Evaluation of the NVIDIA Grace Superchip in the HPE/Cray XD Isambard 3 supercomputer

Supercomputer for Scientific Applications

Thomas Green
University of Bristol, UK
thomas.green@bristol.ac.uk

Will Wishart
University of Bristol, UK
will.wishart@bristol.ac.uk

Simon McIntosh-Smith
University of Bristol, UK
s.mcintosh-smith@bristol.ac.uk

Richard J. J. Gilham
University of Bristol, UK
richard.gilham@bristol.ac.uk

Sadaf R. Alam
University of Bristol, UK
sadaf.alam@bristol.ac.uk

ABSTRACT

The Bristol Centre for Supercomputing (BriCS) has recently deployed 55,296 Arm Neoverse V2 CPU cores in a supercomputing platform, via 384-nodes of NVIDIA Grace CPU Superchips with LPDDR5 memory as part of the Isambard 3 HPC service for the UK HPC research community. Isambard 3 is an HPE/Cray XD series system using the Slingshot 11 interconnect. As one of the first systems of this kind, this manuscript overviews details of the hardware and software configuration and presents early performance evaluation and benchmarking results using a representative subset of scientific applications. The focus is to evaluate Isambard 3 as a "plug-and-play" environment for researchers, especially who are familiar with Cray software environment. We include microbenchmark results to provide insights into the performance behaviour of this unique architecture. We present a small scale scaling comparison between the NVIDIA Grace CPU Superchip with other mainstream CPUs, including Intel Sapphire Rapids and AMD Genoa and Bergamo. We report on issues during the attempts to use several major software toolchains available for Arm, such as the HPE Cray Compiler Environment (CCE), the Arm Compiler for Linux, and the NVIDIA Compiler, therefore focussing on GCC. Our findings include key opportunities for improvements that were discovered during our benchmarking, evaluation and regression testing on the system as we transitioned the service into operations from January 2025.

CCS CONCEPTS

• Architectures • Modeling and Simulation • Hardware validation

In Proceedings of CUG25. ACM, New York, NY, USA

KEYWORDS

Performance, Supercomputing, Interconnect

1 Introduction

Arm has come a long way since its initial offering as an energy efficient processor widely used in mobile phones, and in 2018 saw the first production Arm-based supercomputer with the installation of Isambard 1 [1]. Many different vendors now support Arm in different processors and form factors, but importantly, strict instruction set architecture compatibility across Arm-based CPUs, together with a well-defined common set of extensions, allow features to be added to encourage competition and innovation. Then in 2020 Fujitsu become #1 on the HPL-based Top500 with Fugaku, establishing placed Arm as a serious contender in the HPC arena.

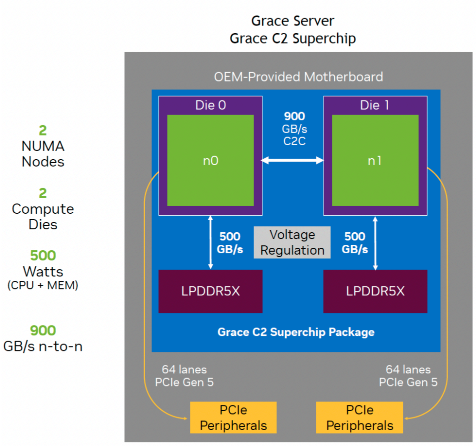

Figure 1 : Nvidia Grace CPU Superchip (source NVIDIA)

Isambard 3 is the latest iteration of the Isambard service that has continued the successful series of Arm-based supercomputers in the UK. Provided by HPE/Cray via the air-cooled XD224 series of servers, and built around the NVIDIA Grace CPU

Superchip (Grace Superchip, see Figure 1). It provides a significant step up in performance from the previous generation based on Marvell's ThunderX2 processor. Since January 2025, Isambard 3 provides UK researchers an alternative choice of architecture and is available to all UK researchers via the United Kingdom Research and Innovation (UKRI) access calls that give researchers the opportunity to apply for time on the service. UKRI covers all research councils, a big difference from the previous Isambard systems that were limited to the UK's Engineering and Physical Sciences Research Council (EPSRC). Isambard 3 is hosted at the University of Bristol as part of the GW4 consortium of universities (University of Bristol, University of Bath, Cardiff University, and University of Exeter). Each University has a dedicated allocation for their researchers and provides extra capacity to the region's HPC capability. The attraction to the Grace Superchip was due to the energy efficiency and high memory bandwidth compared to other CPU offerings on the market, a similar reasoning to the previous iterations of Isambard. Each Grace Superchip contains 144 high-performance cores. The 384 compute nodes are connected with HPE's Slingshot 11 interconnect at 200Gbps, and in total provides 55,296 cores.

Previous submissions to CUG provided an evaluation of earlier generations of the Isambard service, with single node benchmarks in 2018 [1] and further system-wide evaluation in 2019 [2]. This evaluation focusses on the "plug and play" environment and whether users can quickly be up and running with their own codes using Spack. Isambard 3 hosts other small scale partitions via its Multi-Architecture Comparison System (MACS), connected to the same HPE Slingshot interconnect and provides opportunity to check consistency of the build configuration across architectures. It should be noted, the MACS small scale nature, mostly 2 nodes in each partition, and configurations that focus on heterogeneity, e.g. different processor and memory configuration, and place limits on the comparisons. However, Isambard 3 and MACS containing the similar OS, software, filesystems and interconnect, it is unique opportunity to compare node level hardware differences.

2 System architecture

Isambard 3 is hosted in a recently built, air-cooled Modular Datacentre (MDC) in Bristol. This same MDC is also the location for Isambard-AI phase 1 [21], for which direct liquid cooling capability was added to the MDC's otherwise air cooled design.

Isambard 3 consists of 6 racks of 64 XD224 servers each, for a total of 384 nodes containing the Grace Superchip. A Grace Superchip contains two Grace CPUs, each with 72 Arm Neoverse V2 cores, providing 144 cores per node. Each core includes four 128-bit SVE2 execution units for SIMD execution. The Grace Superchip connects the two CPUs on the Superchip, which run at up to 3.4GHz. The C2C interconnect provides communication between all the cores within a superchip (this will be important later in the paper) and provides familiar unified memory across the superchip; see Figure 2 for measured timings for communication between cores, measured using the Github project nviennot/core-to-core.

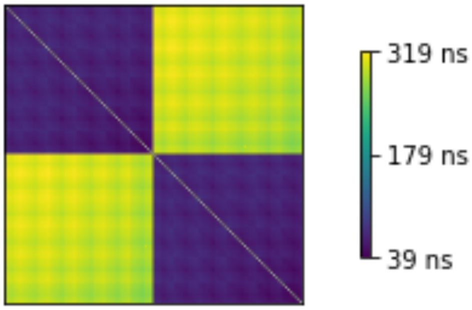

Figure 2 : Core-to-core latency on NVIDIA Grace Superchip, purple blocks are 72 cores from within the same Grace CPU, green blocks are 72 cores each from a different Grace CPU.

Each node's memory consists of 256GB of LP-DDR (240GB useable) running at 8532 MT/s. Each core has its own set of caches: L1I (64 KB), L1D (64 KB), L2 (1 MB). Each Grace CPU includes a shared L3 cache of 114 MB, giving the two socket Superchip a total of 228MB of L3. HPE's Slingshot network provides a 200Gbps high-speed interconnect, with each node containing a single Slingshot Cassini network card (NIC). Slingshot is configured with a Dragonfly topology to provide balanced performance across the cluster. Network storage is provided by a 2 PiB Cray ClusterStor E1000, providing a Lustre filesystem to all nodes. Each Grace node contains local storage with 2 x 1.92TB M.2 SSD to provide an option for faster local storage when required. The same XD224 servers are used for login and services nodes, providing a consistent architecture for the user build environment.

The work in this paper was performed in February 2025 following an upgrade to Slingshot to the main Isambard 3 cluster, but the MACS remained at earlier version. Firmware upgrades were applied in November 2024 during installation to the recommend version at that time. Initial benchmarking in October 2024 showed a 5% increase in memory bandwidth due to the updated firmware. HPCM 1.11 is used as the management software that supports the cluster, with an OS based on Suse Linux Enterprise Software (SLES) 15sp5, with Cray Operating System (COS) 3.1, Cray Programming Environment 24.07 and USS 1.1.0, along with the NVIDIA SDK 24.05 and Arm compilers 24.10.4. SLES15sp5 is based on Linux Kernel 5.14 which contained a particular severe bug (fixed in later kernel version) affecting the memory compaction behaviour on Arm which needed special attention. All the software supports the Grace Superchip with documentation suggesting it is still early support for the architecture. Fabric manager 2.2.0 and Slingshot Host Software (SHS) 11.1 provides the Slingshot environment on the main cluster, but the MACS remained on SHS 11.0. Lustre 2.15.1 is used for the network parallel filesystem. For the comparison on MACS due to using a mix of AMD and Intel based processors, GCC was used for consistency across the processor types. To perform consistent benchmarking, and to manage complex build dependencies, Reframe [3] at version 4.7.4 was used to run and collect benchmark conditions and performance data. This used Spack [4] at version 0.23 to provide the software environment for each test. Supporting users via documentation and configuration information allows a "Bring Your Own Software" model to be possible, compared to a traditional existing module environment. Combining Reframe and Spack, along with being the first system of its kind, will provide a suitable foundation to deliver health monitoring and suitable integration tests for users to take advantage of the latest recommended configuration for their software requirements.

3 Benchmarks

A number of different benchmarks are used and detailed in this section to provide confidence in the system and software has a robust configuration for users and developers of scientific applications rather than to obtain the best result in each case. This will enable researchers to reduce the time to science, or so-called "Out of the box" evaluation. The range of preliminary benchmarks will provide a basis for future performance evaluation and identify behaviour that may require further attention.

3.1 Micro-benchmarks

This section details the micro-benchmarks that provide an easier way to categorise performance compared to larger production codes that can be complex and hard to understand key performance characteristics. These benchmarks are commonly used to model behaviour across many architectures. Previous papers have also described these benchmarks but are included here for convenience.

STREAM [5]: The well-known benchmark commonly used for performance of simple vector kernels. This synthetic benchmark provides a measure of bandwidth in MB/s. The Triad kernel is a popular choice. For this study, in order to avoid measuring cache performance, an array size of 240 million double precision floating points is used. This is large enough to exceed the last level of caches on the Grace CPU. A key benefit of the Grace Superchip is its high memory bandwidth, so it is crucial to understand this behaviour.

Arm-kernels [6]: Developed by NVIDIA to provide an artificial method to measure the performance of Arm CPUs. Written in assembly, the executables are split across all the common instructions such as scaler, NEON and SVE functionality. It provides a measure in operations per second. This synthetic benchmark measures performance at the node level and the corresponding behaviour for each instruction confirms compiler behaviour. The use of clobber lists in the code added minor issue restricting compiler support. Clobber lists prevent the compiler from using specific registers, with GCC providing complete support for all registers listed.

CloverLeaf [7]: A micro-benchmark or mini-app from the Mantevo suite [18], CloverLeaf solves Euler's equations of compressible fluid dynamics on a cartesian grid using an explicit, second order method. This is commonly used to investigate portability across architectures and as an example of a stencil code it is known to be memory bandwidth bound. For this study we used test problem 5 (clover_bm_16 input deck) which uses as grid of 3840 * 3840 cells. We found this provided a good balance of scaling and time to complete benchmark.

TeaLeaf [8]: Referred to as a mini-app and another member of the Mantevo suite [18], TeaLeaf solves the linear heat conduction equation on a spatially decomposed regular grid using a five point stencil with implicit solvers. A choice of solvers are available and we use the conjugate gradient method with the tea_bm_5.in input file. TeaLeaf is known to be memory bandwidth bound, but at scale across many nodes, it becomes communication bound. Our test case has 4000 * 4000 cells in the grid and runs for 10 timesteps.

SNAP [9]: The SNAP mini-app solves the linear radiation pseudo-transport problem on a structured mesh. It has a large memory requirement and a simple kernel that depends on conditions increasing communication costs. It supports MPI for spatial decomposition and OpenMP within the energy domain. In this case we have used weak scaling with 1024 × 12 × 12 cells per MPI rank, and up to 32 threads for OpenMP (due to the 32 energy groups).

Neutral [10]: Neutral is a Monte Carlo neutron transport mini-app. It uses a mesh-based approach along with the traditional Monte Carlo technique. Neutral adds the extra requirement of memory accesses for mesh and particle data on top of the well understood Monte Carlo algorithm. This benchmark was built with OpenMP and therefore limited to a single node.

OSU Micro-Benchmarks [11]: This is a popular suite of tools to explore interconnect performance, with tests for latency and bandwidth across different message sizes. We used osu_latency and osu_bandwidth to capture the performance behaviour. The cray-mpich library was used with GCC, and the Grace Superchip and Slingshot 11 being of interest due to its recent introduction.

3.1 Scientific applications

As previously stated, Isambard 3 will be used mainly by UK researchers sitting alongside other national research compute services such as the UK's national supercomputer, ARCHER2. Common codes are predominantly from the physical science space, and can scale to many hundreds of nodes if needed. For our benchmarking and because Isambard 3 is already in production, we limited our scaling to a maximum of 16 nodes, or 2,304 cores. These codes are all heavily used on ARCHER2 and are a good subset of the real applications researchers will be using on Isambard 3 and most featured in recent project proposals to access the service in "Access to HPC - autumn 2024" UKRI call.

CASTEP [12]: CASTEP is a popular code used to calculate material properties from first principles. Currently free for academic use across the world, the behaviour of the code on various systems allows researchers to choose the most appropriate system to run their cases. The cases used in this paper are based on the common benchmarks provided by CASTEP. Al3x3 is small enough to run across many system configurations but large enough to make it "real-world". The al3x3 benchmark is essentially a 3x3 surface cell of the al1x1 system and consists of 270 atoms (108 Al, 162 O) or 1296 electrons. A second test used the crambin case, which is named after the protein residue consisting of 3660 electrons. This is therefore a larger simulation and places more stress onto the compute and interconnect. The common, but larger, DNA case was found to require too much memory for the node counts being used in this paper.

CP2K [13]: The CP2K code simulates the Ab-initio electronic structure and molecular dynamics of different systems such as molecular, solids, liquids, etc. Fast Fourier Transforms (FFTs) form part of the solution step, but it is not straightforward to attribute these as the performance-limiting factor of this code. The memory bandwidth of the processor and the core count both have an impact. We have used the H2O-64 benchmark, which simulates 64 water molecules, for a total of 192 atoms and 512 electrons. This is an often-studied benchmark for CP2K, and therefore provides sufficient information to explore the performance across the different architectures in this paper.

GROMACS [14]: A molecular dynamics package that solves Newton's equations of motion. Used extensively across many disciplines and therefore a very popular package. Systems of interest, such as proteins, can contain millions of particles. At low node counts it is considered to be FLOP/s-bound, while it becomes communication bound at higher node counts. The behaviour at low node counts resulted in developers using compiler intrinsics to ensure an optimal sequence of these operations, therefore compiler support for these features is critical. For supported intrinsics, operations are packed so that they saturate the native vector length of the platform, e.g. 256 bits for AVX2, 512 bits for AVX-512. For this study, we used the common TestCaseA and TestCaseB benchmarks. TestCaseA is a membrane protein GluCl containing 142,000 atoms, while TestCaseB is a cellulose and lignocellulosic biomass containing 3.3 million atoms. For the Grace Superchip, we used the 128-bit ARM_SVE vector implementation, which is the closest match for the underlying Armv9 architecture. We note that, within GROMACS, the Arm vectorisation implementation is not as mature as the SIMD implementations targeting x86. For this study we run one MPI rank of one OpenMP thread per core.

NAMD [15]: A molecular dynamics simulation program designed to scale up to millions of atoms. Frequently run at high node counts, it is based on a high-level library that abstracts the mapping of processors to work items. The test cases are the ApoA1 and STMV benchmarks and both are common set of inputs for measuring scaling capabilities. STMV simulates one of the smallest viruses in existence at around a million atoms, whilst ApoA1 simulates the protein of the same name at 92,224 atoms. These benchmarks include PME calculations, which use FFTs, and so its performance is heavily influenced by that of the FFT library used. Due to the complex structure of atomic simulation computation and the reliance of distributed FFTs, NAMD's performance has many factors influencing its behaviour.

OpenFOAM [16]: Originally developed as an alternative to early simulation engines written in Fortran, OpenFOAM is a modular C++ framework aiming to simplify writing custom computational fluid dynamics (CFD) solvers. We use the simpleFoam solver for incompressible, turbulent flow from version 2312 of OpenFOAM, a recent supported release in Spack v0.23 at the time we began benchmarking. The input case is based on the HPC Motorbike model[17], which is a representative case of real aerodynamics simulation and thus should provide meaningful insight of the performance. The decomposed grid consists of approximately 34 million cells. OpenFOAM is memory bandwidth–bound, at least at low node counts.

4 Analysis and evaluation

The NVIDIA Grace system along with the comparison systems from the MACS is found in Table 1. The main advantage of the x86 architecture is that is has had many years as the dominant market leader, and therefore software has had much more time to be optimised to make the most of the hardware. Modern x86 processors have at least AVX2 (Milan), all the way up to AVX-512 in the most recent AMD and Intel processors. The range of SKUs available from AMD and Intel make the comparison difficult, but we are using familiar specifications in order to make the comparison as relatable as possible. The Grace Superchip benefits from SVE2 which is provided by processing 4 x 128 bit wide vector instructions. Each Isambard 3 XD224 server contains a Grace Superchip at a base clock speed of 3.1 GHz along with 256GB of LPDDR5X. LPDDR5X is different to many other systems and provides a unique selling point of the Grace Superchip, providing both energy efficiency and high bandwidth. In most modern CPUs, cores are allowed to fluctuate in frequency depending on many factors such as power draw and temperature etc. Thus it is becoming hard to know the exact speed of the processor during a run, and so we provide the guaranteed base speed as comparison.

Processor	Cores	Base Clock Speed [GHz]	FP64 peak [TFLOP/s]	Default TDP [W]	Bandwidth [GB/s]
NVIDIA Grace CPU Superchip	2 x 72	3.1	7.1	1 x 500 (including memory)	1024.0
AMD EPYC 7713 (Milan)	2 x 64	2.0	4.0	2 x 225	409.6
AMD EPYC 9354 (Genoa)	2 x 32	3.25	3.3	2 x 280	921.6
AMD EPYC 9754 (Bergamo)	1 x 128	2.25	4.6	1 x 360	460.8
Intel Xeon Gold 6430 (Sapphire Rapids)	2 x 32	2.1	4.3	2 x 270	614.4
Intel Xeon CPU Max 9462 (Sapphire Rapids)	2 x 32	2.7	5.5	2 x 350	3276.8

Table 1 : Specifications of systems in the test, Grace has 384 nodes, Milan has 12 nodes (upto 4 used), others have 2 nodes

The MACS x86 nodes are configured with simultaneous multi-threading (SMT) enabled. The same feature is not available on the Grace Superchip (previous Isambard with ThunderX2 had 4-way SMT available). On MACS this a factor that can influence software behaviour, when measuring CPU cache bandwidth at the L1 and L2 level (L3 tends to be shared across cores). With SMT, each core is divided up to provide dedicated cache for each thread and therefore the cache available for each thread is smaller. Using the tool cachebw from Github project UoB-HPC/cachebw to estimate cache bandwidth on each CPU, we can clearly show the difference with or without hyperthreading enabled. For comparison, the bandwidth available in the caches had SMT disabled for x86. These results are listed in Table 2 and clearly show the Grace Superchip having high bandwidth throughout the memory hierarchy.

Processor	L1 [kB]	BW [GB/s]	L2 [kB]	BW [GB/s]	L3 [kB]	BW [GB/s]
NVIDIA Grace CPU Superchip (grace)	9,216	15,902	147,456	13,017	233,472	7,568
AMD EPYC 7713 (milan)	4,096	6,054	65,536	5,246	524,288	2,456
AMD EPYC 9354 (genoa)	4,096	17,872	131,072	7,052	262,144	4,778
AMD EPYC 9754 (Bergamo)	2,048	11,303	65,536	6,885	524,288	2,893
Intel Xeon Gold 6430 (spr)	3,072	9,108	131,072	1,146	122,880	688
Intel Xeon CPU Max 9462 (sprhbm)	3,072	12,991	131,072	2,061	153,600	1,544

Table 2 : Cache bandwidth as measured by the cachebw tool.

4.1 Methodology

To allow the rigorous recording of activities across the many different processor types and configurations in this work, we used Reframe [3]. Reframe is a popular framework to create benchmarks and regression tests for use on HPC systems. Many other examples of using Reframe can be found in the UK, including the ExCALIBUR [19] project funded by UKRI. Reframe includes a selection of example benchmarks and configurations. For this work, a set specifically created benchmarks are used, and our intention is to propose committing these tests to projects such as ExCALIBUR. Reframe can make use of pre-installed software, or it can use software provisioning tools such as Spack, as we use in this work. The Reframe configuration we used is openly available online at: https://github.com/isambard-sc/buildit

Spack is another common tool we are recommending to users on Isambard 3 as a method to build their own software. As well as being the tool used by Reframe, this was a good test of using Spack and its configuration on a new Arm platform. A particular point should be made regarding Spack and Cray Compilers. Spack uses a library called archspec. During this work the use of the Cray compiler directly (craycc) did not set the expected default compiler flags, which caused a knock-on problem since craycc would try and compile for the generic architecture instead of the specific architecture. Therefore, at this stage we decided it would be fairest to focus on just the GCC compiler results on all systems. Other compilers experienced issues at build time when building libraries required to run the benchmarks with Spack not handling a mix of compilers well. GCC experienced fewer issues and generally worked well without requiring configuration issues, so it is ideal for this initial evaluation of the system.

The configuration for Reframe using Spack can be found online as stated earlier, and is driven by a configuration for the clusters, along with a configuration for each application and benchmark case. To reduce the number of builds, each test that shares a similar build inherits a common environment that was created for the application; Reframe calls these fixtures and has allowed easy correction of issues identified during this work. This paper will directly benefit the users of the system as configurations are tested, assessed and made available online.

The scaling results in this paper are largely restricted due to the size of jobs that most researchers will likely be using on Isambard 3 estimated at 16 nodes, and the MACS cluster has a maximum of 2 nodes per partition (4 available for Milan), there was no benefit to go larger. When OpenMP is available the default is to use OMP_PLACES set to “cores” and pack with OMP_PROC_BIND set to “close”, but this can be parameterised when needed, such as is required for the STREAM benchmark.

4.2 Micro-benchmarks

This section will present results from each of the micro-benchmarks described earlier.

Arm-kernels: the performance of the instructions seems to be in the expected ballpark; see Figure 3. The SVE instruction that achieved the highest operation rate of 52.766 GOps/s, when multiplied by 144, the number of cores, gives a peak speed of 7.6 TFLOP/s double precision.

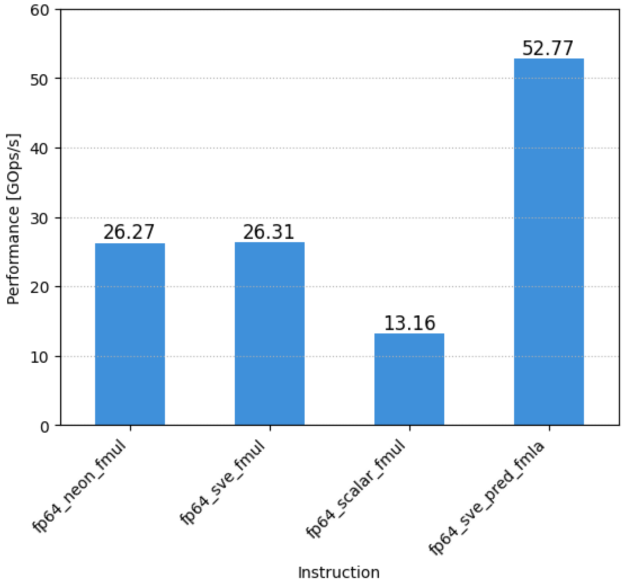

Figure 3 : Comparison of instruction performance on a single core on a Grace Superchip.

STREAM: as shown in Figure 4, the Grace Superchip clearly shows the benefit of its memory design, achieving almost 1TB/s of bandwidth. Only the Sapphire Rapids (SPR) HBM variant could keep up with it, with a performance supported by other evaluations [20], with the other processors in the test trailing behind. With SMT enabled on the MACS we may be impacted by particular OS configurations. The results for many of the older AMD processors required underpopulating the thread counts, which suggests cache contention at higher thread counts. The socket-to-socket variability for Grace was found to follow normal distribution, with standard deviation of ~ 1% of the mean.

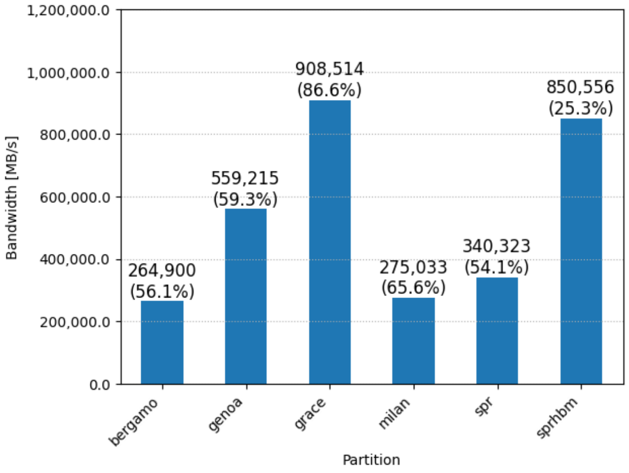

Figure 4 : Stream results using all cores on each system (percentage of peak in brackets).

CloverLeaf: being memory bandwidth bound on a single node, the Grace Superchip and Sapphire Rapids HBM processor are the better performing configurations in Figure 5, with Grace showing scaling to higher core counts providing confidence the system interconnect is working well.

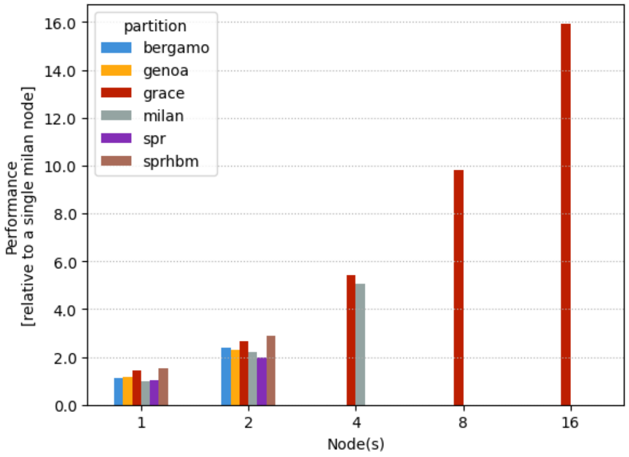

Figure 5 : Cloverleaf performance for each system relative to a single Milan node.

Tealeaf: a memory bandwidth bound code, the Grace Superchip and SPR HBM processors are surprisingly not the highest performing. The super linear scaling of this test is due to the size of the problem being able to exploit large last level caches more. A higher number of x86 nodes would show whether MPI communication could be a difference between the architectures as found with ThunderX2 and its MPI_Allreduce behaviour.

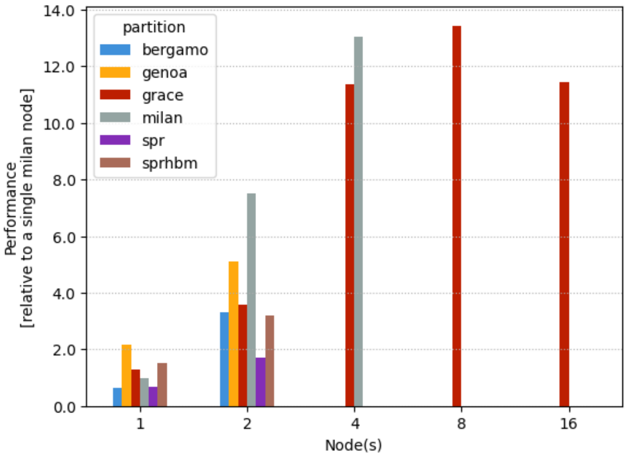

Figure 6 : Tealeaf performance for each system relative to a single Milan node.

SNAP: the weak scaling design of this problem means we would expect to see similar performance across every node count configuration in Figure 7. Higher node counts should eventually show SNAP becoming network bound. Most of the partitions used show this behaviour except for Milan and SPR HBM, where a reduction in performance is seen even on modest node counts. Further investigation will be needed to explain this behaviour since SMT effects would likely be seen across all x86 partitions.

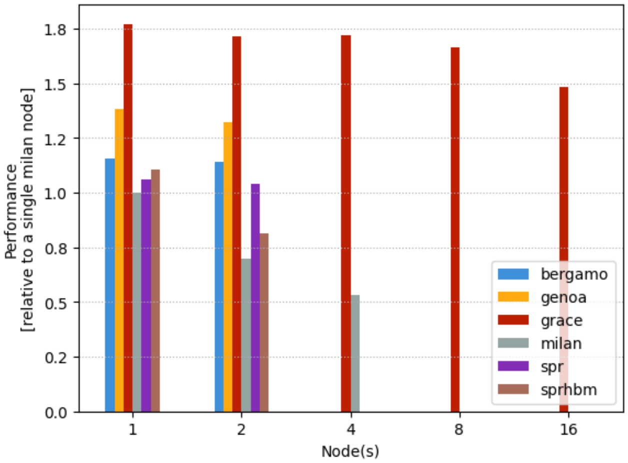

Figure 7 : SNAP performance for each system relative to a single Milan node.

Neutral: this benchmark's random access behaviour within memory results in low cache reuse, as reported in earlier studies, and therefore Grace's high memory bandwidth may not be an advantage for this code. Figure 8 clearly shows the recent Bergamo partition has an advantage, but the Grace Superchip still performs well especially, compared to SPR HBM.

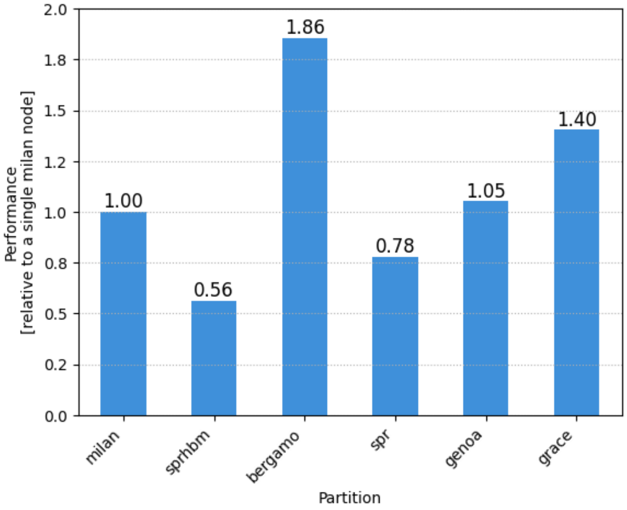

Figure 8 : Neutral performance for each system relative to a single Milan node.

OSU Micro-benchmarks: here we generate a familiar plot to describe network performance. The lowest latency in Figure 9 is found to be just under 2us when run on the core closest to the network card (core 72 on the second CPU). When run on the first CPU of the Grace Superchip, latency is higher. Our current suspicion is that the route across the C2C on each node is adding an extra latency of almost 1 us. If no specific CPU binding is given to Slurm, the default behaviour when the node is exclusive indicates the MPI task is placed near to the network card.

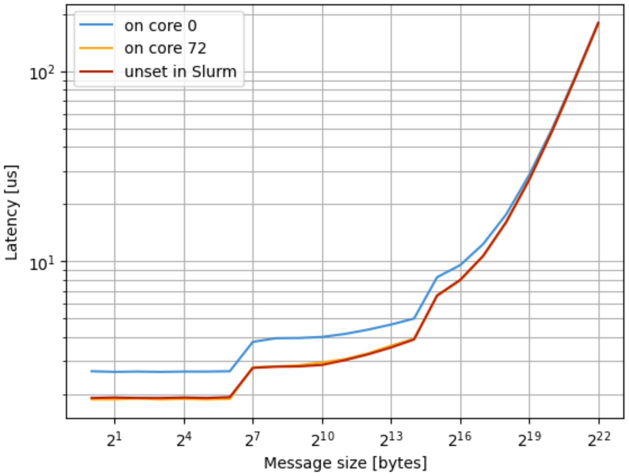

Figure 9 : MPI Latency between two Grace Superchip nodes. Note locality difference, possibly due to C2C effects.

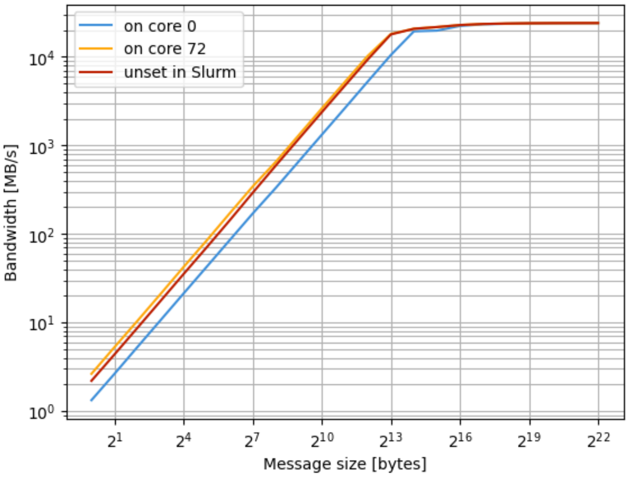

Figure 10 : MPI Bandwidth between two Grace Superchip nodes. Note locality difference, possibly due to C2C effects.

The MPI bandwidth as seen in Figure 10 reflects the previous plot and the location effect of where the MPI tasks run on the Grace Superchip is apparent, with the first core (core 0) showing lower bandwidth. Further investigations have shown a complex relationship between MPI message size and journey through the slingshot network. These will be reported in a future communication.

4.3 Scientific applications

This section will present results from running the scientific applications as listed earlier that will directly benefit users of Isambard 3 and provide guidance for availability and performance.

CASTEP: it is expected that CASTEP is memory bound and indeed the Grace Superchip performs well. The expected limitations for the crambin case for a single node job resulted in not running on a single node for both cases due to memory constraints. The use of threading might be an alternative way of exploring behaviour for future work. The larger crambin model shows good performance in Figure 11, with the Grace Superchip showing good scalability. Bergamo (192 GB) and SPR HBM (120 GB) had out of memory errors even at 2 nodes due to their lower available memory, therefore no results are presented for those partitions.

Being a smaller case than crambin, the al3x3 results in Figure 12 show signs of reduced benefit at higher node counts. All partitions performed well compared to Milan, with the Grace Superchip and SPR HBM performing well, as expected, due to the memory bandwidth bound nature of the code.

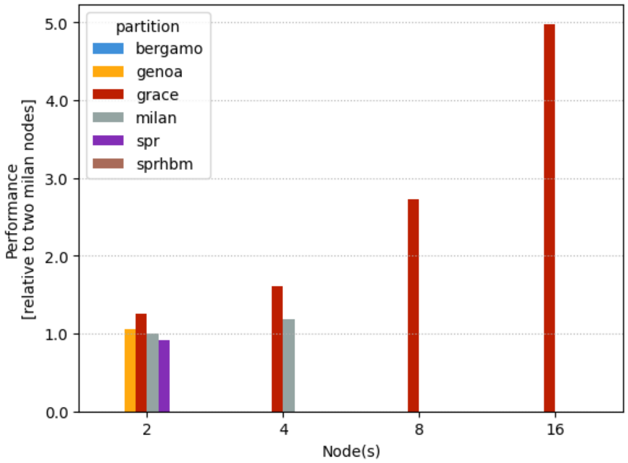

Figure 11 : CASTEP case crambin across different node counts Bergamo and Sapphire Rapids HBM had memory exhaustion at 2 nodes.

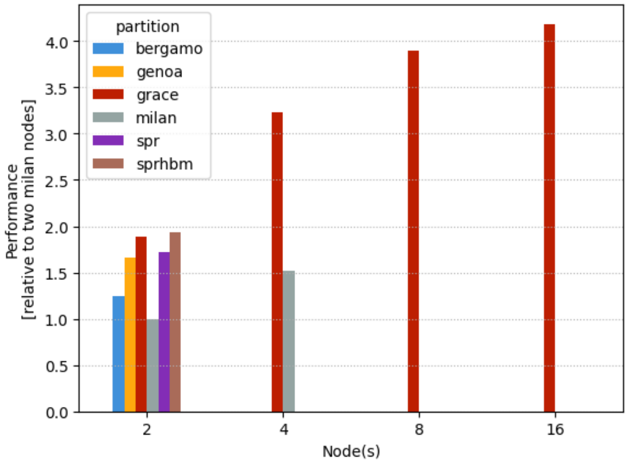

Figure 12 : CASTEP case al3x3 across different node counts.

CP2K: the H2O-64 case plotted in Figure 13 shows its clear limitation, with the Grace Superchip partition showing scaling issues. All other partitions show a reduction in performance compared to a single node. CP2K has many performance characteristics which deserves a different approach for future studies.

GROMACS: a combination of high FLOP/s and high speed caches shows the Grace Superchip performs well on a single node for TestCaseA (only Bergamo performed better than the Grace Superchip). TestCaseA being a smaller model reaches a limitation to run at higher node counts, as shown in Figure 14.

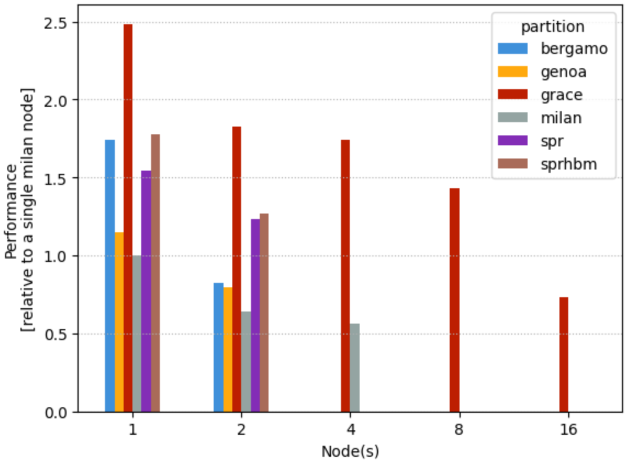

Figure 13 : CP2K H2O-64 case across different node counts

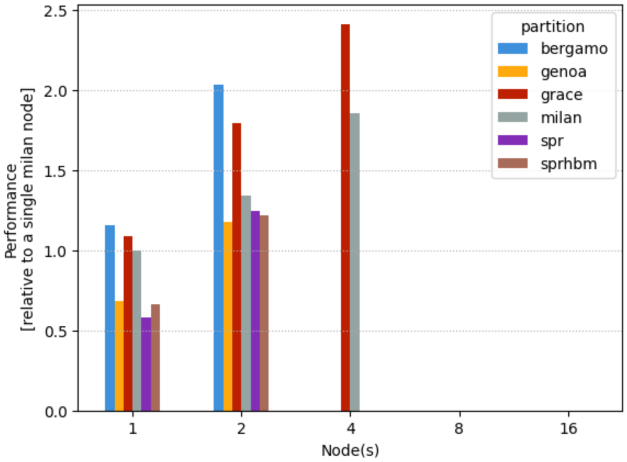

Figure 14 : GROMACS TestCaseA across different node counts

Figure 15 shows performance for TestCaseB and being a larger case allowed the Grace Superchip to run to higher node counts. Possibly due to relatively new support in GROMACS, the Grace Superchip struggles to clearly take a lead. It would be good to see further improvements as development of Arm support in GROMACS continues.

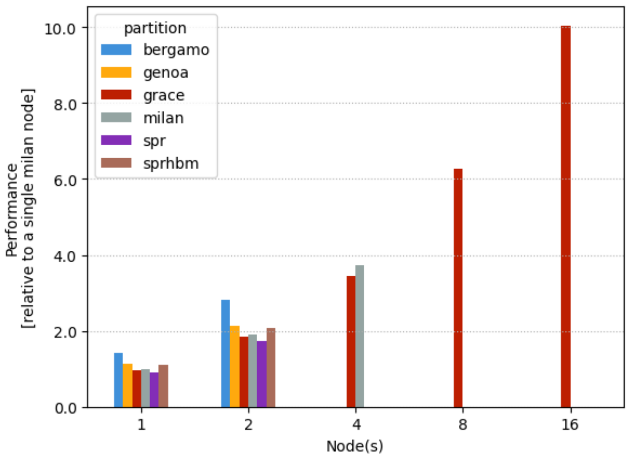

Figure 15 : GROMACS TestCaseB across different node counts

NAMD: having needed a patch to the code to get support for the Slingshot 11 interconnect in Charm++, the Grace Superchip seems to perform very well compared to other sytems. NAMD has a complex set of configuration choices and may benefit with tuning of Slurm parameters. The default option in Spack for Charm++ makes use of shared memory with worker and communication threads, and were set to 1:1 in this study to minimise complexity. On processors with AVX-512, an attempt to use the AVX Tiles option was found to generate a segmentation fault and requires further debugging. The Grace Superchip scales well until the small case size in Figure 16 leads to a sudden drop in performance as is struggles to perform work on the node and communication takes over.

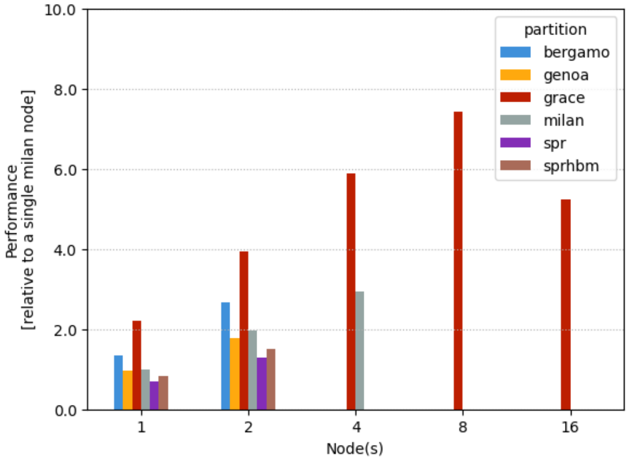

Figure 16 : NAMD case apoa1 across different node counts.

Figure 17 shows similar behaviour to the previous plot. NAMD not being bounded by a particular aspect could be due to the balanced memory bandwidth and FLOP/s of the Grace Superchip, making this particular software well suited to the architecture. More investigation into the software configuration is required to fully understand what is going on here.

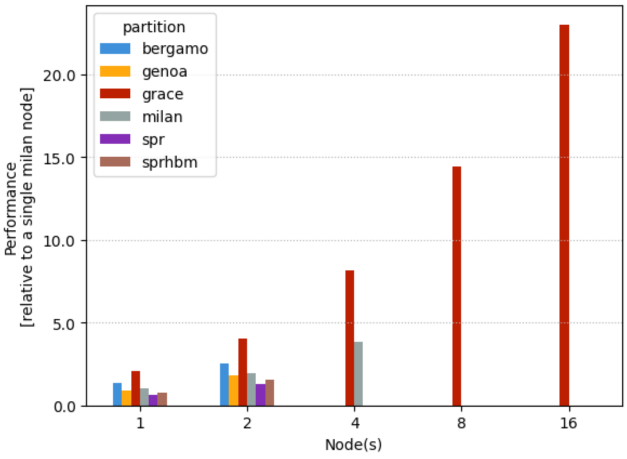

Figure 17 : NAMD case stmv across different node counts.

OpenFOAM: we expected the Grace Superchip to perform well on OpenFOAM due to the memory bandwidth bound nature of the code. Grace clearly shines, along with SPR HBM. Being over 2x faster than Milan, Grace is a good choice for OpenFOAM users.

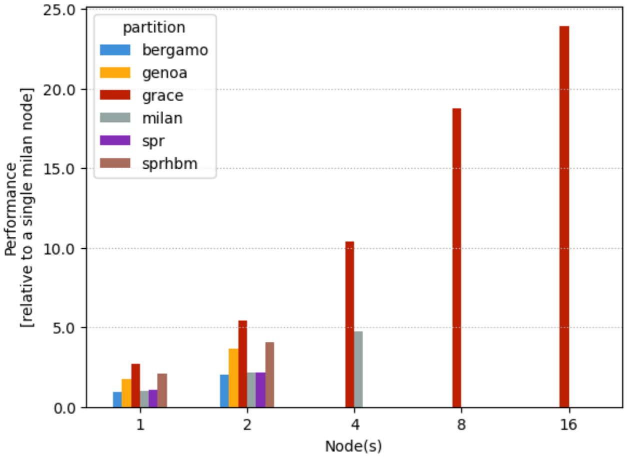

Figure 18 : OpenFOAM HPC motorbike case across different numbers of processors

5 Conclusions

This paper had a number of aims. First the evaluation of the NVIDIA Grace CPU Superchip, second the system design of Isambard 3, including Slingshot 11, and finally the suitability of Spack to build well performing software on new Arm platforms.

The results demonstrate that the NVIDIA Grace CPU Superchip is a well performing CPU, a "workhorse" with no significant limitation being displayed, and with performance competitive with state-of-the-art offerings from the traditional x86 vendors. The Grace Superchip is attractive for applications such as OpenFOAM, due to its greater memory bandwidth benefiting communication performance. The scalability of the Slingshot 11 interconnect of the larger node counts found on the Grace Superchip partition on Isambard 3 displays a few early concerns; the latency effect observed in the OSU benchmarks should be carefully investigated for application behaviour. The use of Spack within Reframe, that shares documented configuration for users of Isambard 3, worked well for GCC. For other compilers requiring specific configurations, it became harder to find the combination that worked together, frequently finding GCC being required for dependencies. There will also be cases where applications package only for x86 and would therefore require additional porting efforts for Arm platforms.

Future work will continue a number of findings found during this work. Firstly, exploring the encountered compiler issues with the soon to be released Spack 1.0 to take advantage of their changes to the compiler configuration. Slingshot 11 performance investigations will continue with recent versions of both fabric manager and host software as they become available. Finally further configuration improvements for applications as various options in each build are explored and provided to users.

Overall, the Grace Superchip is a good choice to provide a national "Supercomputer for Scientific Applications", with experience being gained over time with real science projects increasing the exemplar use-cases for Isambard 3 for future studies.

ACKNOWLEDGMENTS

The continuation of the GW4 Isambard project could not have happened without support from a large community. First, the co-investigators at the four GW4 universities who helped to write the proposal, including: Prof James Davenport (Bath), Prof Adrian Mulholland (Bristol), Prof Martyn Guest (Cardiff), Prof Gavin Tabor (Exeter). Second, the operations group who designed and now run the Isambard system, including: Georgie Wellock (Bath); Jose Munoz Criollo (Cardiff); Andrew Cowley (Exeter); Dave Moore (Bristol); Chris Peart and Qamar Nazir (HPE/Cray); Filippo Spiga (NVIDIA). Finally, thank you to the users of the Isambard 1 & 2 systems who were part of the initial journey to adopt Arm in HPC. The authors acknowledge the use of resources provided by the Isambard 3 Tier-2 HPC Facility. Isambard 3 is hosted by the University of Bristol and operated by the GW4 Alliance (https://gw4.ac.uk) and is funded by UK Research and Innovation; and the Engineering and Physical Sciences Research Council [EP/X039137/1].

REFERENCES

[1] S. McIntosh-Smith, J. Price, T. Deakin, and A. Poenaru, "Comparative Benchmarking of the First Generation of HPC-Optimised Arm Processors on Isambard," in *Cray User Group 2019*, May 2019.

[2] S. McIntosh-Smith, J. Price, A. Poenaru, and T. Deakin, "Benchmarking the first generation of production quality Arm-based supercomputers," in *Concurrency and Computation: Practice and Experience*, 2020. doi: 10.1002/cpe.5569.

[3] V. Karakasis, V. H. Rusu, A. Jocksch, J.-G. Piccinali, and G. Peretti-Pezzi, "A regression framework for checking the health of large HPC systems," in *Cray User Group 2017*, Redmond, 2017.

[4] T. Gamblin *et al.*, "The Spack package manager: Bringing order to HPC software chaos," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2015. doi: 10.1145/2807591.2807623.

[5] J. D. McCalpin, "Sustainable Memory Bandwidth in High Performance Computers," *Silicon Graphics Inc*, 1995.

[6] NVIDIA, "Arm Kernels." https://github.com/NVIDIA/arm-kernels.git, 2023.

[7] A. C. Mallinson, D. A. Beckingsale, W. P. Gaudin, J. A. Herdman, J. M. Levesque, and S. A. Jarvis, "CloverLeaf: Preparing Hydrodynamics Codes for Exascale," in *Cray User Group 2013*, 2013.

[8] S. McIntosh-Smith *et al.*, "TeaLeaf: A mini-application to enable design-space explorations for iterative sparse linear solvers," in *Proceedings - IEEE International Conference on Cluster Computing, ICCC*, 2017. doi: 10.1109/CLUSTER.2017.105.

[9] R. J. Zerr and R. S. Baker, "SNAP: SN (discrete ordinates) application proxy - proxy description." Los Alamos National Laboratories, 2013.

[10] M. Martineau and S. McIntosh-Smith, "Exploring On-Node Parallelism with Neutral, a Monte Carlo Neutral Particle Transport Mini-App," in *Proceedings - IEEE International Conference on Cluster Computing, ICCC*, 2017. doi: 10.1109/CLUSTER.2017.83.

[11] J. Liu *et al.*, "Micro-benchmark level performance comparison of high-speed cluster interconnects," in *Proceedings - 11th Symposium on High Performance Interconnects, HOTI 2003*, 2003. doi: 10.1109/CONECT.2003.1231479.

[12] S. J. Clark *et al.*, "First principles methods using CASTEP," *Zeitschrift fur Kristallographie*, vol. 220, no. 5–6, 2005, doi: 10.1524/zkri.220.5.567.65075.

[13] T. D. Kühne *et al.*, "CP2K: An electronic structure and molecular dynamics software package -Quickstep: Efficient and accurate electronic structure calculations," *Journal of Chemical Physics*, vol. 152, no. 19. 2020. doi: 10.1063/5.0007045.

[14] H. Bekker, H. Berendsen, E.J. Dijkstra, S. Achterop, R. Drunen, and D. Van Der Spoel, "Gromacs: a parallel computer for molecular dynamics simulations – ScienceOpen," *Physics Computing*, vol. 92, no. January, 1993.

[15] J. C. Phillips *et al.*, "Scalable molecular dynamics on CPU and GPU architectures with NAMD," *Journal of Chemical Physics*, vol. 153, no. 4, 2020, doi: 10.1063/5.0014475.

[16] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, vol. 12, no. 6, 1998, doi: 10.1063/1.168744.

[17] I. Spisso, R. da Vià, R. Ponzini, F. Magugliani, and N. Ashton, "HPC Benchmark Project: follow up," *6th OpenFOAM Conference 2018*. Hamburg, 2018.

[18] M. A. Heroux *et al.*, "Improving Performance via Mini-applications," *Sandia National Laboratories*. 2009.

[19] T. Koskela *et al.*, "Principles for Automated and Reproducible Benchmarking," in *ACM International Conference Proceeding Series*, 2023. doi: 10.1145/3624062.3624133.

[20] J. D. McCalpin, "Bandwidth Limits in the Intel Xeon Max (Sapphire Rapids with HBM) Processors," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2023. doi: 10.1007/978-3-031-40843-4_30.

[21] S. McIntosh-Smith, Alam S. R., and C. Woods, "Isambard-AI: a leadership class supercomputer optimised specifically for Artificial Intelligence," in *Cray User Group 2024*, 2024.