The Evolution and Revolution required for exascale

John Goodacre

Professor of Computer Architectures

Advanced Processor Technologies Group

University of Manchester

MANCHESTER 1824

The University of Manchester

Disclaimer

- This presentation summarises my personal views on various topic of which I claim no association or representation of other individuals, associations or affiliations

Disclosure

- I hold the following positions of employment

  Professor Computer Architectures, University of Manchester.

  Director of Technology and Systems, ARM Ltd.

  Co-founder and Chief Scientific Officer, KALEAO Ltd.

Targeting ExaScale: Technological Challenge

Advanced Scientific Computing Research (ASCR)

- ASCR Home
- About
- Research
  - Applied Mathematics
  - Computer Science
  - Next Generation Networking
  - Scientific Discovery through Advanced Computing (SciDAC)
    - Co-Design
    - SciDAC Institutes
  - ASCR SBIR-STTR
- Facilities
- Science Highlights
- Benefits of ASCR
- Funding Opportunities
- Advanced Scientific Computing Advisory Committee (ASCAC)
- Community Resources

Scientific Discovery through Advanced Computing (SciDAC)

Exascale Challenges

Print | Text Size: A A A

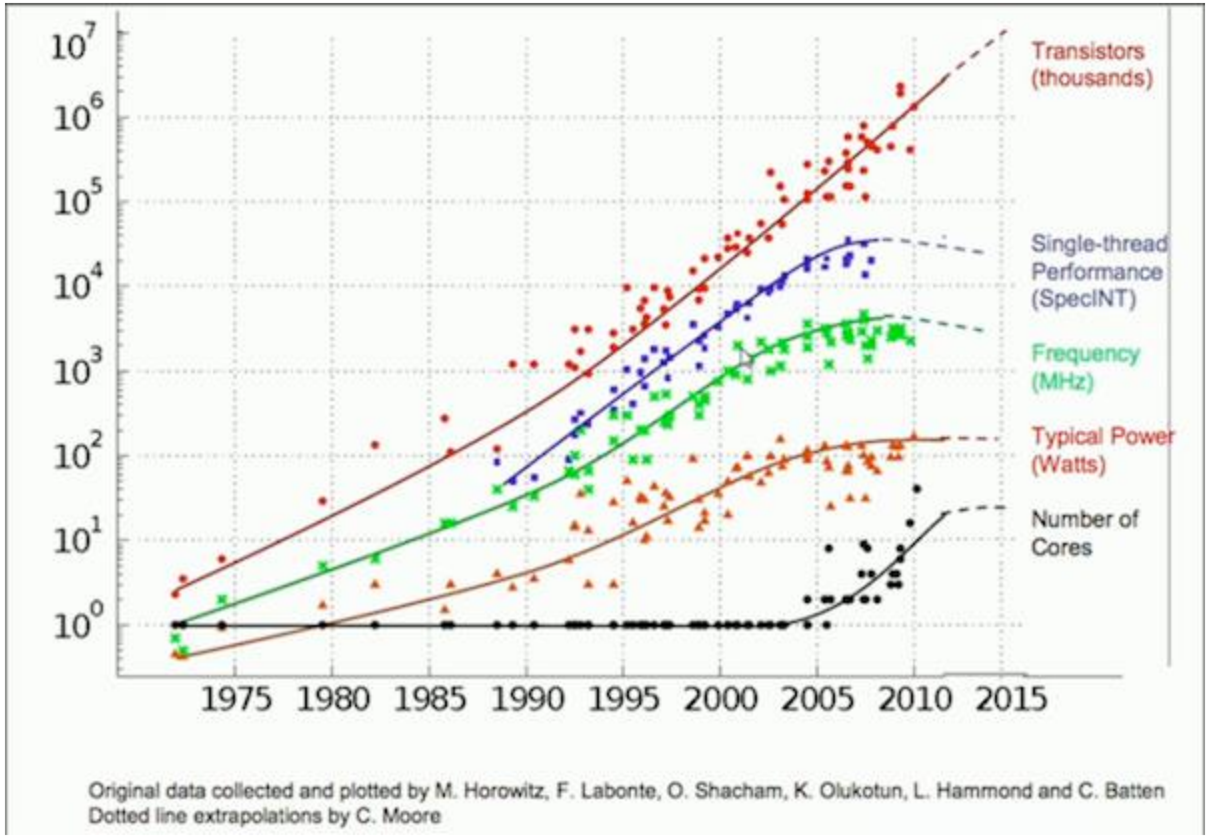
The Challenges of Exascale

The emerging exascale computing architecture will not be simply 1000 x today's petascale architecture. All proposed exascale computer systems designs will share some of the following challenges:

- Processor architecture is still unknown.
- System power is the primary constraint for the exascale system: simply scaling up from today's requirements for a petaflop computer, the exaflop computer in 2020 would require 200 MW, which is untenable. The target is 20-40 MW in 2020 for 1 exaflop.
- Memory bandwidth and capacity are not keeping pace with the increase in flops: technology trends against a constant or increasing memory per core. Although the memory per flop may be acceptable to applications, memory per processor will fall dramatically, thus rendering some of the current scaling approaches useless
- Clock frequencies are expected to decrease to conserve power; as a result, the number of processing units on a single chip will have to increase – this means the exascale architecture will likely be high-concurrency – billion-way concurrency is expected.
- Cost of data movement, both in energy consumed and in performance, is not expected to improve as much as that of floating point operations , thus algorithms need to minimize data movement, not flops
- Programming model will be necessary: heroic compilers will not be able to hide the level of concurrency from applications
- The I/O system at all levels – chip to memory, memory to I/O node, I/O node to disk—will be much harder to manage, as I/O bandwidth is unlikely to keep pace with machine speed
- Reliability and resiliency will be critical at the scale of billion-way concurrency: "silent errors," caused by the failure of components and manufacturing variability, will more drastically affect the results of computations on exascale computers than today's petascale computers

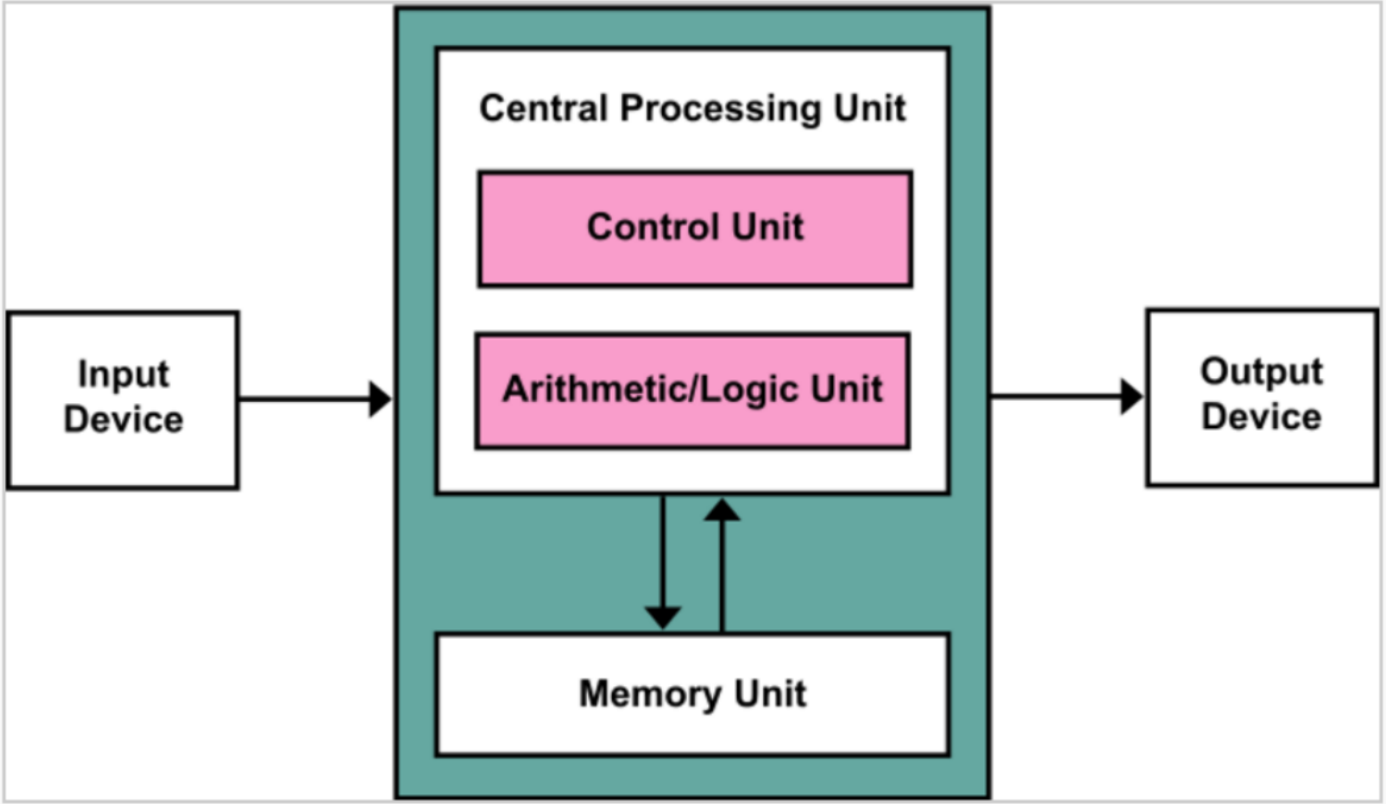
- The Challenge Summary
  - Deliver lots of FLOPS
  - In very little power
  - By 2020
- ...the unspoken challenge
  - It it even feasible using existing paradigms ?
  - Other than a couple of governments, who can afford to build one ?
  - How will software use it ?
  - ..Is HPL the way to measure it ?

Many-core the solution ?



- Since 2005, CPU "complexity" reached a plateau
  - No more GHz
  - No more issue width
  - No more power available
  - No more space to add "pins"
- But still get more transistors
  - Current efforts to increase number processor
  - …but

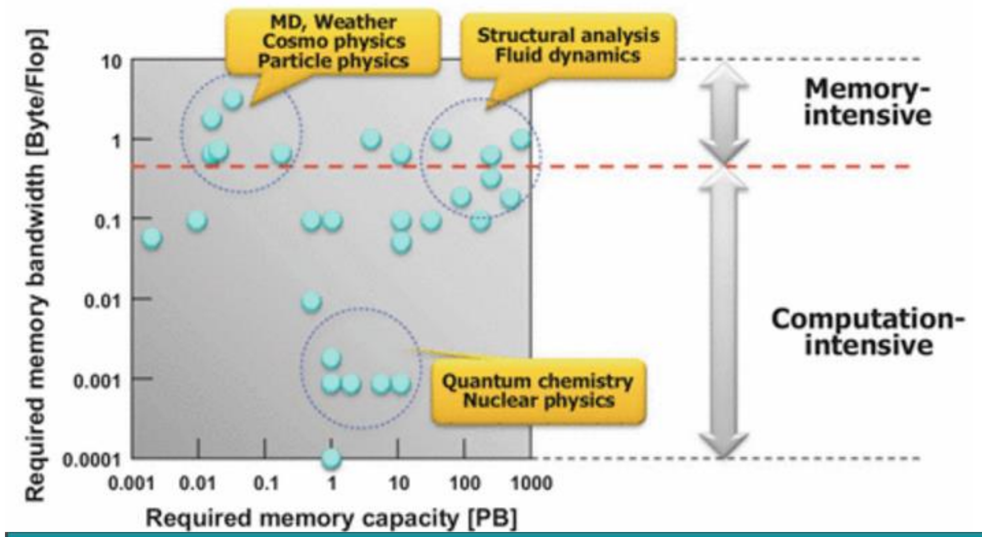
Limitations of von Neumann model

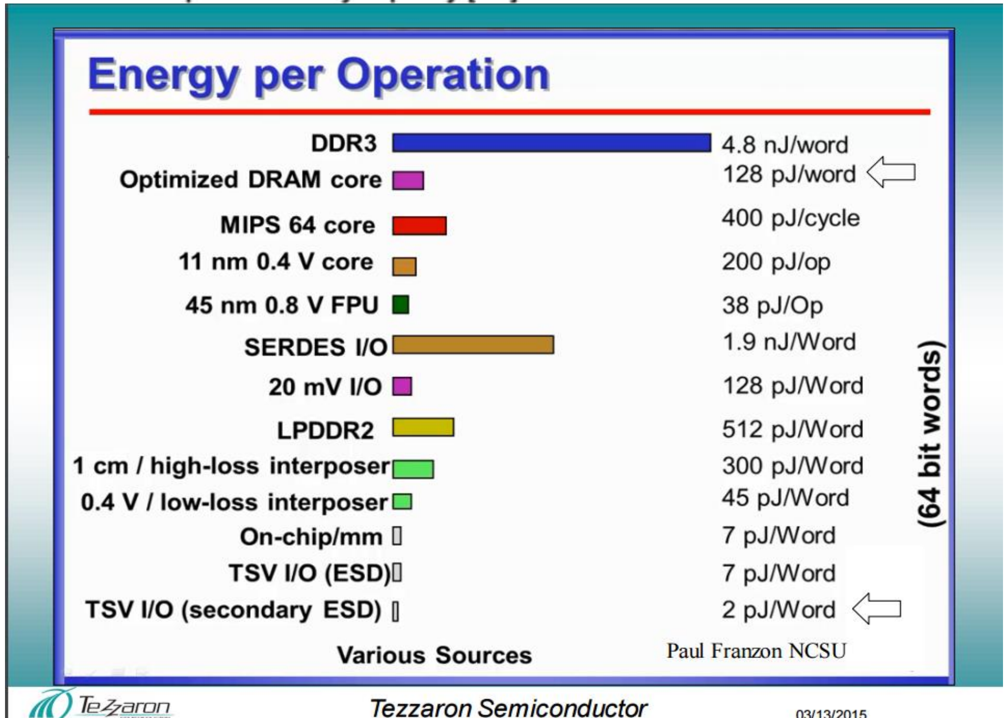


Von Neumann architecture scheme.

- Fundamental model of most of today's systems
- Suffering the memory bottleneck
- Energy ratio between control and arithmetic / IO
- Scalability throughI/O communication
    - Except numa which scales the CPU, a little

How bad is the memory bottleneck ?



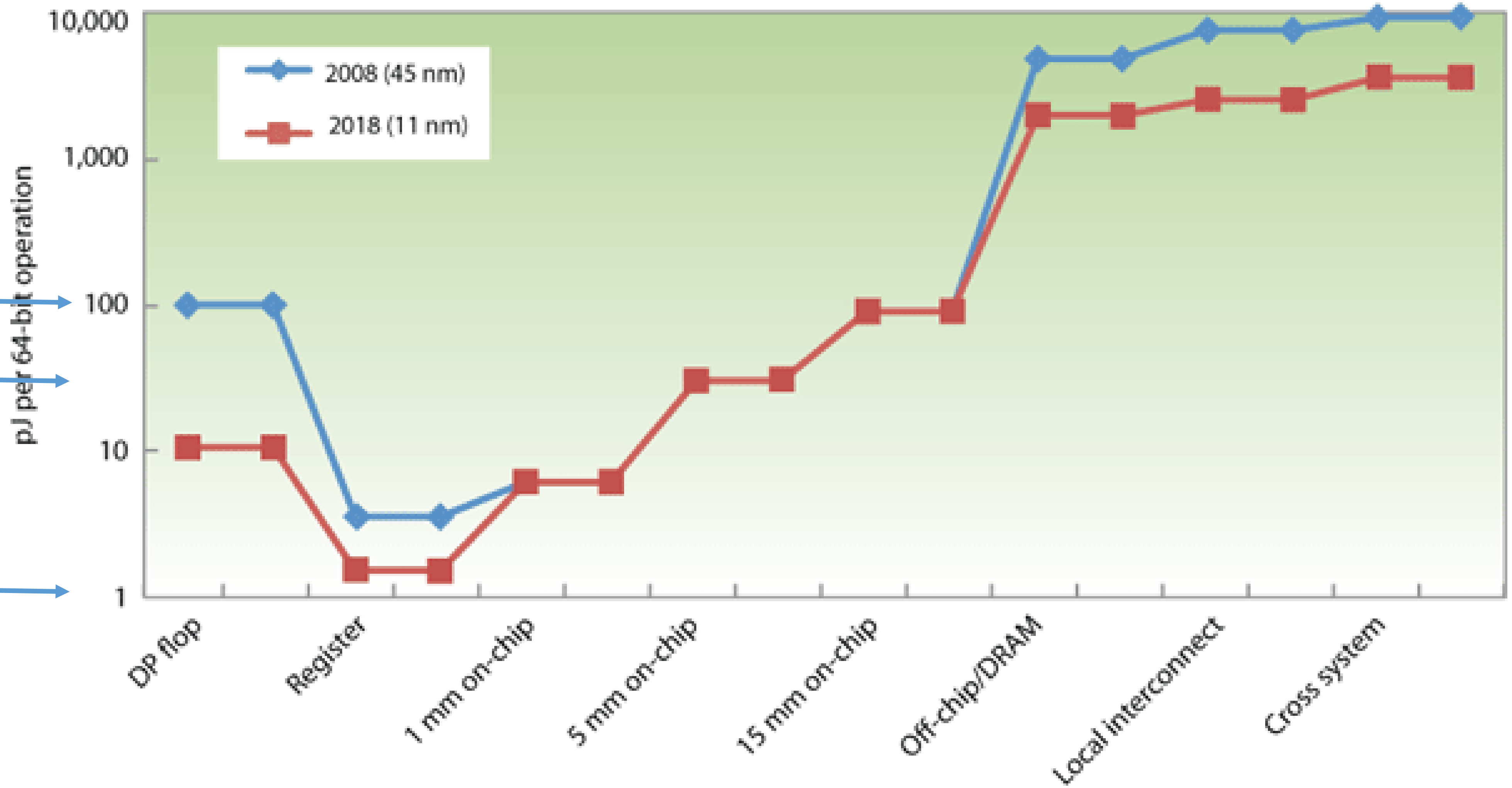


- If designs needs to assume around 1 per FLOPS per byte accessed
  - 500GFLOP processor needs to keep it fed with 500GB/s of main random access memory
- Today’s best DDR is ~100pJ/word
  - So 50pJ/byte, or 50M Watts at 1 flop/byte
  - So, exascale target – BUSTED!
- A few GB of capacity can be placed on chip, to bring this to 5M Watt – excluding any static energy of the memory,
- Will SCM (eg 3DXPT) solve this?

Energy of data movement operations

Energy from
Doing the OP

Energy from
Working out
What OP to do



Ways to increase processing efficiency

Increase the number of arithmetic operations over the amount of control needed

- Incrementally increase control cost to operate on multiple data items
  - Eg. SIMD or vector machines

- Find a more complex compiler to execute multiple operations in a single instruction
  - Eg VLIW, DSP

- Increase number of control units by reduce their complexity, and operate on multiple data items
  - Eg. GPGPU

- "remove" control, and create a fixed sequence of operations
  - Hardware accelerators

- Consider reconfigurable hardware which enables programmability to execute multiple operations in a single cycle over multiple data items
  - Eg FPGA

Ideally without needing to store intermediate values into a memory (hierarchy)

ARMv8-A Next-Generation Vector Architecture for HPC

ARM

Nigel Stephens

Lead ISA Architect and ARM Fellow

Hot Chips 28, Cupertino

August 22, 2016

Expanding ARMv8 vector processing

- ARMv7 Advanced SIMD (*aka* ARM NEON instructions) now 12 years old
  - Integer, fixed-point and non-IEEE single-precision float, on *well-conditioned* data
  - 16×128-bit vector registers
- AArch64 Advanced SIMD was an evolution
  - Gained full IEEE double-precision float and 64-bit integer vector ops
  - Vector register file grew from 16×128b to 32×128b
- New markets for ARMv8-A are demanding more radical changes
  - ✓ Gather load & Scatter store
  - ✓ Per-lane predication
  - ✓ Longer vectors
- But what is the preferred vector length?

ARM

Introducing the Scalable Vector Extension (SVE)

- There is no preferred vector length
  - Vector Length (VL) is hardware choice, from 128 to 2048 bits, in increments of 128
  - *Vector Length Agnostic* (VLA) programming adjusts dynamically to the available VL
  - No need to recompile, or to rewrite hand-coded SVE assembler or C intrinsics

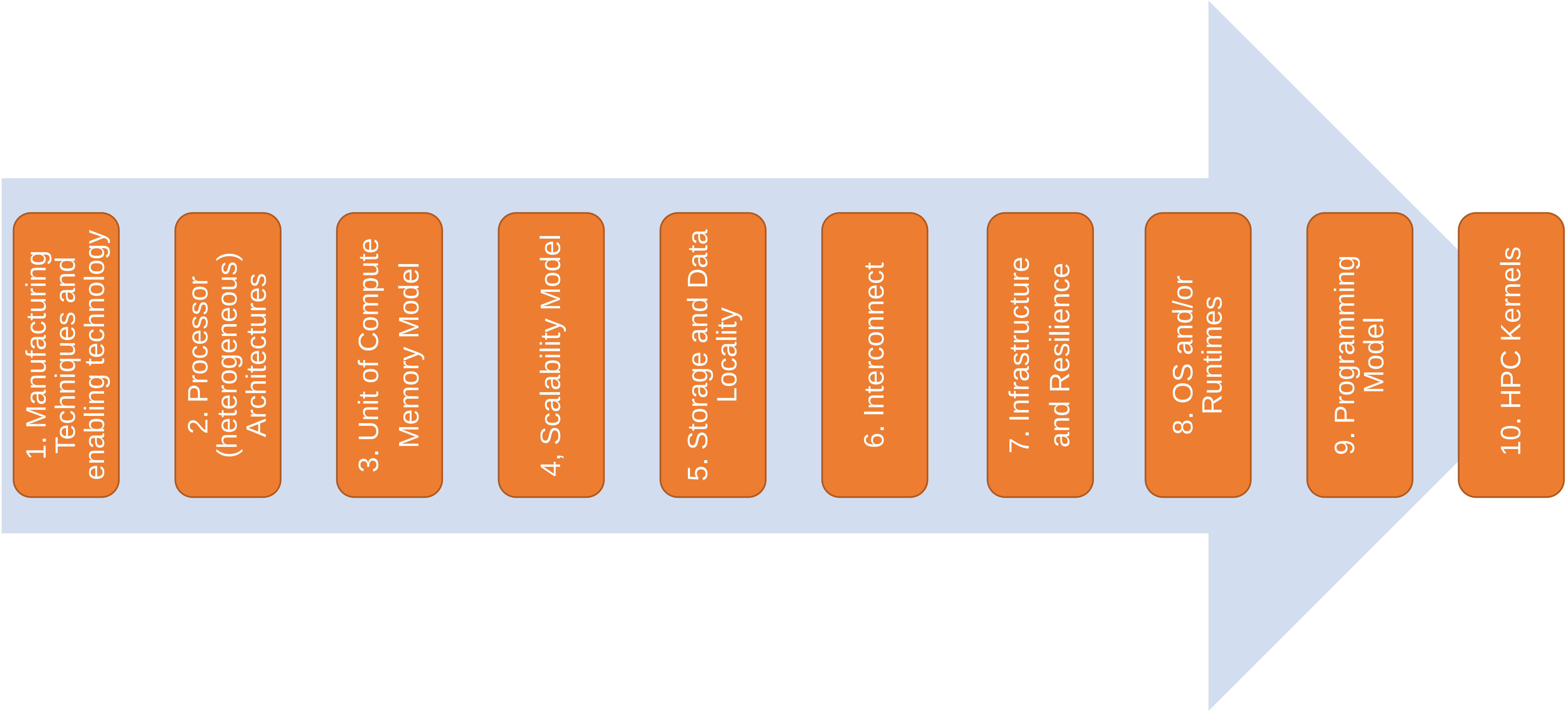
- SVE is not an extension of Advanced SIMD
  - A separate architectural extension with a new set of A64 instruction encodings
  - Focus is HPC scientific workloads, not media/image processing

- Amdahl says you need high vector utilisation to achieve significant speedups
  - Compilers often unable to vectorize due to intra-vector data & control dependencies
  - SVE also begins to address some of the traditional barriers to auto-vectorization

Next Steps

- SVE designed for partners wishing to enter HPC market with ARMv8-A
  - Lead partners are implementing SVE, see recent announcements at ISC16
- Beginning engagement with open-source community
  - Upstreaming of patches and discussions to start within weeks
    - LLVM, GCC, Binutils, GDB
    - Linux kernel & KVM
- General specification availability in late 2016 / early 2017
  - SVE Architecture Overview
  - SVE AArch64 ABI changes
  - SVE C/C++ intrinsics

Research vision:



…how to take the "EuroServer" approach towards exascale (FETHPC-2014)

EUROSERVER: The Unifying Background

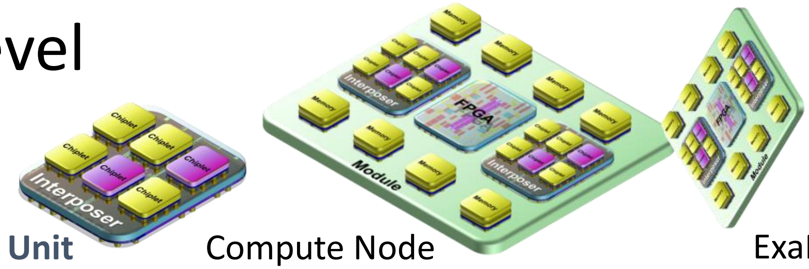
- UNIMEM shared memory architecture
  - Provides backwards SW compatibility while providing solutions to RAM limitation and software challenges
- Unit of Compute processing structure
  - Provides a scalability and modularity re-use approach for compute
- Share-anything scale-out
  - Removes the overhead costs of a share-nothing scalability approach
  - Enables lower cost market specific configuration optimizations
- Everything Close design goals
  - Lowers power and increased performance through data locality
- Silicon Chiplet approach
  - Reduces NRE and unit costs enabling market competition and solution specialization
- Virtualization enhancements
  - Ensuring increased manageability with lower resource cost
- Memory Optimizations
  - Reducing effects of memory bottlenecks while reducing energy of external data access

Unimem Memory Model

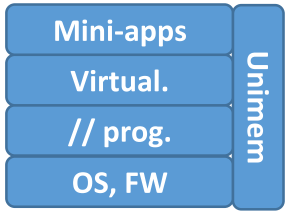
- Today's platforms have simple DRAM or DEVICE memory types
  - Sequentially consistent cached dram memory is very expensive
  - Even more expensive to scale beyond a single processor socket
- Key observations used by Euroserver
  - No need for sequential consistency in communicating / scaleout workloads
    - Applications tend to partition datasets and its memory access
    - Best to place the processor (and its cache) near the dataset of an application task (move task)
- Unimem extends today's memory model and enables:
  - Maintains a consistent and coherent access from each compute node to its local DRAM
  - Adds access to any system-wide memory resource by any workload through unimem
    - Allow local processors to cache local memory on remote accesses
    - Could support changing the cached ownership of any global memory region
  - Quite straight forward to add support in today's communication and shared memory API
- Can be implemented efficiently using ARM + SoC design principles
  - …does not require modifications to software applications
- Enables a platform for future systems and the push to exascale level power efficiencies

Theme 1: Manufacturing Technologies

- Efforts now concentrated in exaNODE, previously part of EUROSERVER
- Reduction in cost of “HPC” silicon device through silicon die reuse
  - Investigating best technologies to assemble a compute unit. Digital vs Analog bridges
  - Assembling an in-package compute node through addition of IO die
- Delivering the physical board that exposed UNIMEM for system scalability
  - Design of enabling firmware to join it all together
  - Virtualization to enable manageability, check pointing
- Evaluated at HPC mini-app level

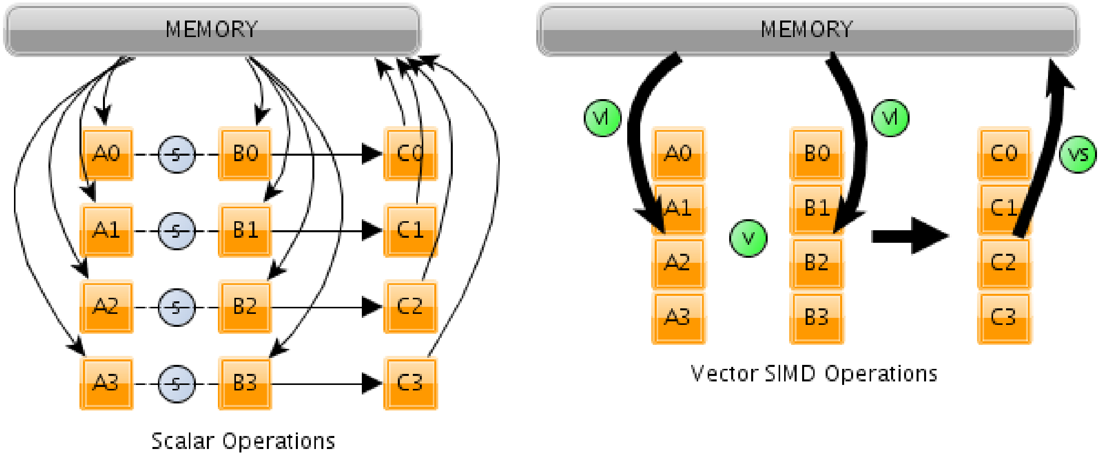


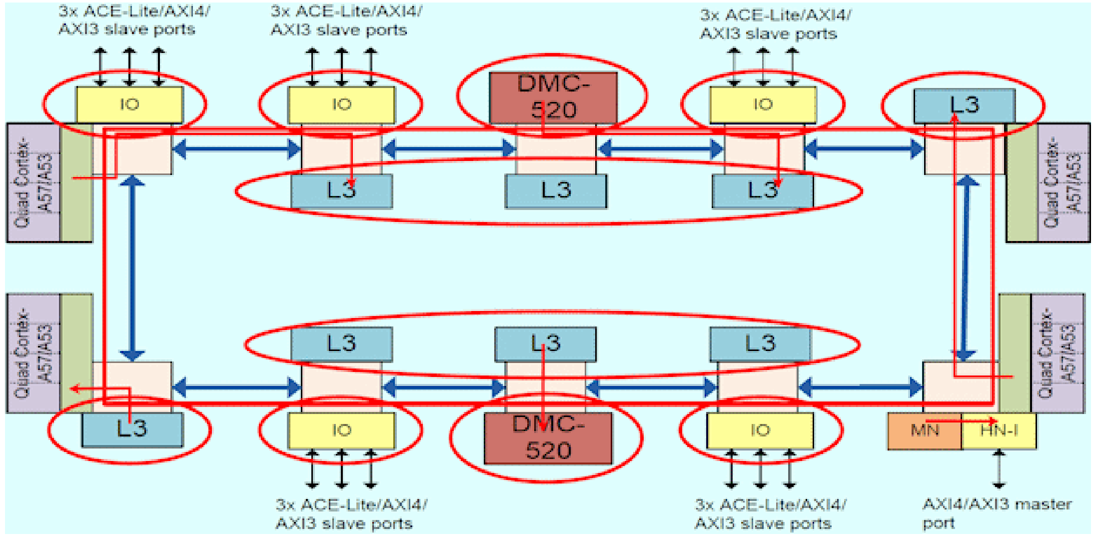
Compute Unit Compute Node ExaNoDe prototype



Theme 2: Processor Architecture

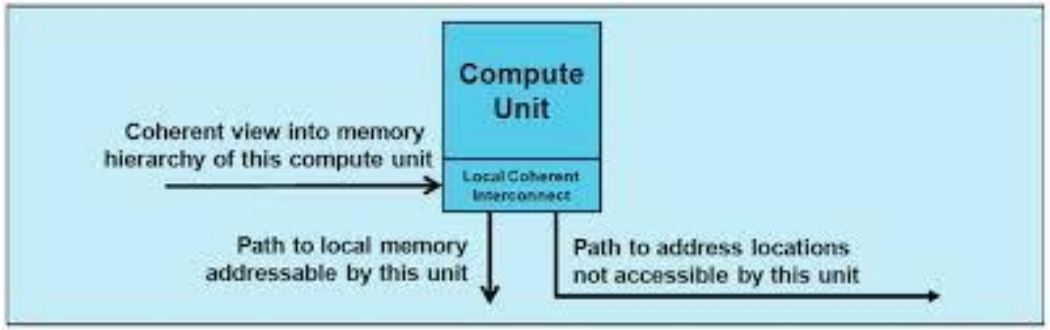
- Something ARM and its partners cover
- Instruction set architecture
- Targeted System Architecture
  - Support for accelerators
  - Unified memory support
  - Path to local memory
  - Path to/from remote memory





Theme 3: Unit of Compute

- Capabilities prototyped and evaluated in EUROSERVER
  - First discussed at DATE 2013
  - Provides the unit of system scalability



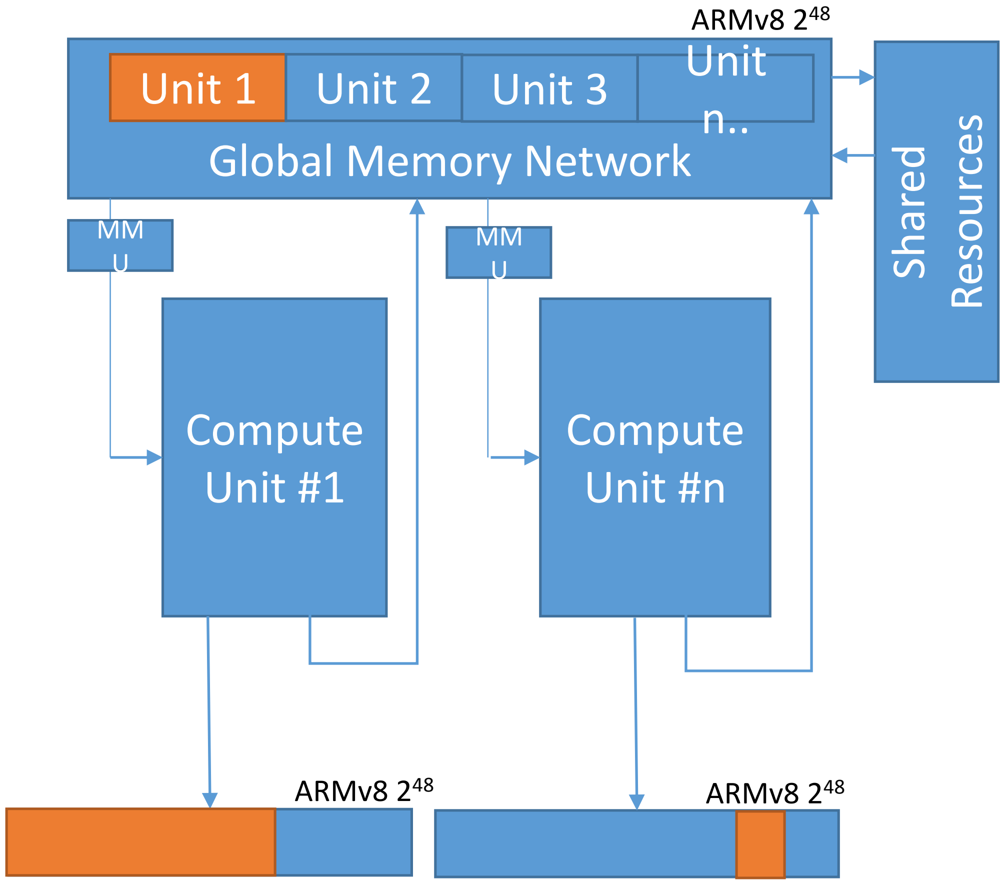
- Processor Agnostic
  - The unit can be any architecture
  - Supports heterogeneity within and between units

- Local resources manage the bridge to/from “remote memory”
  - Mapping of remote address space into local physical address space
  - Defined by only compute and memory resources

- Each Compute Unit is registered at a partition within a system’s global address space (GAS), including units with heterogeneous capability
  - Any unit can access any remote location in the GAS (including cached)
  - DMA can transfer between (virtually address cached) memory partitions

Theme 4: Scalability Model

- First prototyped in EUROSERVER using direct chip-2-chip NoC bus
  - Extended in ecoSCALE to include FPGA acceleration memory and resource model
  - exaNEST developing inter-device bridge and system level global memory interconnect
- IO resources are shared at Global Network level
  - Expected implementation within package
  - Reconfigurable hardware can be used to deliver IO capabilities using "physicalization"
- Difference configurations enable use across different markets
  - EUROSERVER "spinout" targets micro-server



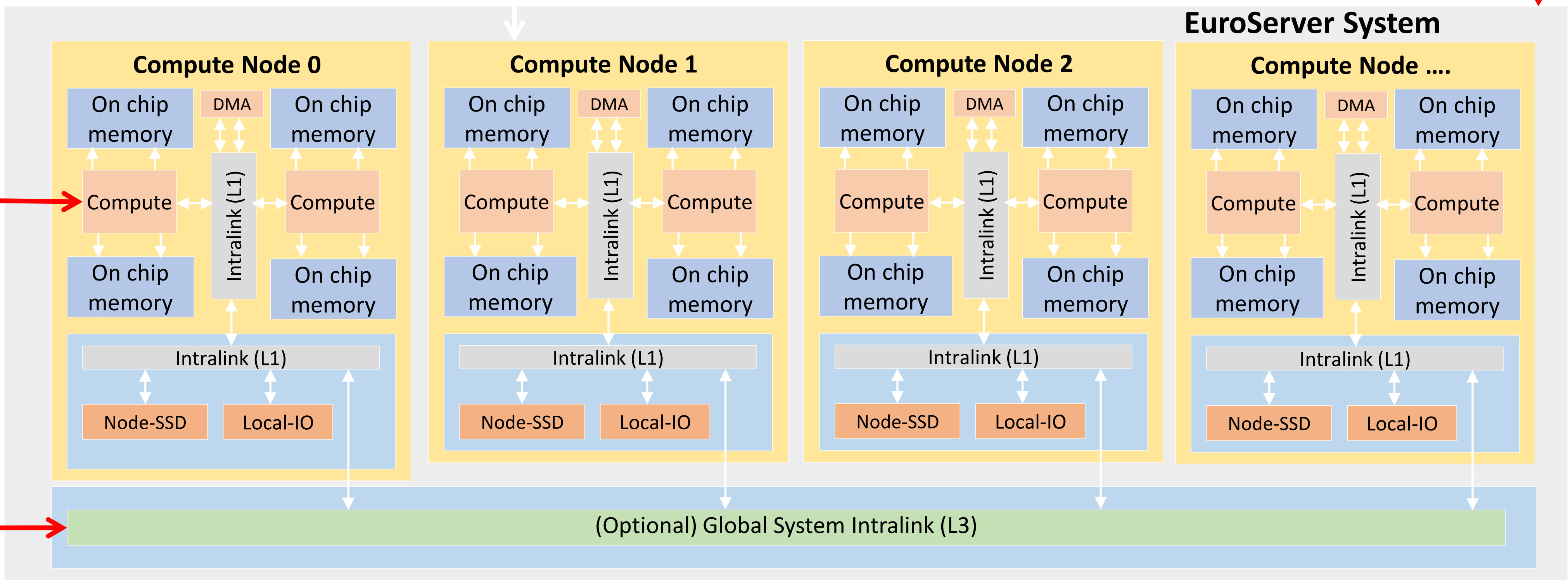
Share-Anything - System Scalability

Compute Unit:
One or more processor cores
Level-0 Interconnect
Single coherence island

Node:
One or more unit as chiplets
Level-1 interconnect
Shared IO Resrouces (eg Ethernet and Storage)

µServer: (EuroServer)
1 or more Nodes
Scale-out server using Local-IO
or HPC via Level-3 interconnect

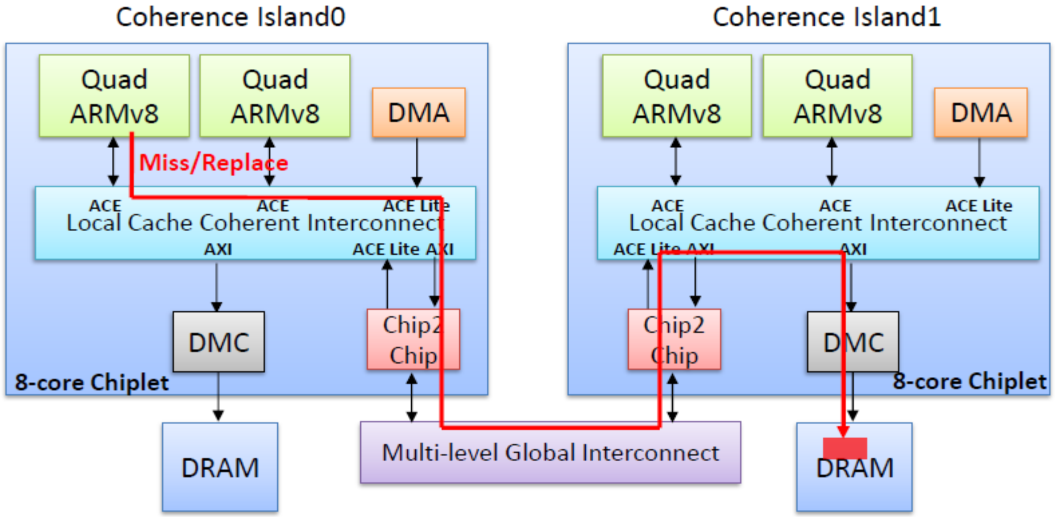
HPC System:
Multiple Nodes sharing Level-3 interconnect (topology agnostic)



- Each Coherence Island has its own local independent global (coherent) address space (GAS^L)
- Coherence Islands communicating through multi-level Interconnect
- Sharing via page mapping a common remote global address space (GAS^R)
- Either Remote DMA or direct Remote Load/Store from application virtual page mapping

DRAM in a single application

Remote Page Borrowing

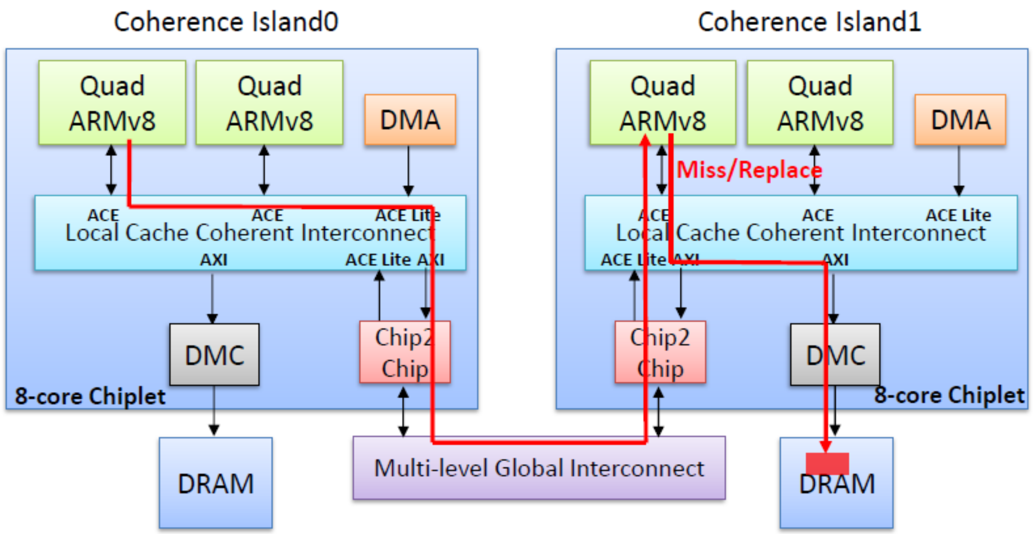


- Locally cacheable (initiator's cache)

- Use the DRAM physically connected to different coherent islands in same and remote devices
- Allows a RAM demanding application access to capacities higher than can be supported by a single device
- Memory consistency rules allow peer memory (same package) to be have similar latencies to local memory

Scalable "shared memory" model

Shared Memory



- Remotely cacheable (owner's cache)

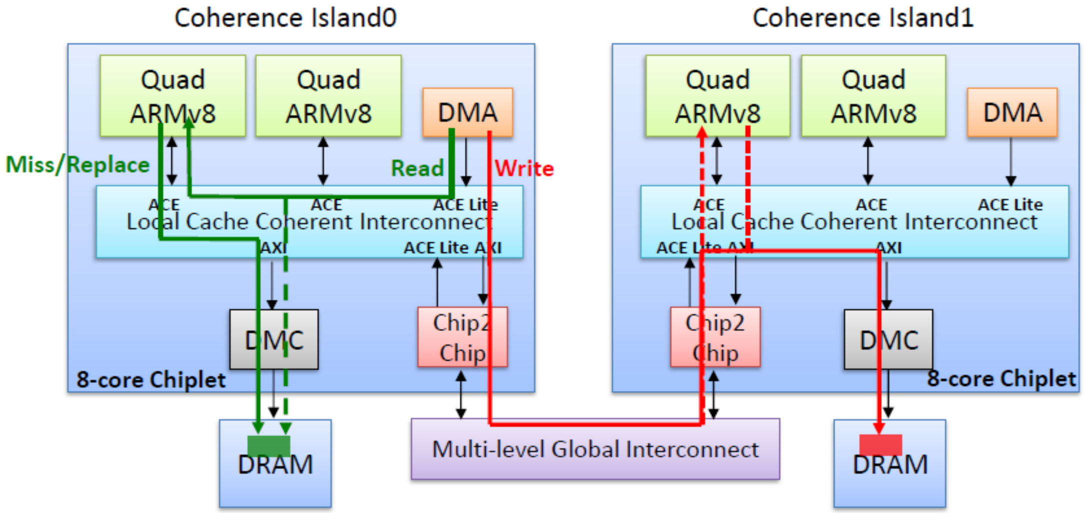
EURO SERVER

FORTH-ICS: Computer Architecture and VLSI Systems (CARV) Lab

- Allows multiple independent coherent islands to share global addresses
  - Virtual mapped DMA copies
  - Absolute shared address pointers
  - Memory based synchronizations
  - Can be managed via Numa OS
- No inter-island coherence protocol
  - No coherence directory
  - Direct "coherent" r/w between islands
- Pipelined/blocks for high bandwidth
- Native processor addressing for low latency communication

Low Latency Communications

RDMA



- DMA reads from (or writes to) DRAM on Coherence Island0 and writes to (or reads from) DRAM on Coherence Island1
- Accesses can also be uncacheable locally or cacheable remotely (dashed lines)

EURO SERVER

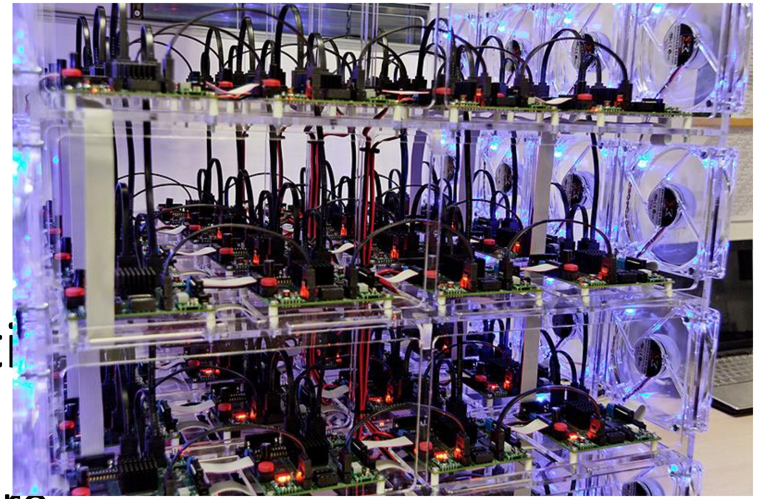
FORTH-ICS: Computer Architecture and VLSI Systems (CARV) Lab

- Either direct load/store for single transaction communication, or virtually mapped DMA for block transfers
- Aliased memories for broadcasts
- Native processor addresses used for non-abstracted communication
- Consistency rules allow data movement directly between LLC of nodes

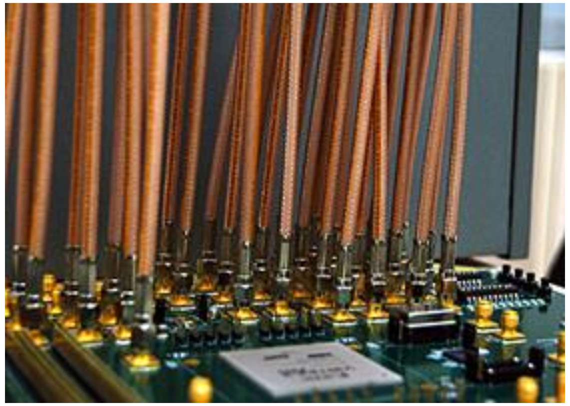
Theme 5: Storage and Data Locality

- EUROSERVER introduced the "every-close" design paradigm
  - exaNODE board design to use converged compute/storage/network deployment scenario
- Storage devices located within millimetres of the processor
  - Enables ultra-short reach physical connection technologies to minimize power and latency
  - Option today is to use "detuned" PCIe to reduce drive power
  - Shared distributed global storage sharing the common Global Network bridge between nodes
- Compute unit main memory extended with "storage-class" NVRAM
  - Fit within memory hierarchy as transcendental cache by hypervisor to provide over-commit of DRAM
    - EuroServer "spinout" using single embedded 128GB flash device
    - ecoSCALE option to use discrete DDR4 to overcommit main SODIMM
  - System architecture "waiting" for real storage class memories

Theme 6: Interconnect



- Currently progressing through exaNEST
  - Can be traced back to initial work in ENCORE
- Exposed as a "physicalized" interface into applicati address space
  - Moving towards zero-copy between application and wire
  - Hardware accelerated and managed interface



- Researching topology, resilience, congestion control…
  - Targeting evaluation of 160Gb/s per node of four compute units

Theme 7: Infrastructure and Resilience

- Current infrastructure limited to around ~800W per blade due to physical size and significant localized hotspots
  - First phase of exaNEST will exchange processing technology and evaluate the effect in removal of hotspot on compute density
  - Phase 2 expects to be able to double compute density to over 1.5kW / blade
    - …petaflops per rack ?
- Manageability and software resilience using virtualization approach
  - Check-pointing
  - Software defined/managed storage/networking
- Evaluated running real applications
  - 1,000 cores, 4TB DRAM testbed

Iceotope Ltd:
Fully Immersed Cooling Technology

Theme 8: OS and runtimes

- Spread across each of the EuroExa projects, and others
- Linux kernel extended to understand management of remote memory
  - Unimem API then used by various standard shared-memory libraries
- BeeGFS distributed file system is being extended to understand hardware memory model enhancements
- The large global memory capability significant for in-memory DB
  - MonetDB
- HPC runtimes
  - MPI, PGAS, OpenStream and OmpSs also being ported and enhanced

PGAS

mmap

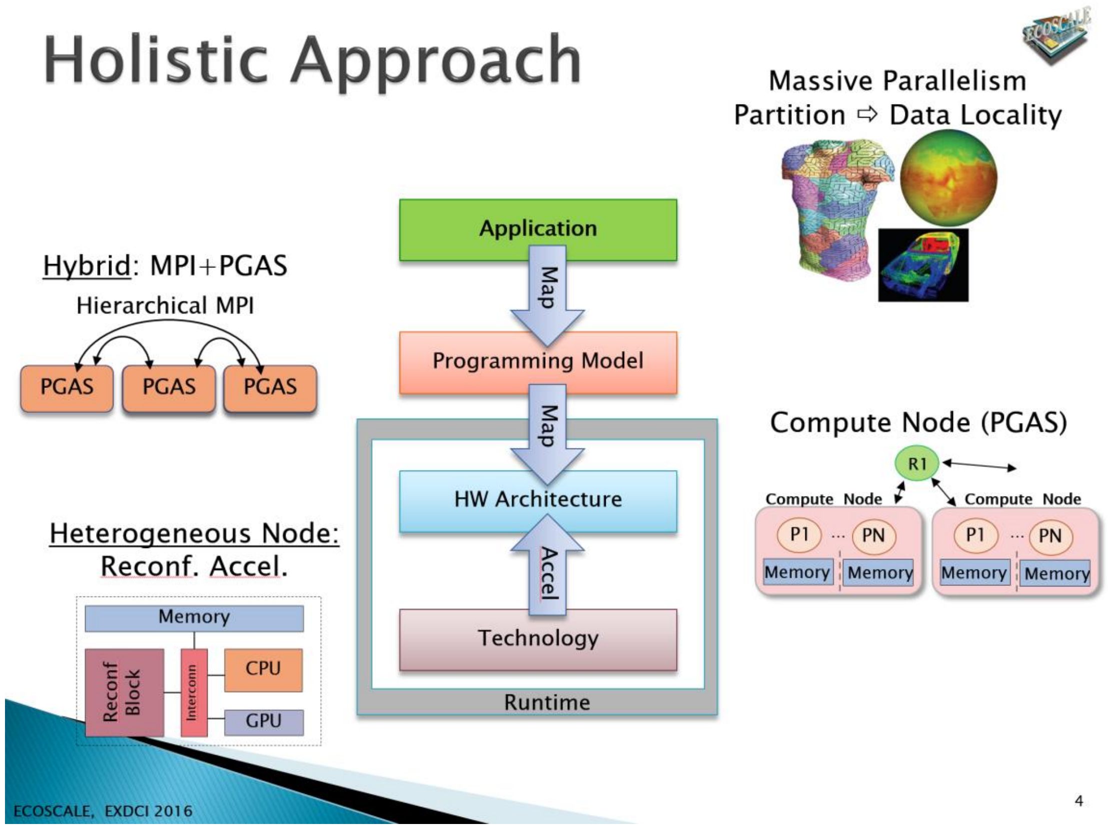
RDMA

Sockets

Theme 9: Programming Model

- Domains communicate using MPI
- Within a Domain PGAS is used to access global memory
- OpenMP/OmpSS within the compute unit

- Accelerators using coherent unified memory approach
  - Accelerator can create “global” pools of resource through UNIMEM
  - Exposed using standard API such as OpenCL
  - Focusing on reconfigurable compute acceleration
    - Partial reconfiguration used to manage resource pool



Theme 10: HPC Kernels and Applications

- Initial participation from the HPC community in each of the EuroEXA projects
- Evaluating the impact and capability of UNIMEM at the mini-app/kernel level
- Testing the scalability model and interconnect through real applications
- ….time to move to a true co-design over the sizing and choice of hardware components and the requirements and evolution in the design of full applications
    - FETHPC-2016 co-design
    - Looking at assembly of a HPC specific device
    - Creation of a at-scale testbed platform

Single Slide: How they all fit together

EUROSERVER: The first to test and realize -
- A reusable "Chiplet" based delivery for silicon devices
- The UNIMEM share-anything, "remote memory" paradigm
- Scalability model of "Compute Units"

ExaNODE:
- Mature the "chiplet" approach
- Define a HPC "node" and build a physical PCB
    - Technology to group multiple compute units
    - So to enable sharing of peer memory
- Test and enable HPC kernels and runtimes

ExaNEST:
- Define the "remote memory" capable interconnect
- Share the interconnect with distributed storage
- Design a cooling and system to efficiently maximise the deployment of the ultra-dense designs
- Extend the HPC community and software support

ecoSCALE:
- Extend the unimem model to FPGA / reconfigurable accelerator
- Enable FPGA to be an efficient, unified memory accelerator
- Find the limits of the UNIMEM model and define exascale model

..continuing the vision

EuroEXA: Mature the vision.
- Co-design the system metric
- Create a European HPC pilot device
- Bring a system together
- Evolve the systems maturity
- Test and evaluate at Petaflop level

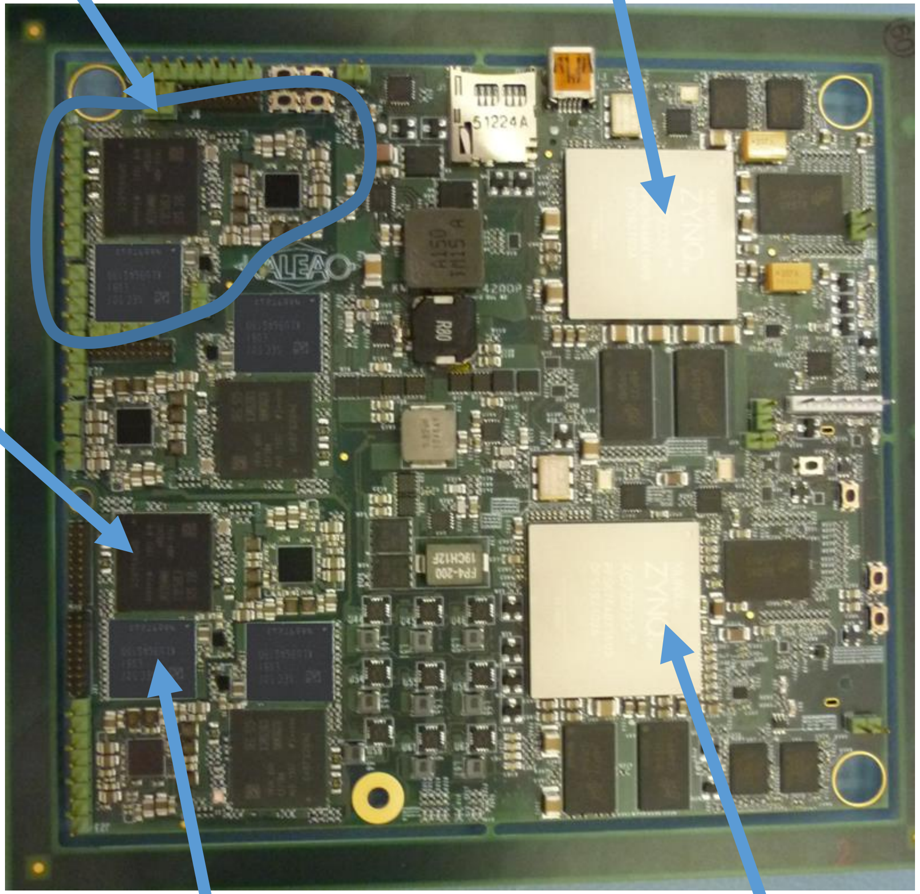
Create a focus for European "crowdfunding" of HPC
- To define market size through extreme scale demonstration
- Secure the product level components

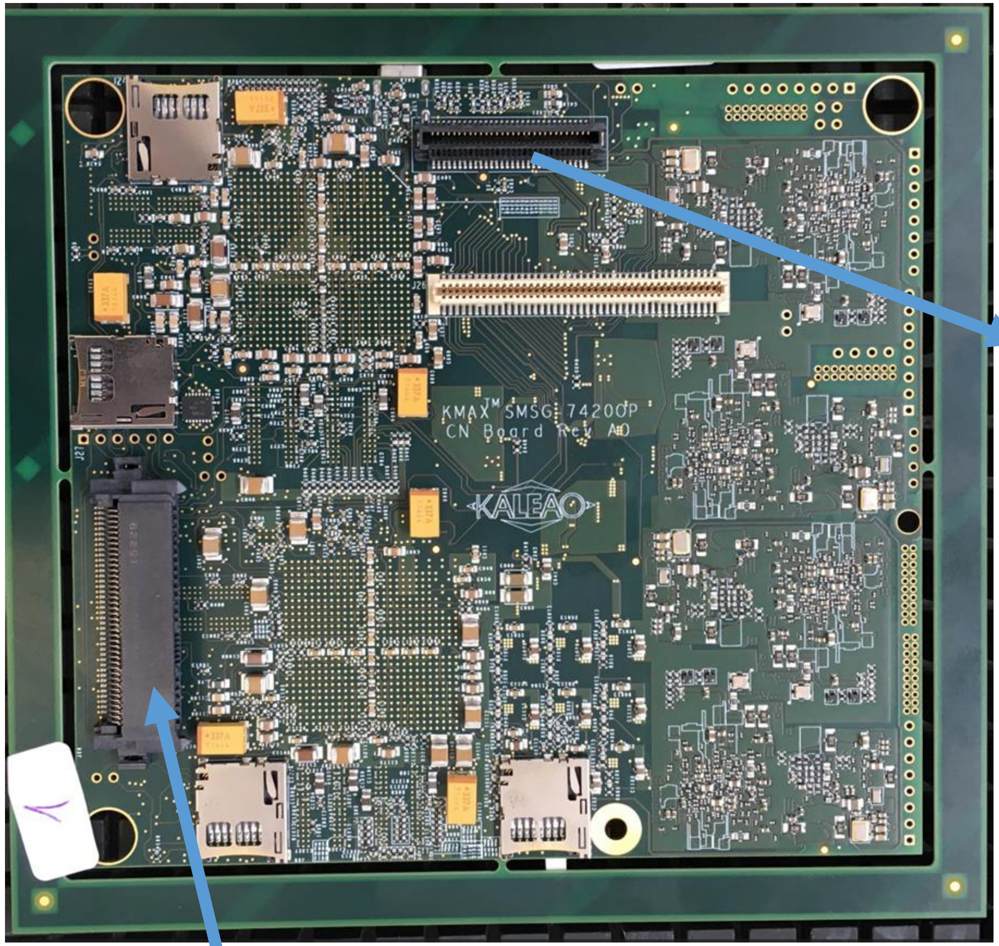
What this looks like for scaleout

Multiple compute units
Sharing the nodes resources

Physicalized multi-NIC
10-40Gb adaptor

In package
Processors and DRAM





Networking up to embedded
blade level 10/40Gb switching

New level in memory
heirarchy per unit

Hardware accelerated
distributed storage controller

NVMe SSD close to compute
to reduce power to access local data

See http://www.Kaleao.com

Concluding remarks –Towards exascale projects

- Lets assume we can have a flat, optically switched, 200 or so racks to keep the physical size manageable
  - …that supports an efficient way to share global state (GAS) and communicate between racks

- This proposed architecture with apps in the 10 or so flops per byte range would offer:
  - Exascale at around 60 to 70MW when working from a few GB of on-chip memory. (ok for a benchmark!)
  - More RAM capacity will cost something like 5MW for every 5mm it sits away from the processor

- Targeted system level optimisations should move this to 50 to 60MW in next couple of years.

CONCLUDING REMARKS – Crystal Ball

- To get it lower, then the flop per byte accessed ratio must be increased
  - So that the FLOPS physical silicon can deliver within its thermal / die size limits can be balanced with the IO count and interface speed that can be used to connect memory
  - Maybe an application target of at least 100:1 of FLOPS/byte would be nice ;)
  - I see this will need apps/kernels to move to a data flow or functional type of paradigm so as not to store to forward intermediate values – along with the unified microarchitectural accelerators that explicitly support these models
  - If this happens, then maybe we could see Exascale at around 50mw by 2020.

- To get lower than this, I believe a new blade-level (< 30cm) conductive material is required:
  - Today's optical/photonic approaches won't solve it unless they can build lasers that are ~100 times more efficient. (or that 200-way optical switch grows to support 100K's nodes)
  - If carbon nanotube impregnated materials are indeed more than 10x better conductors than todays flex-cables…
  - …then you might reach 40 to 50mw, but not before 2024 so as to have time to integrate the new material

- Any lower than this will also need a materials change within the "processor" or a way to run at superconducting levels.

Thank you

Time for questions ?