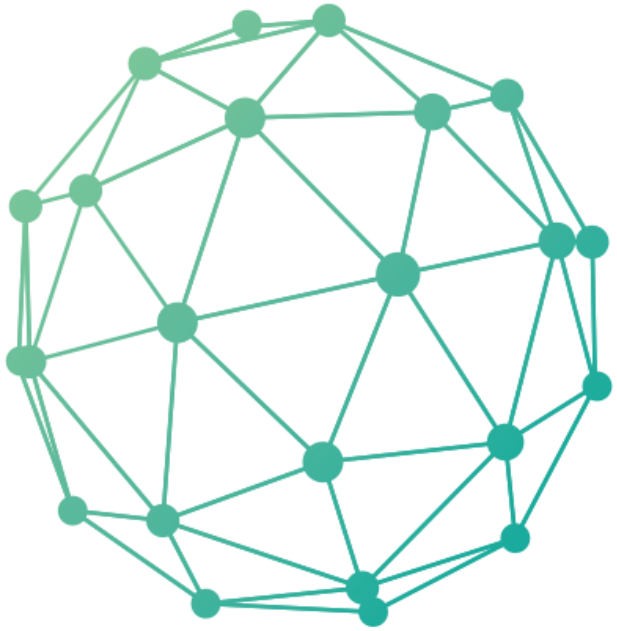
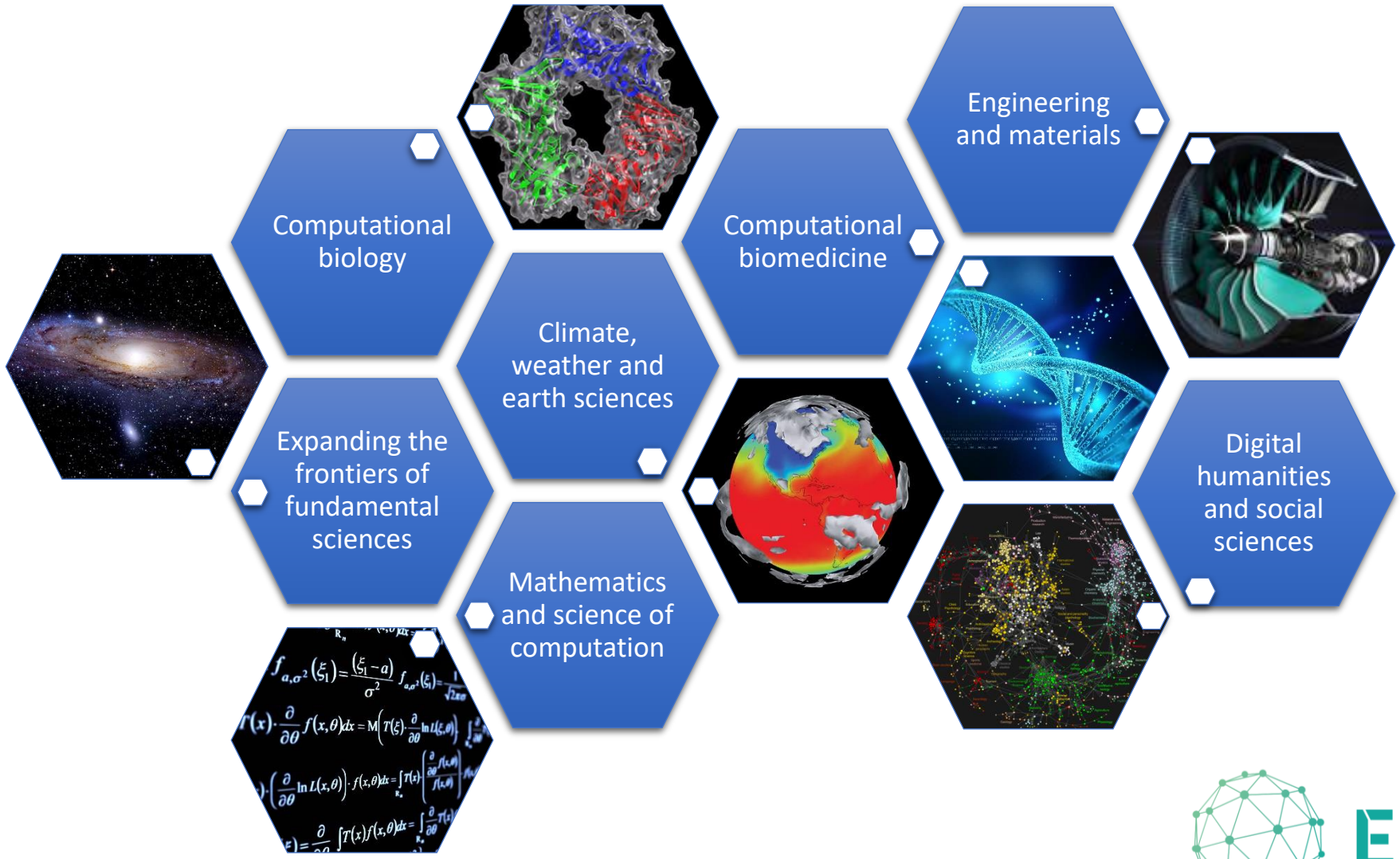


ELEMENT

Implications of Exascale Hardware for Mesh Generation

09/06/2021

Supercomputing Science Case completed

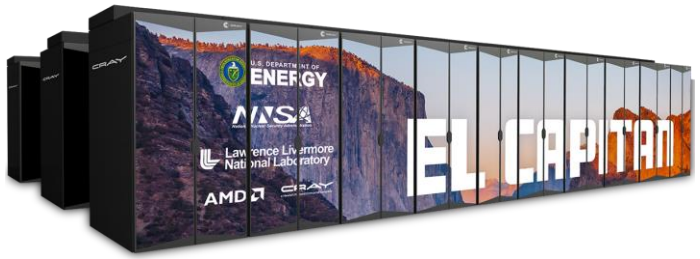


ELEMENT

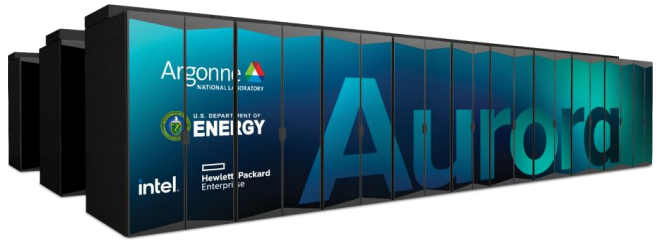
Current state of the art

- Exascale hardware expected in the 2022-23 timeframe
- Mesh generators generally developed for single-user work stations
  - Interactive *vs* batch processing

By Thomas Schanz - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=30234789



El Capitan & Frontier: https://www.amd.com/en/products/exascale-era



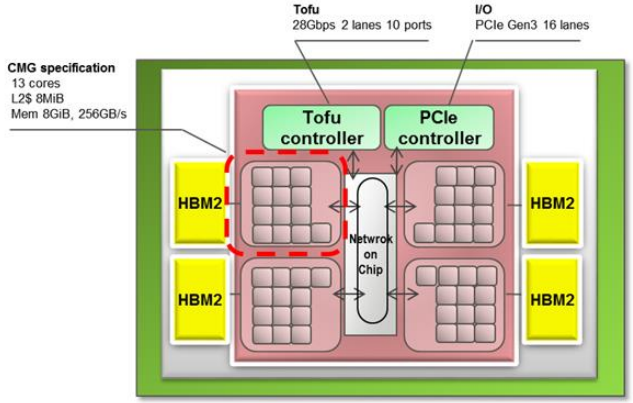
Aurora: https://www.intel.co.uk/content/www/uk/en/high-performance-computing/supercomputing/exascale-computing.html

ELEMENT

Current state of the art: Scale of parallelism

- Current machines approaching or exceeding $\mathcal{O}(10^6)$ threads of execution
  - ARCHER2: 750k cores, 1.5M threads
  - Order of magnitude more parallelism ➔ Exascale[1]
- Unlikely a mesh generator *needs* to run at this scale
- Possible mesh generator will be *forced* to run at this scale, or at least on this type of hardware
- Existing programs can exploit parallelism
  - Octree-like algorithms seem well-suited to parallel mesh generation, demonstrating scalability to $\mathcal{O}(10^5)$ threads *Ghisi et al. (2014)*

Fugaku: fujitsu.com



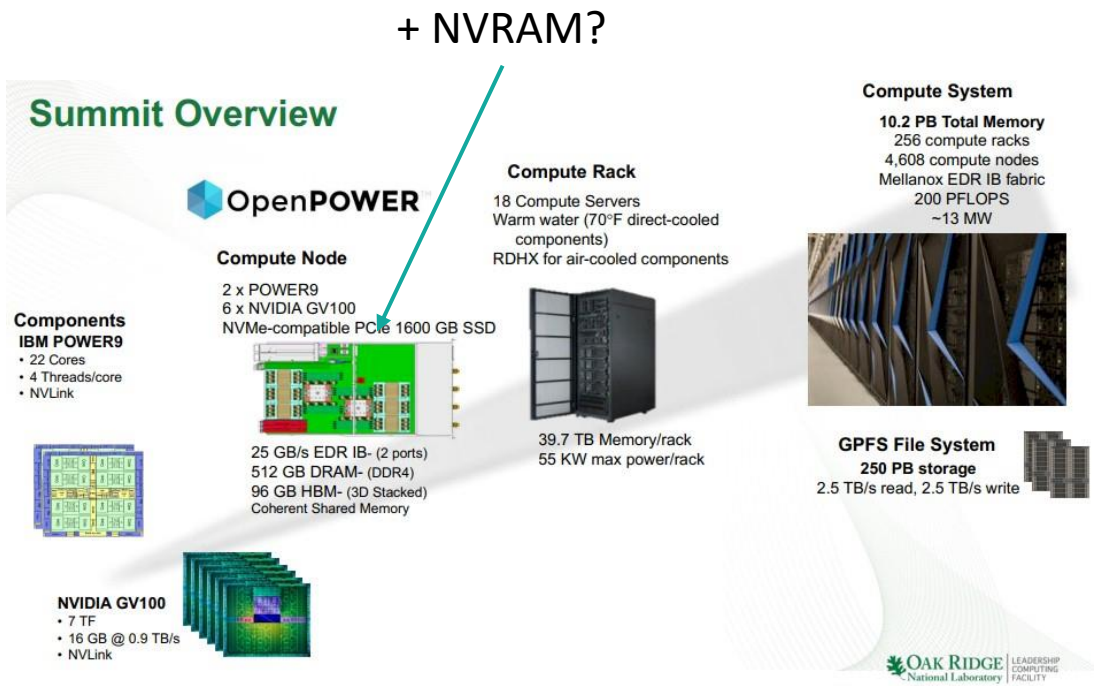
A64FX: fujitsu.com

[1] Parsons, M.; ELEMENT workshop 1

ELEMENT

Current state of the art: Accelerators and performance portability

- Changing environment
  - Not just traditional simulation, ML/AI workloads
  - Need to support industrial and scientific users
- Accelerators are becoming the norm
  - All 3 US planned Exascale supercomputers GPU-accelerated
  - EU: LUMI (550 PF), Leonardo (248 PF) GPU-accelerated
  - Fugaku (Japan) bucks this trend
    - Successful co-design effort
- If running in-place, performance portability will be required
  - Similar to what has been seen in solver developments
- New hardware opportunities/challenges[1]
  - Ray-tracing
  - Larger vector units
  - Vector scatter/gather operations
  - GEMM-type operations in hardware
  - Faster/higher BW memory, persistent fast memory
  - Chiplets and complex memory hierarchies
  - Support for low precision ML operations
  - **Really exciting time to be working in HPC**



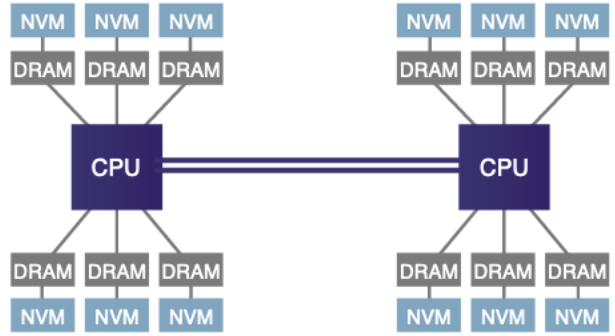
SUMMIT hardware: www.nextplatform.com

[1] Parsons, M.; McIntosh-Smith, S.; Homölle, B.; Dubé, N.; ELEMENT workshop 1

ELEMENT

Limitations

- Memory per core trending down
- Per core memory bandwidth also decreasing, favouring FLOPs over bandwidth (FLOP monsters)
  - HBM2 (*e.g.* A64FX) working in opposite direction, but limited capacity
- Underpopulating (i.e. not using all available cores on a CPU) can help with bandwidth
  - Reduced memory capacity will force distributed parallelism
- Mesh size may be a significant data handling issue
  - Need to generate in-place?
  - Need to generate on demand?
- Checkpointing may become necessary at scale
  - Single-user workstation focus means overlooked so far
  - Concerns over Mean-Time to Failure at scale
  - *Tan et al. (2019)* demonstrated use of NVRAM as a form of semi-persistent memory for AMR[1]
    - Need to account for asymmetric performance



NEXTGenIO: www.nextgenio.eu

[1] Tan, H. et al.; Persistent octrees for parallel mesh refinement through non-volatile byte-addressable memory; IEEE trans. PARALLEL distrib. syst. (2019)

ELEMENT

Goals for Exascale

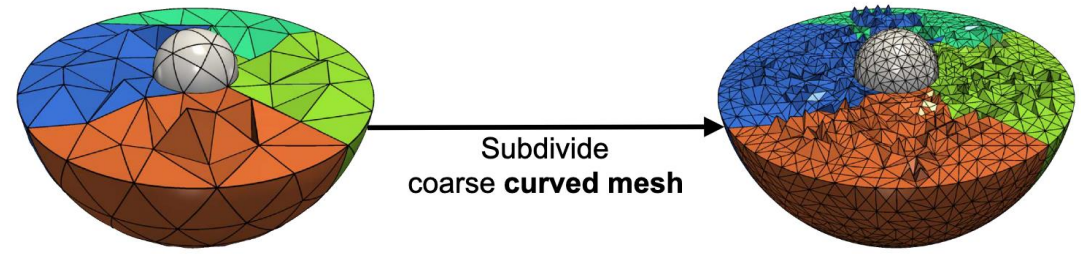
- We need to be able to generate meshes required by Exascale problems – both scale and quality
- The size means we may be forced to generate meshes on the Exascale machines
  - The use of parallelism may not be driven by time to mesh
  - Not suited to interactive use
- Heterogeneous architecture is expected to become the norm
  - We need to be able to use the hardware we have access to
  - Performance portability

ELEMENT

Research agenda: Short term (1-2 years)

Mesh compression

- Facilitate movement between machines
- IO benefits through smaller data size
- Could we represent the mesh+geometry as a coarse (high-order) mesh + sequence of refinements?[1]
  - Would ease generating the mesh on-demand at scale
  - Replicable refinements ➔ independence of order



(Gargallo-Peiró, Ruiz-Gironés, Houzeaux & Roca, ICOSAHOM'18)

[1] Roca, X.; Meshing from CAD vision: curved adaption to geometry and solution; ELEMENT workshop 1; https://epcced.github.io/ELEMENT/assets/css/slides/Slides_Roca.pdf

ELEMENT

Research agenda: Short term (1-2 years)

Use of parallel programming frameworks

- Aids performance portability
- May still require (significant) rewrite[1]
- Requires identification of a core kernel(s) to test *cf.* Babel Stream[2]
  - Definition of standard set of benchmarks
    - Input from industry, scientific use cases
  - Definition of key metrics: cells/s, cell quality, mesh accuracy, …

[1] Eichstädt, J. et al.; Accelerating high-order mesh optimisation with an architecture-independent programming model; Comput. Phys. Comm. (2018)

[2] Babel Stream: https://github.com/UoB-HPC/BabelStream

ELEMENT

Research agenda: Short term (1-2 years)

Assessing use/suitability of heterogeneous hardware

- Linked to performance portability
- What are the benefits of new hardware?
  - Accelerators/novel compute units
  - Deep memory hierarchies
- Do our algorithms require a rethink (answer is almost certainly yes)?

ELEMENT

Research agenda: Medium term (5 years)

Evaluating meshing applications on Exascale hardware

- Short-term research overlaps early Exascale machines coming online
- Reasonable timeframe to begin testing the outcomes
  - Meshing for complex scientific/industrially relevant applications at Exascale
  - Can we efficiently exploit the available hardware?
    - Using metrics identified in the short term

Research agenda: Medium term (5 years)

Improving the efficiency of meshing workflows

- Exascale meshing must become first class citizens of the workflow
  - Must address IO and data management challenges
    - *In-situ* (de-)compression and verification of mesh
    - Marshalling data to/from compute nodes
- Robust “hands-off” mesh generation/adaptivity is a must!
  - One of the core points of CFD vision 2030

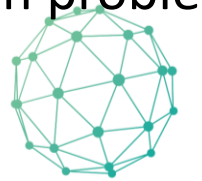
Research agenda: Long term (10 years)

Standardisation

- What has been found beneficial?
- Encourage reuse of optimised components
  - APIs – mesh generation, adaptation
  - Data exchange formats (storage and in-memory)
- Interoperability - MDAO is a significant use case for Exascale
  - Meshes for each “Discipline” need to be connected

Review progress

- Assess against other parts of the workflow
  - Demonstration of efficiency at Exascale
  - Actual experience with Exascale machines will uncover unforeseen problems



ELEMENT