

HPC Future Look

Exascale and Challenges



Reusing this material



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Acknowledge EPCC as follows: “© EPCC, The University of Edinburgh, www.epcc.ed.ac.uk”

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

Outline

- Future architectures
 - Exascale initiatives
 - Processors
 - Memory
 - Impacts on performance
- Software challenges
 - Parallelism and scaling
 - New algorithms
 - What about software that does not scale?
- Impact for standard computing

Future architectures

What will HPC machines look like?

What will future systems look like?

	2016	2019	2022
System Perf.	35 Pflops	100-200 PFlops	1 EFlops
Memory	1 PB	5 PB	10 PB
Node Perf.	200 Gflops	400 GFlops	1-10 TFlops
Concurrency	32	O(100)	O(1000)
Interconnect BW	40 GB/s	100 GB/s	200-400 GB/s
Nodes	100,000	500,000	O(Million)
I/O	2 TB/s	10 TB/s	20 TB/s
MTTI	Days	Days	O(1 Day)
Power	10 MW	10 MW	20 MW

Processors

- More Floating-Point compute power per processor
 - Only exploit this power via parallelism (SIMD instructions)
 - Lots of low power compute elements combined in some way
- Processors look more like GPU or Xeon Phi than traditional CPU
- More cores per node
 - Clock speed not increased over current values and may decrease
- Increased flexibility for core use through containerisation/virtualisation technologies

Memory

- Will be packaged with processor
 - Increases power efficiency, speed and bandwidth...
 - ...at the cost of smaller memory per core
- Memory hierarchy will become more complex
 - Still unclear how this will be exposed to developers
 - Additional levels of shared cache
 - Multiple levels of memory with different performance characteristics
- CPU/Accelerator will share coherent memory space

Interconnect

- Modest increases in bandwidth
 - Not keeping up with growth of compute power per node
- Not much change in latency
- Topologies probably will not change much
- Links will move closer to the processor
 - Maybe interface will be even be on the processor itself
- Additional features
 - Fault tolerance
 - Improved remote memory addressing
 - Direct links to accelerators
 - ...

Storage

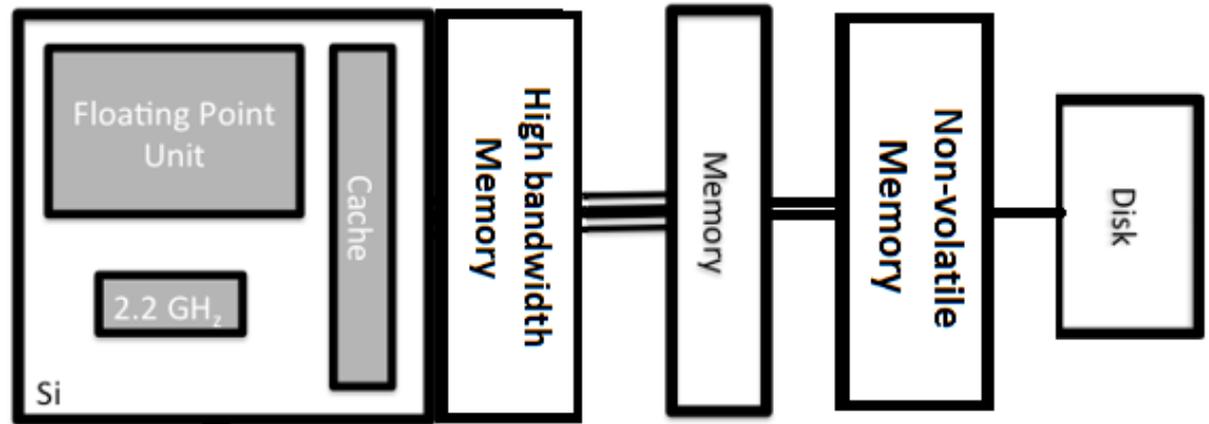
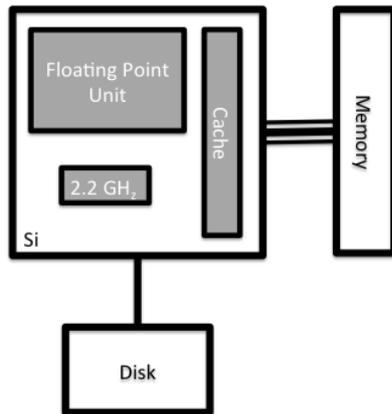
- Increased complexity of storage stack
 - Additional levels of hardware with different performance characteristics: e.g. solid state data staging
 - How will this be exposed/managed?
- Rise of non-file system technologies
 - Growth of object storage
- Ability to efficiently support modern data analysis tools
 - e.g. Apache Spark
 - Graph analysis

System on a chip

- Instead of separate:
 - Processor
 - Memory
 - Network interface
- Combined system package where all these things are included in one manufactured part
 - This is the only way to improve power efficiency
 - Less scope for customisation
 - If you need more memory than in package you will have to have levels of memory hierarchies

Memory hierarchies

- Moving from: To something like this:



Example: ARCHER2 vs ARCHER

- ARCHER had:
 - 4920 nodes connected by the high-speed ARIES Cray network
 - each node is a 24-core, shared-memory, NUMA machine
 - 2 x 2.7 GHz Intel E5-2697 v2 (Ivy Bridge) processors
- ARCHER2 will have:
 - 5,848 nodes connected by the new Slingshot Cray network
 - each node is a 128-core, shared-memory, NUMA machine
 - 2 x 2.2GHz AMD Rome 64-core CPUs
- Roughly 10 times more powerful (peak around 28 PFlops)
 - about the same number of nodes
 - but nodes are much more powerful (with many more cores)
 - no GPUs in main production machine

Example: Cirrus2



- Current Cirrus system almost entirely CPU-based
 - only two GPU nodes
- Being extended with 144 additional GPUs
 - same architecture as current GPU nodes
 - NVIDIA Tesla V100 (Volta)

Software challenges

What does software need to do to exploit future HPC?

What does this mean for applications?

- The future of HPC (as for everyone else):
 - Lots of cores per node (CPU + co-processor)
 - Little memory per core
 - Lots of compute power per network interface
 - Increased complexity in memory and IO hierarchy
- The balance of compute to communication power and compute to memory are both radically different to now
- Must exploit parallelism at all levels
- Must exploit memory/IO hierarchy efficiently

Algorithms

- For many problems new algorithms will be needed
- May not be optimal but contain more scope for parallelisation
- Mixed-precision will become more important

Applications that do not scale

- The good news is that if you do not need to be able to treat larger/more-complex problems then you can access more of current resource size
 - May be caught out by decrease in memory per core
 - Options to scale in trivial-parallel way: increase sampling, use more sophisticated statistical techniques
 - This may well be the best route for many simulations

Impact on standard computing

What does this mean for my workstation/laptop?

Parallel everywhere

- All current computers are parallel
 - From supercomputers all the way down to mobile phones
 - Most parallelism is task-based on 4-8 cores – each application (task) runs on an individual core.
- In the future:
 - More parallelism per device – 10s to 100s cores running at lower clock speeds
 - All applications will have to be parallel
 - Parallel programming skills will be required for all application development.
- More system on a chip – more things will be packaged together

Summary

Summary

- Additional compute power mostly from increased parallelism at processor/socket level
 - Due to power constraints
- Balance of compute to memory/interconnect/storage is orders of magnitude different to current systems
- Increase in complexity of memory/storage hierarchy
- All lead to challenges for software developers
 - New algorithms may be required with different performance characteristics
 - How to deal with more complex hardware hierarchies?