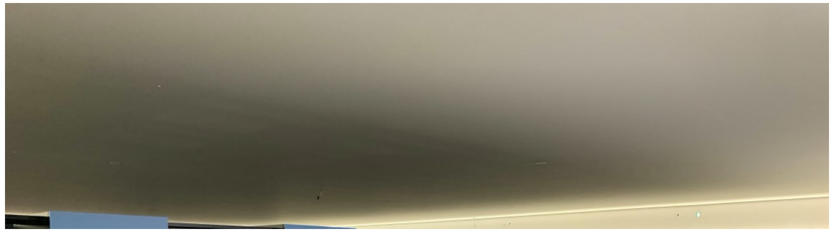
Prof. Simon McIntosh-Smith
Head of the HPC research group
University of Bristol, UK

Twitter: @simonmcs
Email: simonm@cs.bris.ac.uk

Supercomputer trends

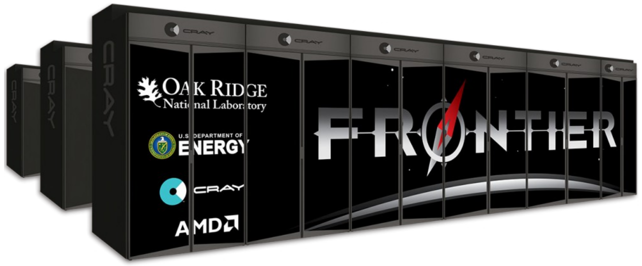
http://uob-hpc.github.io

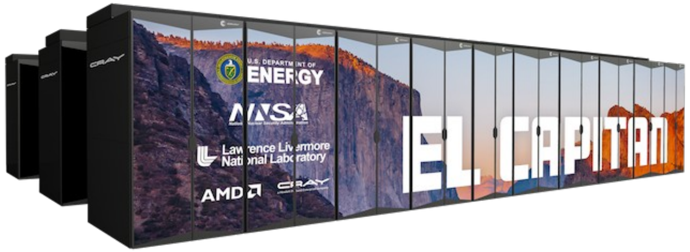
Next-generation supercomputers will be increasingly diverse



The coming generation of Exascale systems will include a diverse range of architectures at massive scale, all of which are relevant to weather/climate:

- **Fugaku**: Fujitsu A64FX Arm CPUs
- **Perlmutter**: AMD EYPC CPUs and NVIDIA GPUs
- **Frontier**: AMD EPYC CPUs and Radeon GPUs
- **Aurora**: Intel Xeon CPUs and Xe GPUs
- **El Capitan**: AMD EPYC CPUs and Radeon GPUs

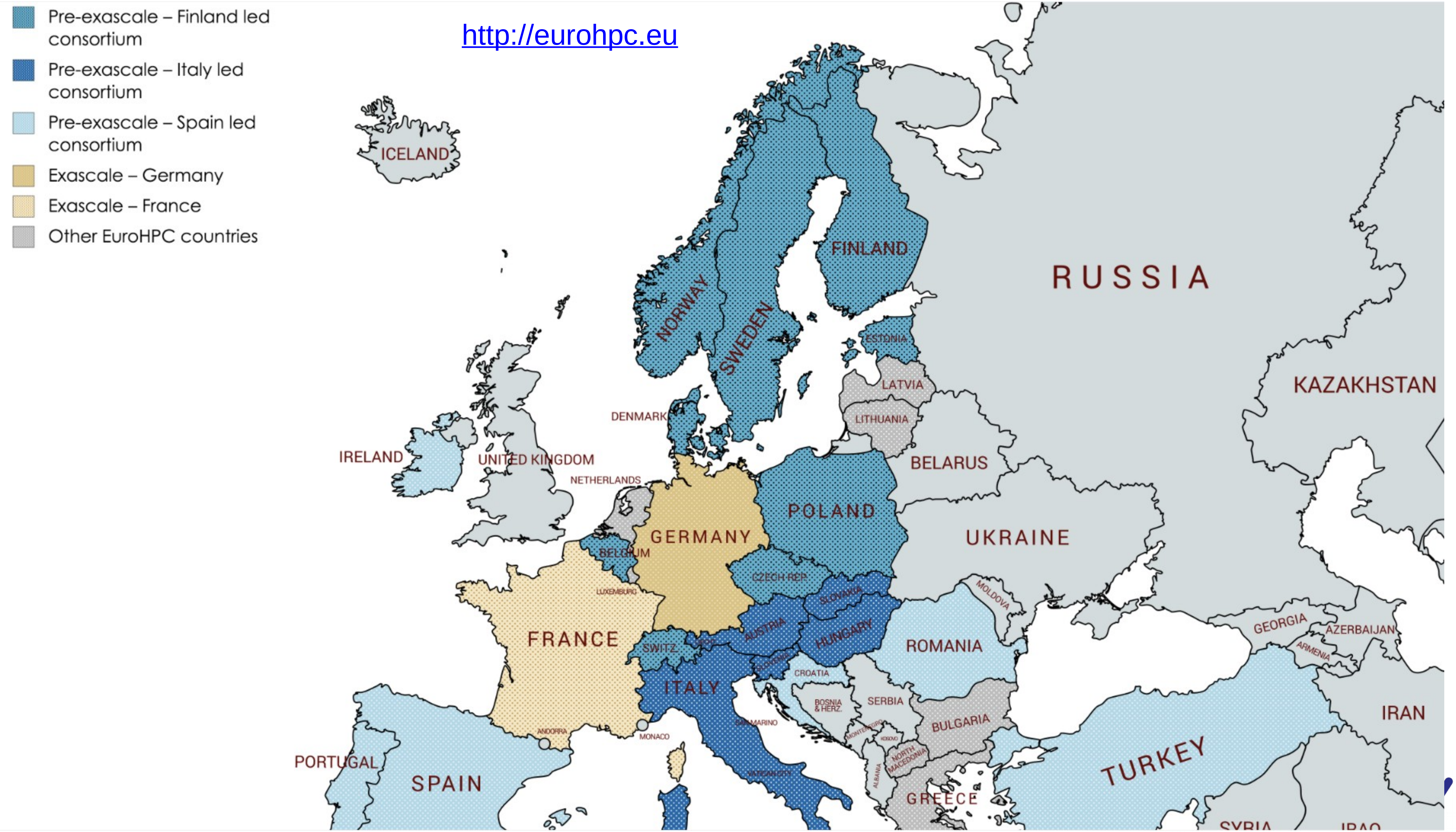




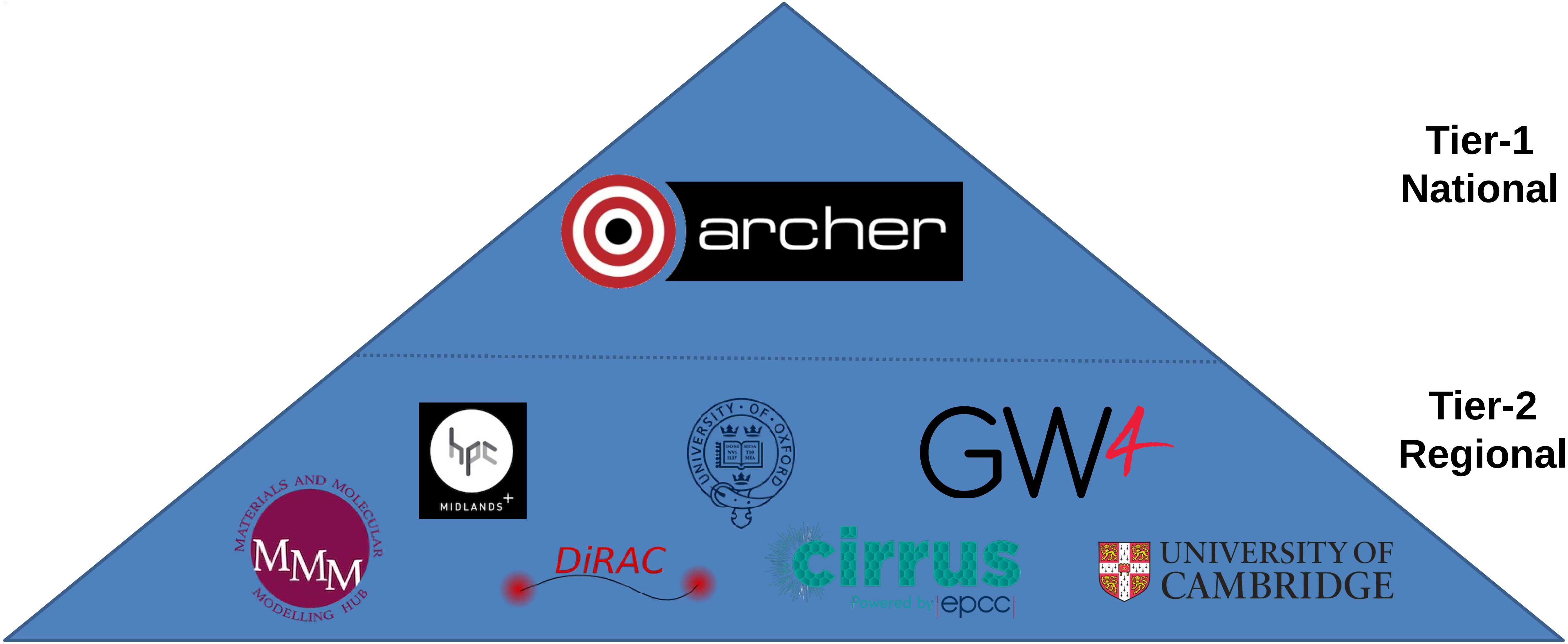
University of BRISTOL

The Next Platform, Jan 13th 2020: "HPC in 2020: compute engine diversity gets real"
https://www.nextplatform.com/2020/01/13/hpc-in-2020-compute-engine-diversity-gets-real/

http://eurohpc.eu



The UK's HPC ecosystem reflect this diversity



The UK's Tier-2 exploring options

Isambard

- **First production Arm-based HPC service**
- **10,752** Armv8 cores (168n x 2s x 32c)
  - **Marvell ThunderX2 32core 2.5GHz**
- Cray XC50 'Scout' form factor
- High-speed **Aries** interconnect
- Cray HPC optimised software stack
- **>420 registered users**, >100 of whom are from outside the consortium

University of BRISTOL

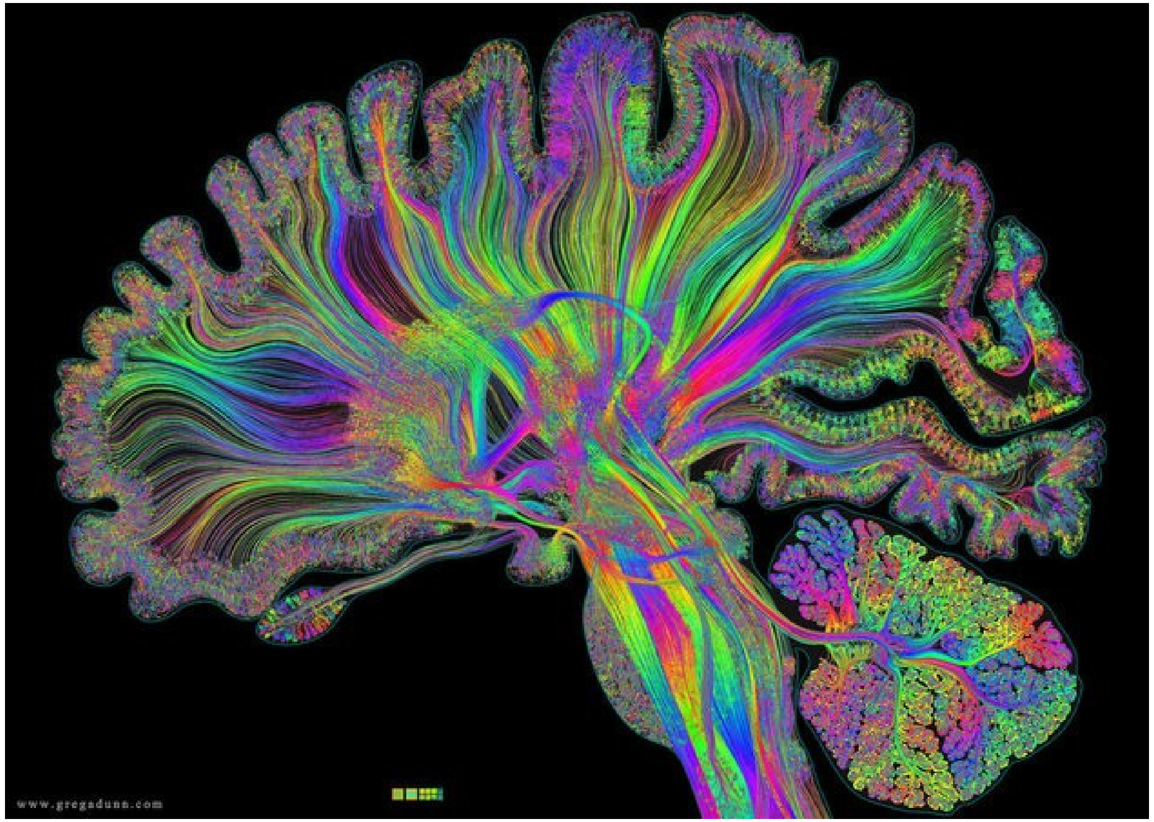
A performance analysis of the first generation of HPC-optimized Arm processors, S. McIntosh-Smith, J. Price, T. Deakin & A. Poenaru, CC:PE, Feb 2019. https://doi.org/10.1002/cpe.5110

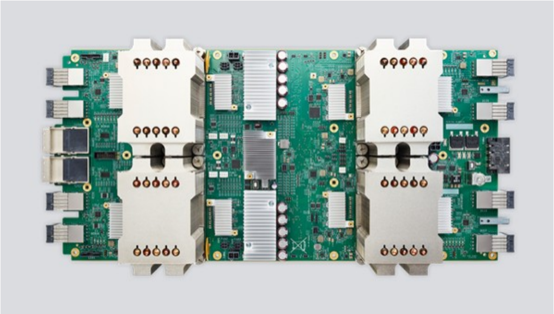
UK Tier-2 dense GPU systems

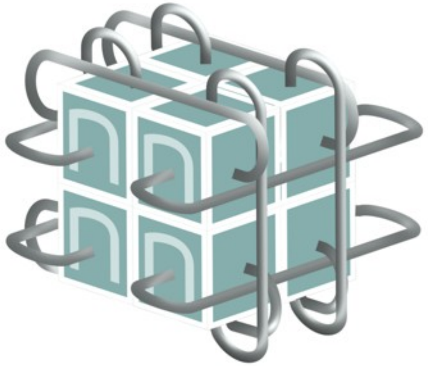
JADE
Tier 2 HPC

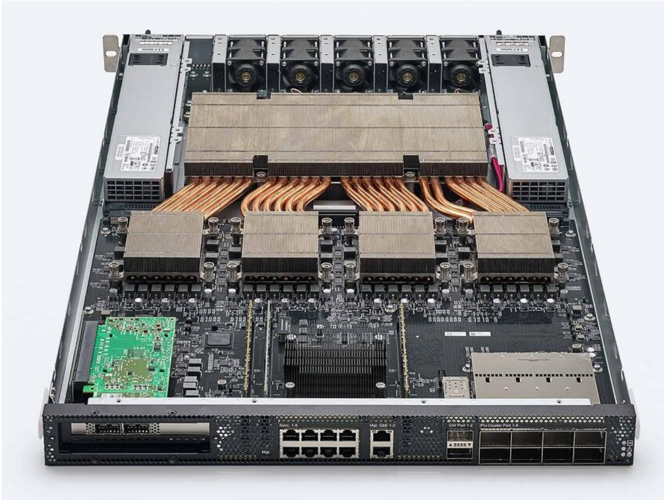
- 22 NVIDIA DGX-1 Deep Learning Systems, each comprising:
  - 8 NVIDIA Tesla V100 GPUs
  - NVIDIA's high-speed NVlink interconnect
  - 4 TB of SSD for machine learning datasets
- Over 1PB of Seagate ClusterStor storage
- Mellanox EDR networking
- Optimized versions of Caffe, TensorFlow, Theano and Torch etc

Emerging architectures for AI / Machine Learning



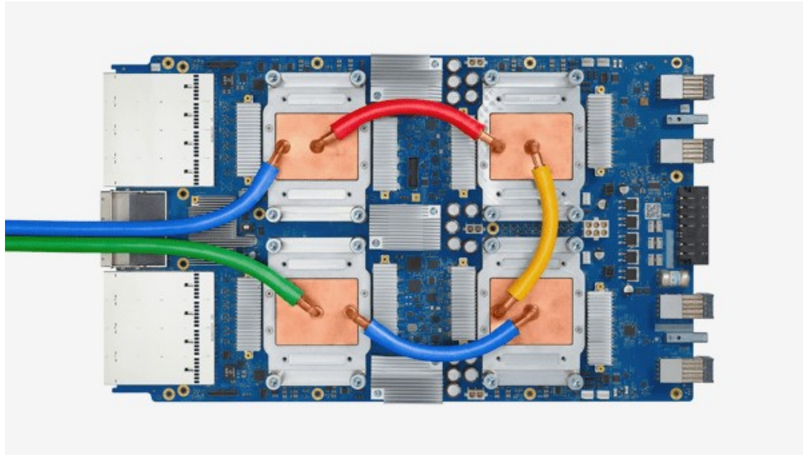






Google's Tensorflow Processing Unit (TPU), GraphCore, Intel's Nervana

Google's Tensor Processing Units:



Cloud TPU v3:
420 TFLOP/s
128 GB HBM
$2.40 / TPU hour

V4 supposedly improves performance by 2.7x

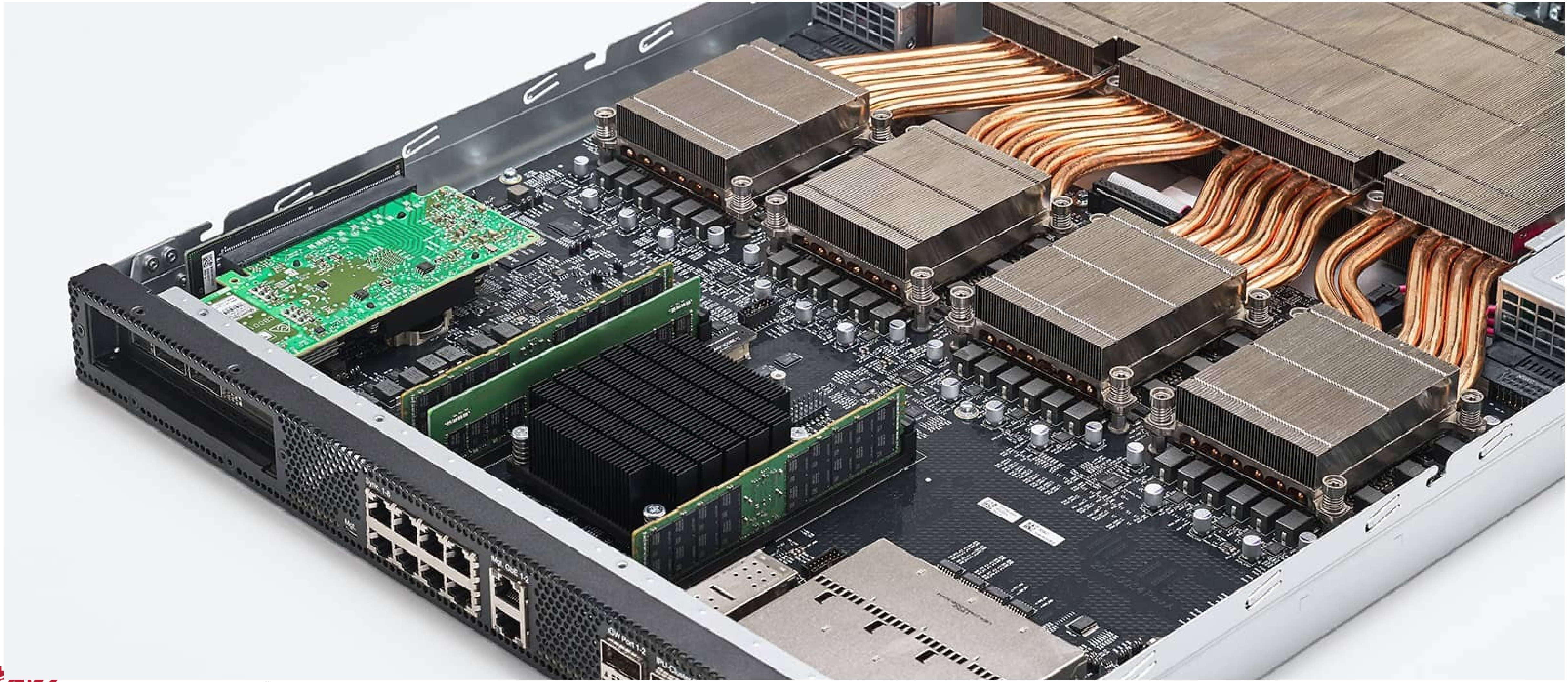
Cloud TPU v3 Pod:
100+ PFLOP/s
32 TB HBM
2-D toroidal mesh network

University of BRISTOL

https://cloud.google.com/tpu

ASIMOV

Graphcore has just announced their 2nd generation "IPU"



University of BRISTOL

ASIMOV

Graphcore IPU-M2000

- 4 x Colossus MK2 GC200 IPUs in a 1U box
- 1 PetaFLOP “AI compute” (16-bit FP)
- 5,888 processor cores, 35,328 independent threads
- Up to 450 GB of exchange memory (off-chip DRAM)

- 2nd gen IPU has 7-9X more performance on AI benchmarks
- 59.4B 7nm transistors in 823mm^2
- 900MB of on-chip fast SRAM per IPU (3x first gen.)
- 250 TFLOP/s AI compute per chip, 62.5 TFLOP/s single-precision

Massive scale AI/ML supercomputers

Three of the big issues facing parallel programming

1. **Massive parallelism**
   - Fugaku has over 7.63 **million** cores, each with 2x 512-bit wide vectors
2. **Heterogeneity**
   - CPUs, GPUs and more, from multiple vendors
     - Intel, AMD, NVIDIA, Fujitsu, Marvell, IBM, Amazon, …
   - Non traditional architectures
     - Graphcore IPUs, Google TPUs, vector engines, FPGAs, …
3. **Complex memory hierarchies**

The USA's ECP program

8 programming model and run-time projects funded in ECP:

- Two focus on MPI at Exascale (MPICH, OpenMPI)
- Two focus on task-level parallelism approaches (Legion, PaRSEC)
- One focuses on PGAS approaches (UPC++, GASNet)
- One focuses on parallel C++ (Kokkos, RAJA)
- Two focus on low-level on-node parallelism (ARGO, SICM)

Source: https://www.exascaleproject.org/research-group/programming-models-runtimes/

Do's and don'ts

Do:

- **Expose maximum parallelism** at all levels within your codes
  - Data, loop, thread, task, core, socket, node, system…
- Plan for the long term
  - FLOPS becoming free, data movement and storage increasingly expensive
  - O(10^9) parallelism required at Exascale (or more!)
- Use **standard** parallel programming languages and frameworks
  - **MPI, SYCL/oneAPI** and **OpenMP** are the main candidates, possibly **Julia**
  - **Domain Specific Languages (DSLs)** are a good way of isolating the science from the implementation

Don't:

- Get hooked on vendor-proprietary programming languages
  - NVIDIA's CUDA and OpenACC are the crack cocaine of HPC, **avoid!**

Key takeaways for scientific software developers

- **Orders of magnitude more parallelism** at Exascale, ≥ $O(10^9)$
- **Increased heterogeneity** (CPU+X, AI-optimized processors etc.)
- **MPI+X and DSLs likely to remain the most widespread approaches**
- **If starting from scratch**, worth evaluating some of the alternatives
  - Julia, parallel task frameworks etc.

These are some of the most exciting times to be developing scientific software!

For more information

Bristol HPC group: https://uob-hpc.github.io/

Email: **S.McIntosh-Smith@bristol.ac.uk**

Twitter: **@simonmcs**