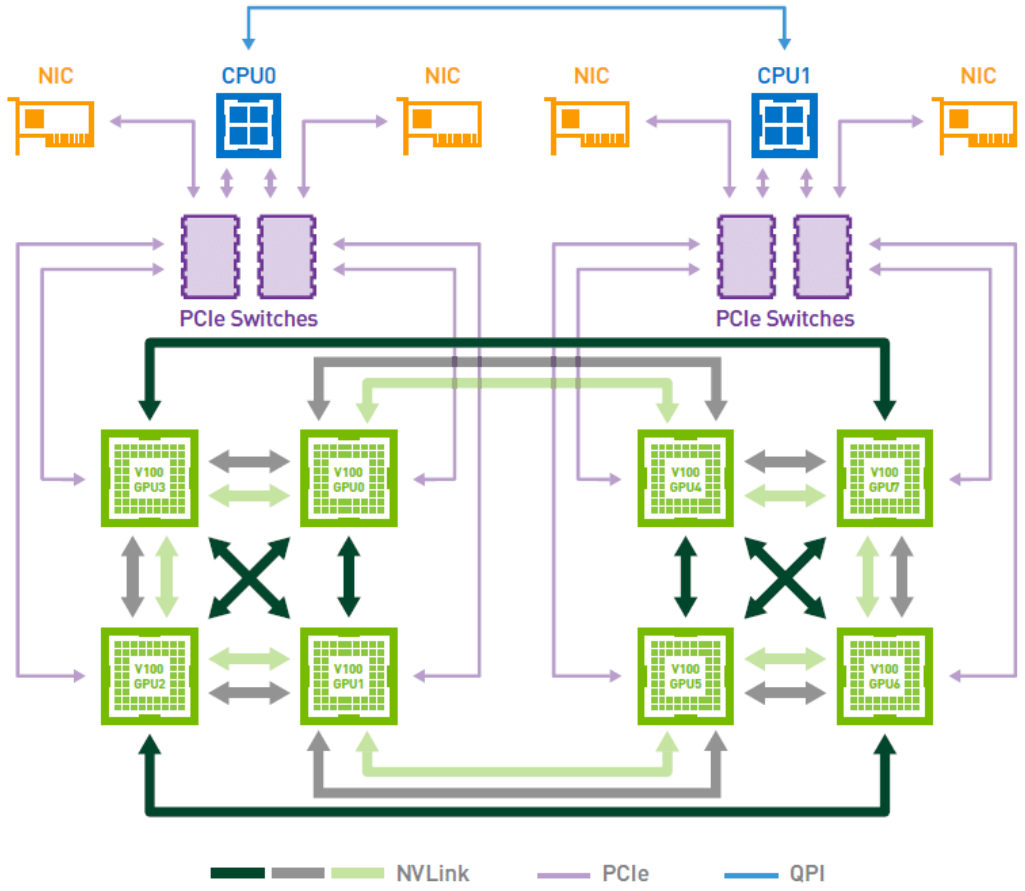
Current main interconnects





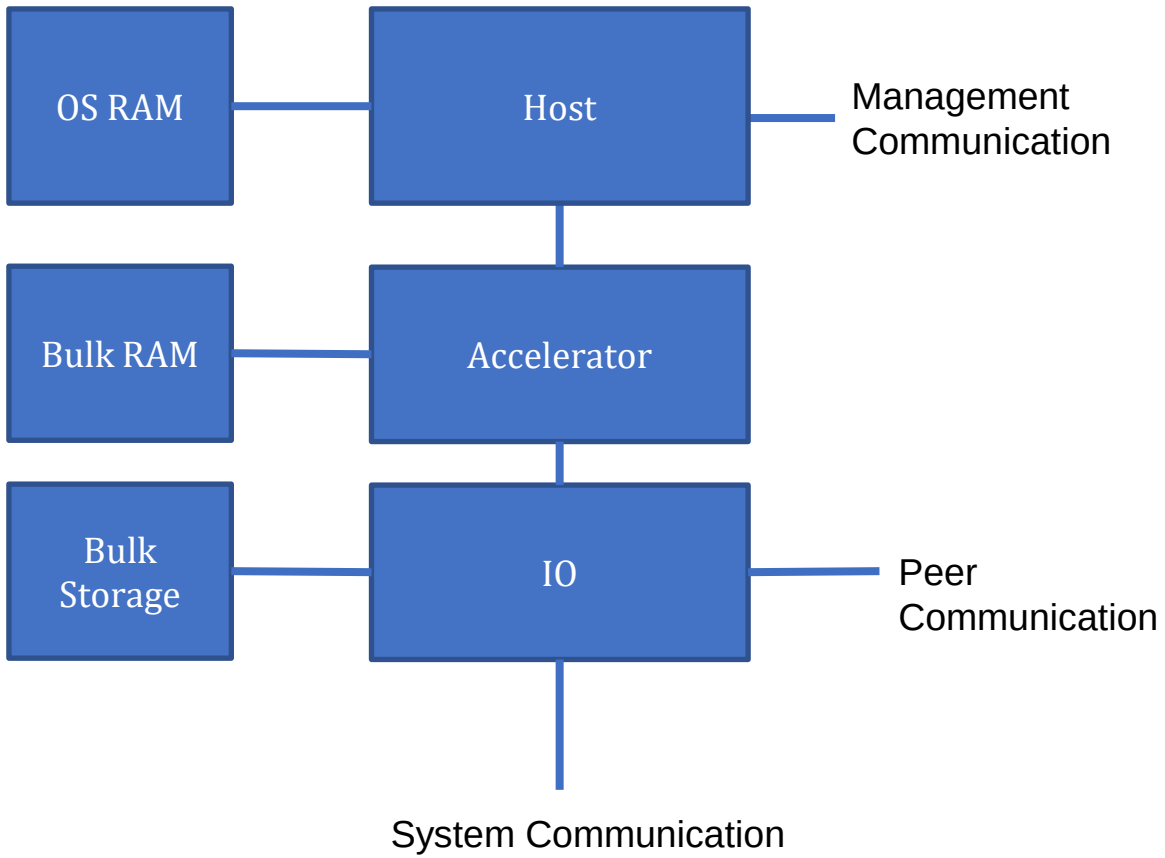
- All-to-all topologies
  - Some node aggregation happening due to limits of radix vs hop/node count
- Tree structures
  - Increasing bandwidths towards top to reduce radix limits on upper level switches

Example Newer Architecture



- Accelerators are getting independent interconnects with peer to peer communication paths
  - NIC can bypass CPU for direct IO links using PCI
- Hosts still need a main interconnect
  - Storage is remote to node
  - Node communication needed between groups of accelerators

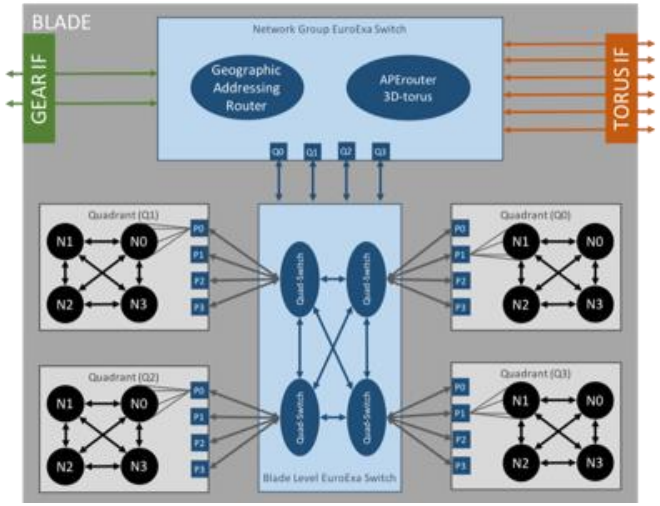
Euroexa Node Architecture

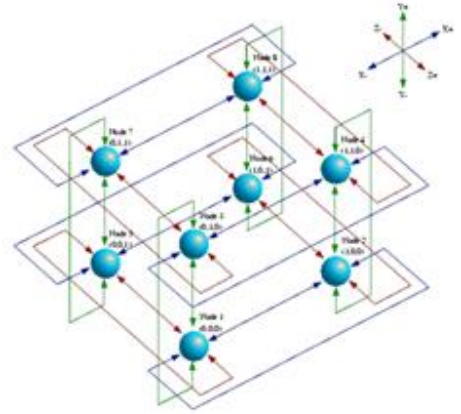


- Interconnect built using IO “NIC”
- IO provides:
  - Low latency peer links
  - System level communications
  - Creates distributed storage out of directly attached storage
- Accelerator has direct path to IO
  - Owns the bulk RAM (globally shared)
- Host manages application on accelerator
  - Can also use the IO interconnect

Interconnect architecture

Single top of rack switch per cabinet



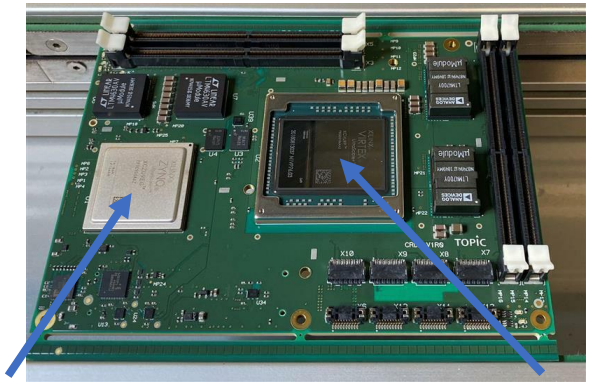


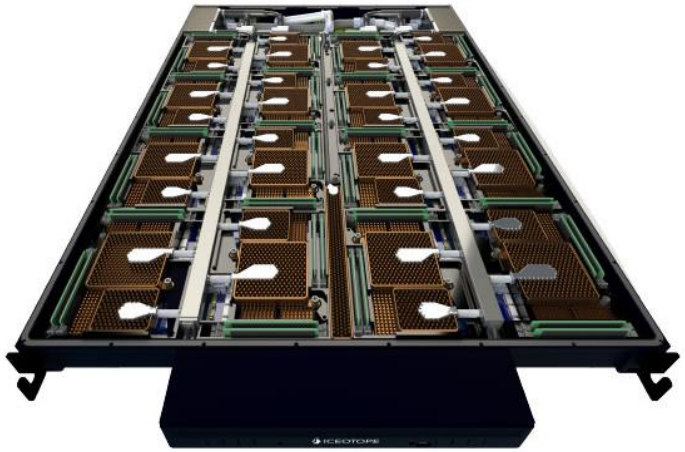
Creates a Network Group across 8 blades

- Minimizing communication distances
- Reducing pressure on radix needed at each level

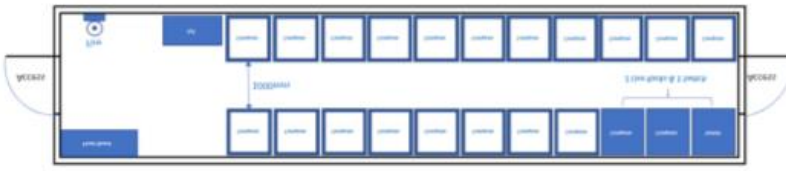
- Four nodes grouped into fully connected quadrants with four quadrants grouped into fully connected Blade
- Blade level NIC create a single hop 3D-torus across a Network Group of 8 blades
- Blade also bridges out to high bandwidth hierarchy of switches. Evaluation using OpenFlow configured switches

Mechanical




Blade (8x100Gb uplink)

- The EuroExa Blade
- Modular Compute
- High Thermal Density
- Hybrid Interconnect
- ~100,000 op per cycle
- Multiple Tb/s interconnect
- 100’s TB local storage
- Exascale within 35m x 35m

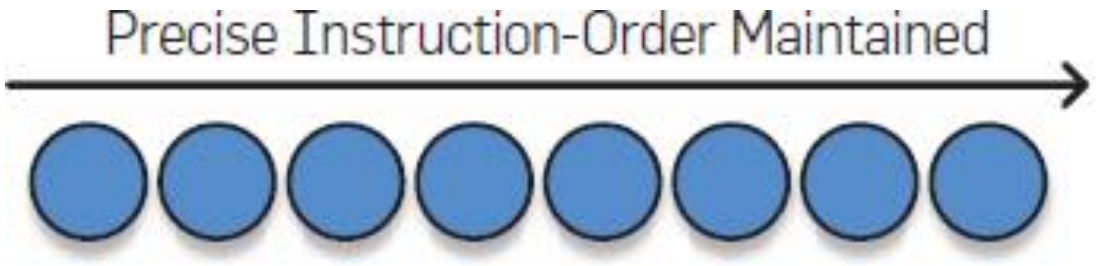

Container

EUROEXA

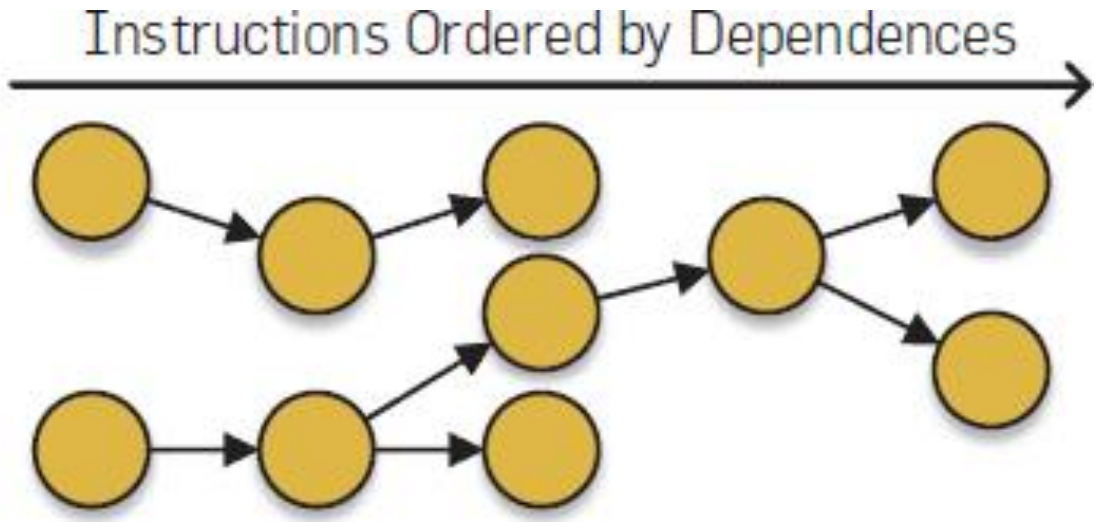
FUNDED BY THE EUROPEAN UNION

Software Story

Execution Models

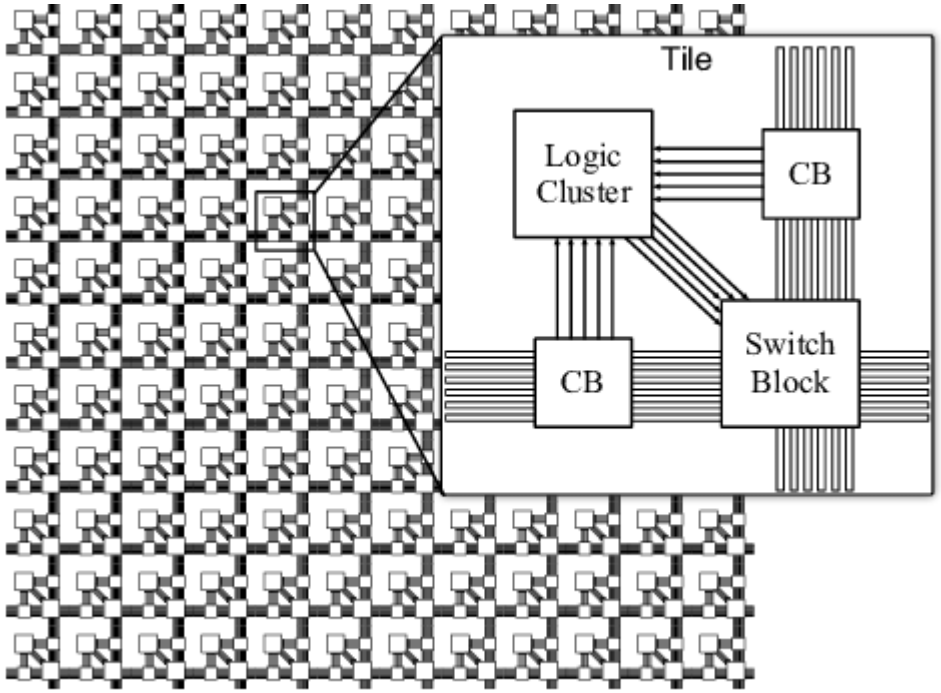


Von Neumann Execution Model



Dataflow Execution Model

Why Data flow? Why FPGA?



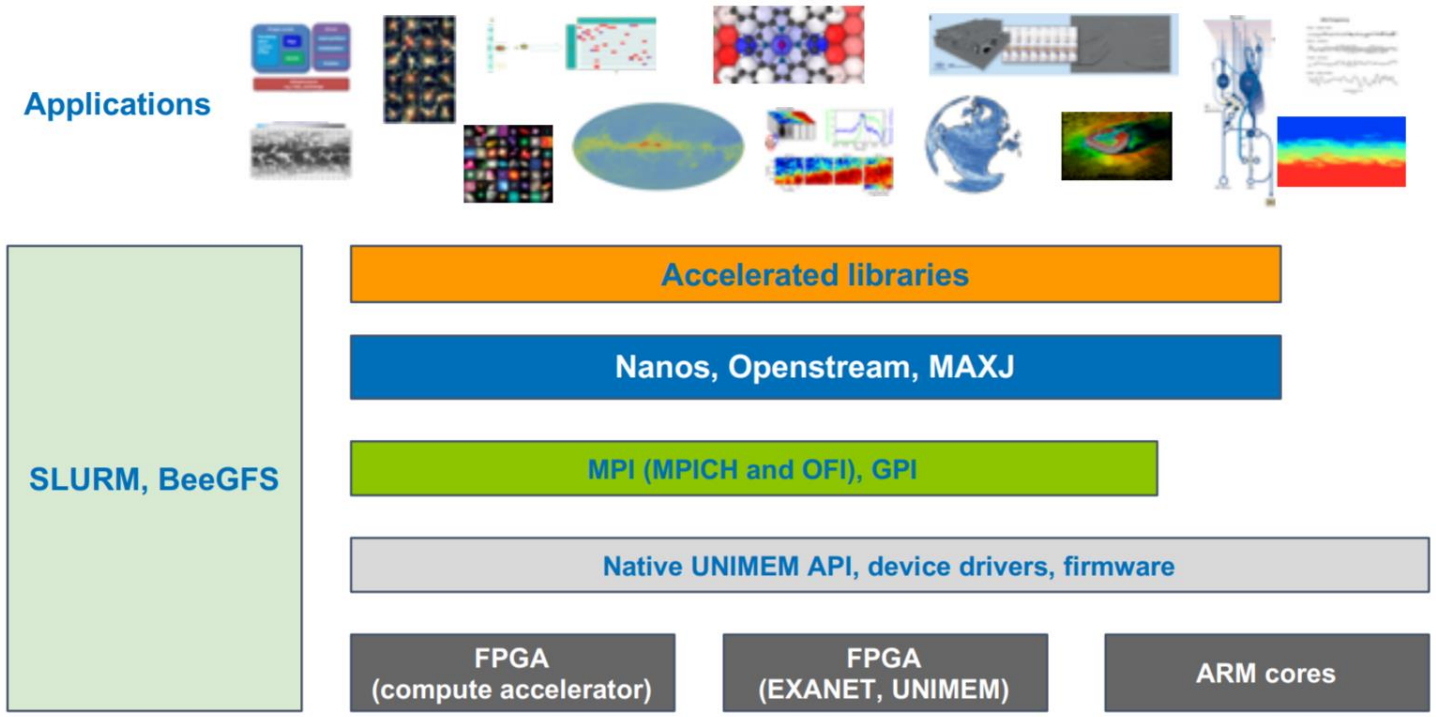
Euroexa is not trying to define a data-flow processor

- Unlike a von-Neumann processor, a FPGA is just a lot of “MACs” and wires
  - 1,000’s of MACs per cycle vs. 10’s in a CPU
- A program is not defined as a sequence of operations on data, but is defined as how data flows through operations
  - No need to store intermediate values to RAM
- Since the dataflow of an application does not need a control unit, the power efficiency can be significantly higher than a CPU

Application stack

Applications will target portable programming models / communication libraries

- MPI, GPI, OpenStream, OmpSs, MaxJava and PSyClone



More details see: https://web.fe.up.pt/~specs/events/wrc2019/presentations/Paul%20Carpenter-WRC2019.pdf

Conclusions

- Euroexa has taken a holistic approach to the power/performance challenge for Exascale
- Outcomes are indicating that applications need to consider a mixed-granular data-flow approach with increased focus on locality
- Prototype hardware and infrastructure indicating positive results

EUROEXA

FUNDED BY THE EUROPEAN UNION

Thankyou

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 754337

MANCHESTER 1824 The University of Manchester

EURO EXA