Are we ready for broader adoption of ARM in the HPC community: Performance and Energy Efficiency Analysis of Benchmarks and Applications Executed on High-End ARM Systems

NIKOLAY A. SIMAKOV, ROBERT L. DELEON, JOSEPH P. WHITE, MATHEW D. JONES, and THOMAS R. FURLANI, Center for Computational Research, SUNY University at Buffalo, USA

EVA SIEGMANN and ROBERT J. HARRISON, Institute for Advanced Computational Science, Stony Brook University, USA

A set of benchmarks, including numerical libraries and real-world scientific applications, were run on several modern ARM systems (Amazon Graviton 3/2, Futjutsu A64FX, Ampere Altra, Thunder X2) and compared to x86 systems (Intel and AMD) as well as to hybrid Intel x86/NVIDIA GPUs systems. For benchmarking automation, we used the application kernel module of XDMoD. XDMoD is a comprehensive suite for HPC resource utilization and performance monitoring. The application kernel module enables continuous performance monitoring of HPC resources through the regular execution of user applications. It has been used on the Ookami system (one of the first USA-based Fujitsu ARM A64FX SVE 512 systems) since the resource was deployed. In most cases the ARM processors are comparable or only slightly slower than x86 processors. In some cases, ARM delivers faster performance than its counterpart. In most tested applications, the ARM solution is more energy efficient than x86 CPU systems but smaller than that of a GPU-enabled system (for those cases where the application has GPU support). Given the high core count per node, comparable performance, and competitive pricing, current high-end ARM CPUs are already a valid choice as a primary HPC system processor. In this work, we present our results and experience from deploying and running HPC applications on numerous ARM systems and compare their performance to x86 processors.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Applied computing**; • **Hardware**;

Additional Key Words and Phrases: benchmarks, energy efficiency, HPC, ARM, x86, GPU

ACM Reference Format:

Nikolay A. Simakov, Robert L. DeLeon, Joseph P. White, Mathew D. Jones, Thomas R. Furlani, Eva Siegmann, and Robert J. Harrison. 2023. Are we ready for broader adoption of ARM in the HPC community: Performance and Energy Efficiency Analysis of Benchmarks and Applications Executed on High-End ARM Systems. In *IWAHPCE '23: International Workshop on Arm-based HPC: Practice and Experience, February 27, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The ARM CPUs are well known for their usage in embedded systems and mobile computing devices such as smartphones and tablets. For some time ARM CPUs have also been used in niche Linux server products like file and web servers. More recently, several ARM CPUs have been adapted to HPC workloads and some of them were specifically designed for scientific calculations. The former are Ampere Altra and Amazon Graviton 2. The latter are Fujitsu A64 and partially Amazon Graviton 3. The homogeneous, Fujitsu A64FX based, Fugaku supercomputer was the fastest supercomputer

Manuscript submitted to ACM

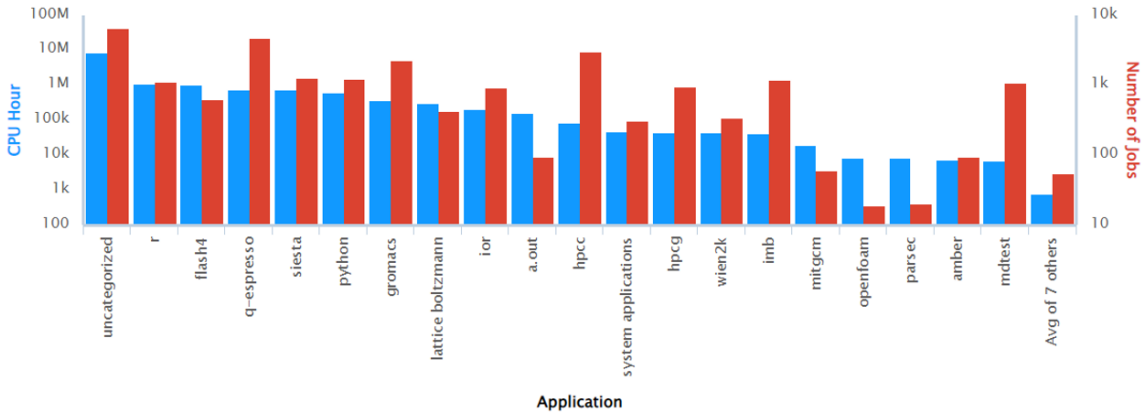

Fig. 1. Application usage from 2021-01 to 2022-09-30 on Ookami cluster (an ARM Fujitsu A64FX machine with SVE support, 512 bit wide)

for almost two years (June 2020 - May2022). Many computational centers are already experimenting with ARM-based servers and the purpose of this work is to evaluate the readiness of software and hardware for a potential migration to the ARM platform.

Our group provides utilization monitoring for one of the first Fujitsu A64FX machines in the USA, the Ookami cluster, which is installed in Stony Brook University. The XD Metrics on Demand (XDMoD) tool was used to make Figure 1 which shows the application usage on Ookami. XDMoD is designed for the comprehensive management of HPC systems, provides users, managers, and operations staff with access to utilization data, job and system level performance data, and quality of service data for HPC resources [6]. Originally developed to provide an independent audit capability for the XSEDE program, XDMoD was later open-sourced and is widely used by the university, government, and industry HPC centers [15].

The application kernel performance monitoring module of XDMoD [20] allows automatic performance monitoring of HPC resources through the periodic execution of application kernels, which are based on benchmarks or real-world applications implemented with sensible input parameters. In the past, the performance monitoring module was used to study the effect of node-sharing policies on the performance of individual jobs [17] and the analysis of the effects of Meltdown-Spectre remedies on application performance [18]. This module is used in this work to benchmark the ARM CPUs.

2 RELATED WORK

There are a number of works assessing the compute performance and energy efficiency of the ARM system. In 2013 Jarus and others [9] compared the 48-processor ARM Cortex A9 system to x86 machines of that time using seven benchmarks, including High Performace LINPACK. The authors found that the tested ARM CPU was often among the slowest CPUs but always had good performance per Watt. Later, Maqbool and others [11] obtained similar results using a different ARM Cortex A9; they also included a MySQL benchmark in their study. In 2019 McIntosh-Smith [12] and others compared the performance of Cavium ThunderX2 in eight scientific applications, including CP2K, GROMACS, NAMD, NEMO, OpenFOAM and VASP. The authors found that ThunderX2 provides the same level of performance and similar energy efficiency as its x86 counterpart but for a smaller unit price. The most recent study [5] compares the performance

Table 1. Compute systems tested in this study. Used abbreviation: SBU - Stony Brook University, AWS - Amazon Web Services, PSC - Pittsburgh Supercomputing Center, SDSC - San-Diego Supercomputing Center, TACC - Texas Advanced Computing Center, Purdue - Purdue Univeristy, UB - University at Buffalo, KNL - Knights Landing. Neov. - Neoverse

Resource	CPU	Arch/Core Name	Proc. , nm	SIMD	Release Date	Cores	Freq,GHz base/turbo
ARM							
SBU Ookami	Fujitsu A64FX	v8.2-A	7	SVE 512b	~2019	48	1.8
SBU-Ookami	Cavium ThunderX2	v8.1	14	NEON 128b	2018	64	2.0-2.5
AWS 48cores	Amazon Graviton 2	v8.2, Neov. N1	7	128b	Nov-19	48	2.6
AWS 48cores	Amazon Graviton 3	v8.5, Neov. V1	5	SVE 512b	Nov-21	48	2.5
AWS 64cores	Amazon Graviton 3	v8.5, Neov. V1	5	SVE 512b	Nov-21	64	2.5
Google 48cores	Ampere Altra	v8.2+, Neov. N1	7	128b	Mar-21	48	Up to 3.0
Azure-Altra-48	Ampere Altra	v8.2+, Neov. N1	7	128b	Mar-21	48	Up to 3.0
Azure-Altra-64	Ampere Altra	v8.2+, Neov. N1	7	128b	Mar-21	64	Up to 3.0
x86 AMD							
PSC-Bridges 2	EPYC 7742	Zen2(Rome)	14	AVX2 256b	Mid-19	128	2.25/3.4
SDSC-Expanse	EPYC 7742	Zen2(Rome)	14	AVX2 256b	Mid-19	128	2.25/3.4
Purdue-Anvil	EPYC 7763	Zen3(Milan)	7+	AVX2 256b	Mar-21	128	2.45/3.5
x86 Intel							
TACC-Stampede 2	Xeon Phi 7250	KNL	14	AVX512	Q2 2016	68	1.4/1.6
TACC-Stampede 2	Xeon Platinum 8160	Skylake-X	14	AVX512	Q3 2017	48	2.1/3.7
TACC-Stampede 2	Xeon Platinum 8380	Ice Lake	10	AVX512	Q2 2021	80	2.3/3.4
UB-HPC 32core	Xeon Gold 6130	Skylake-X	14	AVX512	Q3 2017	32	2.1/3.7
UB-HPC 56core	Xeon Gold 6330	Ice Lake	10	AVX512	Q2 2021	56	2/3.7
x86 Intel and	**NVIDIA GPU**						
UB-HPC V100x2	Xeon Gold 6130	Skylake-X	14	AVX512	Q3 2017	32	2.1/3.7
UB-HPC A100x2	Xeon Gold 6330	Ice Lake	10	AVX512	Q2 2021	56	2/3.7

of three ARM CPUs (Cavium ThunderX2, Ampere Alta and Fujitsu A64FX) individually and in combination with NVIDIA GPUs (A100 and V100) on ten scientific applications. The study concluded the modern ARM CPU performed on a par with the modern x86 and PowerPC CPUs.

The current work extends the earlier analysis with more modern CPUs, as well as, with cloud-based ARM machines. The study is conducted on one benchmark and five applications. Several of these were not used in earlier studies. We also perform an energy efficiency estimation.

3 METHODS

The tests were executed automatically using the XDMoD application kernel remote runner (AKRR) module. The automated process parses the application output and ingests the results into the XDMoD database (see [20] for more details). Later the metrics were queried from the database and analyzed in R. An application kernel consists of an application run with a with a particular set of input parameters. AKRR executes each benchmark or application as an individual batch job. So each job is simply a single individual test run executed on the compute resource. For statistical analysis, we need to have multiple jobs executed on the same hardware.

3.1 Compute Systems

We have tested five ARM CPUs including Fujitsu A64FX, Amazon Graviton 2 and 3, Ampere Altra and Cavium ThunderX2. They were compared to two x86 AMD CPUs (Zen2-Rome and Zen3-Millan) and five x86 Intel CPUs (one Knight Landing, two Skylake-X, and two Icelake). In addition to that we also tested two systems with NVIDIA GPUS (with 2xV100 GPUs and 2xA100 GPUs). Overall, eighteen different hardware configurations were used from nine different resource providers, including cloud and traditional HPC services. The summary of the tested systems is shown in Table 1. All calculations were performed on a single node or single virtual machine instance.

The Fujitsu A64FX and Cavium ThunderX2 were accessed on the Ookami system at Stony Brook Univerity. Amazon Graviton 2/3 and Ampere Altra were accessed through cloud services as virtual machines. Amazon Web Services was used for Graviton 2 and 3. Google Cloud and Microsoft Azure were used for Ampere Altra but the exact CPU model is unknown to us. CloudBank [14] was used to access these clouds.

The reference x86 CPU systems were from UB's Center for Computational Research as well as from several centers under the NSF funded ACCESS-CI program [1], namely Pittsburgh Supercomputing Center (PSC), San-Diego Supercomputing Center (SDSC), Texas Advanced Computing Center (TACC) and Purdue University.

3.2 Application Kernels

Below is a description of the test applications and their associated input parameters. Building compute-intensive scientific applications to work on such diverse compute systems is a challenging job. To simplify and automate the process, we used Spack [7], a package manager for supercomputers. In the past, we had spent some time ensuring that the recipes would engage the proper flags for optimal compilation of the test applications, and our changes were merged with upstream Spack.

HPCC (HPC Challenge) benchmark combines multiple benchmarks together. Here we are reporting on three of them: High Performance LINPACK, Matrix-Matrix multiplication and Fast Fourier Transform (FFT). LINPACK solves a linear system of equations using all cores in parallel. The performance is measured in Giga Floating point Operations Per Second (GFLOPS) and corresponds to the performance of the application on all allocated compute resources. We also report GFLOPS/Core, which are the total GFLOPS divided by the number of cores. Matrix-matrix multiplication is calculated using `dgemm` routine from the BLAS library; the calculation is done on all cores in an embarrassingly parallel way. The performance is measured in GFLOPS/core, the cumulative performance per all allocated resources is also calculated. The FFT is calculated in parallel by all allocated resources. The performance is reported in GFLOPS. Similarly to LINPACK we also report GFLOPS/Core. The original HPCC only supported FFTW2 and build-in FFTE libraries. We have added FFTW3 API support and our changes was recently accepted to upstream. We also implemented a Spack recipe for HPCC, which has been added to the main Spack repository.

There are a large number of numeric libraries which perform linear algebra and FFT. On Fujitsu A64FX, we tested the following combinations OpenBlas and FFTW3 (GCC toolchain), Fujitsu Numeric Library and Fujitsu FFTW3 (Fujitsu toolchain), Cray LibSci and Cray FFTW3 (Cray toolchain) and ARM Performance library (ARM toolchain). For other ARM machines and x86 AMD machines we used OpenBlas and FFTW3 (OSS toolchain). For x86 Intel machines, in addition to OpenBlas and FFTW3 (OSS toolchain) we also used Intel MKL libraries.

Extracting energy efficiency metrics from HPCC is complicated because it combines multiple tests and setups, some of which are serial. Nevertheless, we are able to report on mean power, and energy efficiency the number of complete HPCC runs per kWh (jobs/kWh).

GROMACS is a computational software for the simulation of biomolecular systems like proteins, membranes, DNA, and RNA [16]. It calculates how the atoms move over time under a classical physics approximation by solving Ordinary Differential Equations based on Newton's first law. The benchmark used is a protein embedded into a membrane and contains 81,743 atoms [10]. As a performance metric, we use simulated nanoseconds in a day (ns/day); a higher number corresponds to better performance. For energy efficiency, we use simulated nanoseconds per user kilo-Watt-hours (ns/kWh, larger-more efficient). Here the simulated nanoseconds correspond to useful work done and kilo-Watt-hours is the energy used. To estimate energy used in a day we used power averaged over the second half of the job.

The tested compute systems have significantly different compute abilities. Consequently, the wall time for the same test can differ widely across the systems. The test also has to run for some time so that the machine gets into a sustained state rather than an initial frequency boost mode. Fortunately, because of the performance metric that we used and given that the problem complexity doesn't change much over time, we can use a different number of steps for the different compute systems. We use 50,000 steps for the slower systems, 100,000 for the medium speed systems, and 200,000 for the GPUs. Due to the problem size we can only utilize one GPU efficiently, thus we use only one GPU of the two available GPUs.

NWChem [2] is a *ab initio* computational chemistry software package developed by Pacific Northwest National Laboratory. The input to the benchmark runs is the Hartree-Fock energy calculation of a single gold ion (Au+) with MP2 and Coupled Cluster corrections. This test case has been used by us for over a decade for performance monitoring purposes. Modern systems have outgrown it as the run time is typically under a minute, and in the future, we intend to switch to a larger case. For the performance metric we used wall time (i.e. smaller-better), and for energy efficiency, we use the number of the test calculations done per kWh energy (jobs/kWh, larger-better).

OpenFOAM is a library and a collection of applications for the numerical solution of Partial Differential Equations. It is often employed for computational fluid dynamics. The test case is a calculation of incompressible airflow around a motorcycle. It is based on one of the tests included in the OpenFOAM suite (`incompressible\simpleFoam\motorBike`). We have quadrupled the initial grid in each direction to increase resolution and problem size. The grid is further refined around the obstacle, and the Navier–Stokes equations are solved on an unstructured grid. Similar to NWChem for performance metrics, we used wall time (i.e. smaller-better) and the test calculations done per kWh energy (jobs/kWh, larger-better).

ENZO [3] is an Adaptive Mesh Refinement (AMR) code for astrophysics and cosmology simulation. The test case is a reionization simulation and is based on the `ReionizationRadHydro` example provided with the Enzo software. In the test, the initial grid was increased to 128x128x128, and the size was increased to 20 Mpc/h.

Similar to NWChem and ENZO for performance metrics, we are using wall time (i.e. smaller-better) and the test calculations done per kWh energy (jobs/kWh, larger-better).

AI-Benchmark-Alpha [8] includes multiple machine learning tasks utilizing deep convolutional neural networks. The benchmark utilizes Tensorflow for the computation. Tests includes classification, image to image mapping, image segmentation, image inpainting, sentence sentiment analysis and text translation. The performance is reported as an AI Score, which is split into training and inference scores. Each score is the geometric mean of the individual test's score multiplied by 10,000 (in order to be integer). The individual score is the ratio of reference time and the actual test time. NVIDIA TITAN X Pascal was used as a reference platform. For this metric, higher scores correspond to better performance. The energy efficiency is calculated as AI score per W.

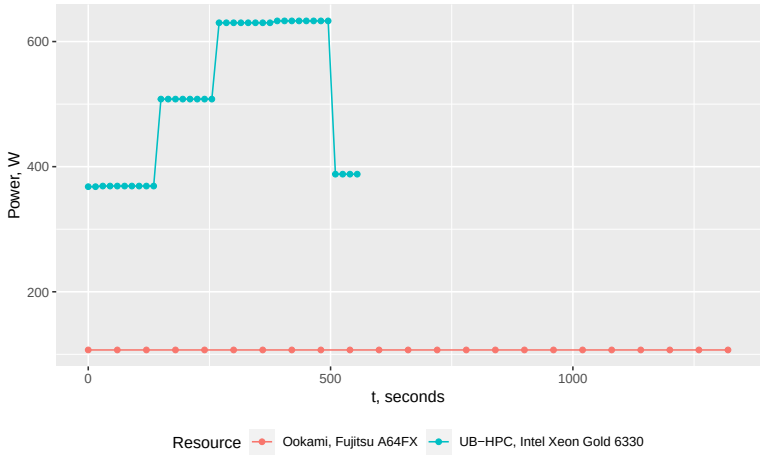

Fig. 2. Instantaneous electric power usage on Ookami (Fujitsu A64FX) and UB-HPC (Intel Xeon Gold 6330) resources during OpenFOAM application execution. Ookami has the same energy consumption throughout the whole job while UB-HPC has four distinct regions with different power loads.

Due to a large number of applications and systems tested as well as multiple ways how applications can be built (different compilers and libraries) we didn't test every possible combination. That is if the combination is not present it is usually because we didn't attempt running it.

3.3 Energy Consumption Measurements and Estimations

On Ookami (Fujitsu A64FX machine) the power metrics were collected from the baseboard management controller (BMC) on each chassis via the intelligent platform management interface (IPMI). Each chassis on Ookami has six compute nodes and the BMC provides one-minute average power usage for each compute node. The Fujitsu A64FX CPUs on Ookami have a fixed clock rate and power consumption is very stable, and it is the same throughout the duration of the batch job (see Figure 2). Thus for the energy efficiency, we used maximum power drained and mean power interchangeably since in A64FX they are the same. Some of the tests were very short and we were not able to get power measurements (for example, HPCC). In this case, we used the mean power from OpenFOAM runs since the A64FX power usage doesn't change much over time.

To estimate the energy consumption of Ampere Altra CPU we used the reported mean power during the GROMACS application execution [13] and corrected it by the thermal design power (TDP) difference between the CPU model used in the reference calculation and the presumed model used in the Microsoft Azure and Google Cloud (300 W (performance in GROMACS [13]) - (210 W - 180 W) (which is the TDP difference between the CPU models)). For other applications, we assumed that they are as compute-intensive as GROMACS and will have similar power demands.

The compute nodes on the academic HPC cluster at UB CCR are instrumented with the Prometheus monitoring software [4]. The Prometheus IPMI exporter is used to collect power consumption data from the data center manageability interface (DCMI) interface on the BMC of each compute node. Intel's Icelake CPU has a variable power consumption depending on load pattern (Figure 2).

Table 2. HPCC performance. On Fujitsu A64FX performance of different libraries is reported, they are identified by the note in the System column: GCC - OpenBlas and FFTW3, Fujitsu - Fujitsu SSL II Numeric Library and Fujitsu FFTW3, Cray - Cray LibSci and Cray FFTW3, ARM - ARM Performance library, and ICC - Intel MKL for linear algebra and FFTE for FFT. The difference in FFT for PSC-Bridges2 and SDSC-Expanse is due to different FFTW libraries (2 vs 3). * - a calculated estimate for azure-altra-64. ** - estimate from openfoam run. N - number of runs.

CPU/System	Cores	Matrics Multiplication		LINPACK		FFT		Power, W	Energy Eff., Jobs per kWh	N
		GFLOPS	GFLOPS/Core	GFLOPS	GFLOPS/Core	GFLOPS	GFLOPS/Core			
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, GCC)	48	1363	28.4 ± 0.1	828 ± 27	17.3	6.2 ± 0.7	0.13	110**	560	60
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, FJ)	48	1978	41.2 ± 0.2	1177 ± 19	24.5	24.4 ± 0.9	0.51	110**	185	60
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, ARM)	48	1651	34.4 ± 1.8	884 ± 72	18.4	0.3 ± 0.0	0.01	110**	335	60
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, Cray)	48	917	19.1 ± 3.7	758 ± 149	15.8	6.8 ± 0.6	0.14	110**	204	60
ARM Cavium ThunderX2 (SBU-Ookami)	64	742	11.6 ± 2.1	522 ± 106	8.2	33.5 ± 3.7	0.52			14
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	816	17 ± 0.0	682 ± 1	14.2	27.1 ± 0.6	0.56			20
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	48	907	18.9 ± 0.3	776 ± 10	16.2	55.4 ± 0.5	1.15			20
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	64	1158	18.1 ± 0.0	965 ± 1	15.1	71 ± 0.7	1.11			20
ARM Ampere Altra, Neoverse N1 (Azure)	48	816	17 ± 0.0	675 ± 17	14.1	26.5 ± 0.4	0.55			11
ARM Ampere Altra, Neoverse N1 (Azure)	48	826	17.2 ± 0.0	691 ± 18	14.4	26.8 ± 0.7	0.56			11
ARM Ampere Altra, Neoverse N1 (Azure)	64	1037	16.2 ± 0.0	850 ± 4	13.3	33.1 ± 1.1	0.52	270*	314	20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (PSC-Bridges-2)	128	2624	20.5 ± 0.7	1895 ± 42	14.8	50.3 ± 0.3	0.39			20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (SDSC Expanse)	128	3200	25 ± 1.4	1721 ± 47	13.4	71.8 ± 2.0	0.56			20
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	3046	23.8 ± 1.6	2176 ± 100	17.0	54.7 ± 4.8	0.43			20
x86 Intel Xeon Phi 7250, KNL, AVX512 (TACC-Stampede 2)	68	340	5 ± 0.3	986 ± 8	14.5	46.5 ± 0.7	0.68			20
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2	48	2122	44.2 ± 1.7	1158 ± 34	24.1	35.8 ± 1.9	0.75			20
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	3824	47.8 ± 0.6	1713 ± 5	21.4	76.4 ± 2.0	0.96			12
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UB-HPC)	32	1536	48 ± 2.0	997 ± 54	31.2	50.9 ± 1.9	1.59	345±31	74	53
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UB-HPC)	56	2761	49.3 ± 1.1	1396 ± 37	24.9	47.9 ± 0.7	0.86	588±64	109	12
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UB-HPC, ICC)	56	2845	50.8 ± 1.0	1399 ± 13	25.0	28.2 ± 0.3	0.50	501±107	299	12

It is important to note that our power measures and estimates only include the energy consumption of a single compute node and do not take into consideration the energy use of other systems such as network switches or cooling.

3.4 Source Code

- The XDMoD and application kernel remote runner source code is given in https://open.xdmod.org/.
- Application kernel input files can be found in https://github.com/ubccr/akrr/tree/master/akrr/appker_repo/inputs.
- The results of the tests and analysis scripts are available at https://github.com/nsimakov/ARM_Benchmarks_IWAHPCE_2023.

4 RESULTS AND DISCUSSION

We will open this section by describing our experience with building all these benchmarks and applications on ARM machines and how it is related to traditional x86 machines. For all cloud-based instances, we used Spack for almost all applications, anecdotally we build a few applications manually and their performance was actually slightly slower. For Fujitsu A64FX, roughly half of the applications were built manually and the remainder with Spack. AI-Benchmark and Tensorflow were installed from binaries on all platforms. On x86 systems we utilized both manual and Spack installations. Overall the building experience on ARM was similar to that of x86 systems.

HPCC benchmark utilizes linear algebra (BLAS) and FFT libraries, which are widely used in scientific applications and therefore are of particular interest. The matrix-matrix multiplication is one of the few practical calculations capable of approaching theoretical FLOPS and Single Instruction Multiple (SIMD) width plays a crucial role here. As can be seen from Table 2 (matrix multiplication, GFLOPS/core, column) CPUs with 512 bit wide SIMD instructions have twice

Table 3. Gromacs performance. In tests with the GPU only one GPU was used due to the small problem size. The GCC tag in the System column indicates compilation with the GCC compiler, Fujitsu with the Fujitsu compiler and ICC with the Intel compiler. N is the number of performed runs.

CPU/System	Cores	Simulation Speed, ns/day	Simulation Speed per Core, ns/day/core	Power, W	Energy Efficiency, ns/kWh	N
ARM Fujitsu A64FX, SVE 512bit (SBU-Ookami, GCC)	48	22.3 ± 0.2	0.46	111 ± 8	8.43 ± 0.6	32
ARM Fujitsu A64FX, SVE 512bit (SBU-Ookami, Fujitsu)	48	22.8 ± 0.3	0.48	105 ± 5	9.06 ± 0.4	12
ARM Cavium ThunderX2 (SBU-Ookami)	64	28.8 ± 4.2	0.45			14
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	37.8 ± 0.1	0.79			20
ARM Amazon Graviton 3, Neoverse V1, SVE 256bit (AWS)	48	57.0 ± 0.4	1.19			20
ARM Amazon Graviton 3, Neoverse V1, SVE 256bit (AWS)	64	71.4 ± 1.0	1.12			20
ARM Ampere Altra, Neoverse N1 (Google)	48	39.0 ± 1.8	0.81			11
ARM Ampere Altra, Neoverse N1 (Azure)	48	41.0 ± 2.2	0.85			11
ARM Ampere Altra, Neoverse N1 (Azure)	64	56.5 ± 0.6	0.88	270 *	8.71	20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (PSC Bridges-2)	128	109.6 ± 4.8	0.86			20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (SDSC Expanse)	128	99.8 ± 8.6	0.78			20
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	169.9 ± 4.4	1.33			20
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	70.4 ± 0.8	1.47			11
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	133.3 ± 6.0	1.67			20
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UBHPC)	32	37.6 ± 0.9	1.18	379 ± 33	4.17 ± 0.4	21
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UBHPC ICC)	32	39.3 ± 0.9	1.23	367 ± 35	4.5 ± 0.5	11
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UBHPC)	56	81.7 ± 6.9	1.46	633 ± 28	5.38 ± 0.4	12
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UBHPC ICC)	56	103.0 ± 2.0	1.84	619 ± 17	6.94 ± 0.2	17
x86 Intel Xeon Gold 6130, NVIDIA V100x2 (UBHPC)	32	145.1 ± 2.8		435 ± 7	13.91 ± 0.3	12
x86 Intel Xeon Gold 6330, NVIDIA A100x2 (UBHPC)	56	236.5 ± 10.8		707 ± 9	13.94 ± 0.8	11

Table 4. NWChem performance.** - NWChem of version 6.8, other setups uses 7.0.2. N is the number of performed runs.

CPU/System	Cores	Wall Clock Time, Seconds	Power, W	Energy Efficiency, Jobs per kWh	N
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, GCC)	48	62.7 ± 0.7	110 ± 0	522 ± 6	60
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	61.1 ± 0.9			12
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	48	36.6 ± 0.7			11
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	64	29.8 ± 0.4			20
ARM Ampere Altra, Neoverse N1 (Azure)	48	56.5 ± 2.7			11
ARM Ampere Altra, Neoverse N1 (Azure)	64	42.8 ± 0.5	270*	312	20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (PSC-Bridges-2)	128	32.4 ± 4.4			20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (SDSC Expanse)	128	28.6 ± 7.8			20
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	26.7 ± 0.3			20
x86 Intel Xeon Phi 7250, KNL, AVX512 (TACC-Stampede 2)**	68	262.1 ± 22.1			20
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)**	48	50.3 ± 0.3			12
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	31.2 ± 0.2			8
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	19.2 ± 1.2			11
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UB-HPC)	32	90 ± 1.6	332 ± 50	124 ± 25	27
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UB-HPC)	56	46.9 ± 0.6	376 ± 2	204 ± 3	11

the performance of CPUs with more narrow SIMD instructions. Overall the CPUs with 512 bit wide SIMD show the highest performance in this test, among them ARM Fujitsu A64FX with the 512 bit wide SVE instruction set and Intel CPUs with the AVX512 instruction set (excepting Xeon Phi). With the exception of older and slower ThunderX2, the other ARM CPUs show slower performance but comparable to AMD Zen2 CPUs.

Table 5. OpenFoam performance. N is the number of performed runs.

CPU/System	Cores	Wall Clock Time, Minutes	Meshing Time, Minutes	Solver Time, Minutes	Power, W	Energy Efficiency, Jobs per kWh	N
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, GCC)	48	28.4 ± 0.9	14.6 ± 0.9	12.4 ± 0.1	110 ± 7	19.3 ± 1.6	21
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, FJ)	48	22.4 ± 0.3	8.5 ± 0.1	10.9 ± 0.2	111 ± 7	24.1 ± 1.6	21
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	11.9 ± 0.3	3.5 ± 0.2	8 ± 0.1			10
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	48	7.1 ± 0.2	2.2 ± 0.2	4.7 ± 0.0			5
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	64	6.8 ± 0.1	2.2 ± 0.1	4.4 ± 0.1			20
ARM Ampere Altra, Neoverse N1 (Azure)	48	11.1 ± 0.2	3.2 ± 0.2	7.6 ± 0.1			10
ARM Ampere Altra, Neoverse N1 (Azure)	64	10.9 ± 0.4	3.2 ± 0.2	7.2 ± 0.2	270*	20.4	20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (SDSC Expanse)	128	9.5 ± 1.9	5.6 ± 1.4	3.2 ± 1.1			20
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	6.6 ± 0.2	3.1 ± 0.5	2.9 ± 0.5			19
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	10.7 ± 0.4	3.7 ± 0.3	6.4 ± 0.1			10
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	6.8 ± 0.3	2.6 ± 0.2	3.7 ± 0.3			20
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UB-HPC)	32	13.2 ± 0.8	4.1 ± 0.4	7.7 ± 0.1	375 ± 35	12.3 ± 1.0	23
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UB-HPC)	56	8.9 ± 0.5	2.8 ± 0.3	4.7 ± 0.2	505 ± 34	13.4 ± 0.9	20

Table 6. AI Benchmark Alpha performance. N is the number of performed runs.

CPU/System	Cores	AI Score	Inference Score	Training Score	Power, W	AI Score per W	N
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami)	48	1034 ± 3	535 ± 2	499 ± 2	111 ± 7	9.4 ± 0.6	20
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	3030 ± 12	1676 ± 7	1355 ± 6			12
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	48	4581 ± 12	2407 ± 10	2174 ± 8			11
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	64	4850 ± 31	2708 ± 21	2143 ± 13			20
ARM Ampere Altra, Neoverse N1 (Azure)	48	3177 ± 15	1803 ± 10	1375 ± 6			11
ARM Ampere Altra, Neoverse N1 (Azure)	64	3214 ± 20	1977 ± 15	1238 ± 6	270*	11.9	20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (SDSC Expanse)	128	2696 ± 17	1761 ± 14	936 ± 9			11
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	3079 ± 26	1992 ± 16	1087 ± 13			11
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	3606 ± 20	2292 ± 18	1314 ± 4			11
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	8805 ± 27	3725 ± 20	5081 ± 14			11
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UB-HPC)	32	3233 ± 253	1941 ± 165	1292 ± 88	403 ± 14	8 ± 0.5	11
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UB-HPC)	56	10197 ± 53	4398 ± 31	5799 ± 29	543 ± 33	18.9 ± 1.2	12
x86 Intel Xeon Gold 6130, NVIDIA V100x2 (UB-HPC)	32	32628 ± 433	15656 ± 278	16972 ± 163	379 ± 34	86.8 ± 8.3	11
x86 Intel Xeon Gold 6330, NVIDIA A100x2 (UB-HPC)	56	59323 ± 378	29691 ± 290	29631 ± 152	561 ± 69	107.2 ± 13.6	11

Table 7. ENZO performance.

CPU/System	Cores	Wall Clock Time, Minutes	Power, W	Energy Efficiency, Jobs per kWh	N
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, GCC)	48	115.7 ± 17.7	112 ± 7	4.7 ± 0.5	10
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	23.6 ± 1.1			12
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	48	17 ± 1.2			11
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	64	13.2 ± 0.7			20
ARM Ampere Altra, Neoverse N1 (Azure)	48	21 ± 1.0			11
ARM Ampere Altra, Neoverse N1 (Azure)	64	15.9 ± 0.8	270*	14	20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (PSC-Bridges-2)	128	7.1 ± 0.4			20
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (SDSC Expanse)	128	6.6 ± 0.4			20
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	6.9 ± 0.3			20
x86 Intel Xeon Phi 7250, KNL, AVX512 (TACC-Stampede 2)	68	14.7 ± 0.3			20
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	4.2 ± 0.1			20
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UB-HPC)	32	4.8 ± 0.3	338 ± 31	37.1 ± 3.9	50
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UB-HPC)	32	25.8 ± 0.9	379 ± 26	6.2 ± 0.4	11
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UB-HPC)	56	15.5 ± 0.6	559 ± 34	6.9 ± 0.4	11

In the LINPACK test, the SIMD width plays a lesser role and a higher core count can lead to higher overall performance. Still in per-core performance results are very similar for matrix multiplication. ARM Fujitsu A64FX and Intel CPUs with AVX512 show the highest per-core performance. Overall AMD Zen3 CPU shows the highest per node performance.

For FFT, SIMD width importance is even less important and although the Intel CPUs still show strong per core performance Amazon Graviton 3 shows per core performance higher than several current and older Intel CPUs. The performance of other the ARM CPUs are very comparable to the x86 chips.

GROMACS and other molecular dynamics simulation programs are among the top applications used on HPC resources [19]. Therefore the fast performance of such programs is important for ARM adoption. The per core performance of Graviton 2 and Ampere Altra is similar to AMD Zen2 and the larger number of cores per node allows it to outperform some Intel Skylake-X platforms (Table 3). The Graviton 3, per-core performance, approaches the Intel CPUs and outperforms the older AMD Zen2. GROMACS has an efficient GPU implementation and not surprisingly, the NVIDIA A100 GPU shows the highest performance. It is also the most energy efficient. The ARM Fujitsu A64FX and Ampere Altrare are more energy efficient than the Intel chips (by 25-30%).

Graviton 3 is the fastest in the ARM camp for **NWChem**, it is faster than older or smaller Intel CPUs and similar to much larger (core-wise) AMD CPUs. The fastest system is the 80 core Intel Ice Lake machine. Fujitsu A64FX is two times slower than that but it has 40% fewer cores. It is also 1.3 times slower than Xeon Gold 6330 but 2.6 times more energy efficient. Here the test problem is rather small and some caution is needed with these conclusions.

For **OpenFOAM** we found very similar results to NWChem. Graviton 3 is the fastest among the ARM CPUs and very close to the newer and the fastest x86 system Table 5. Fujitsu A64FX is 3.4 times slower than the fastest solution. Interestingly A64FX is 2.5 times slower than the Xeon Gold 6330 but 1.8 times more energy efficient.

AI-Benchmark-Alpha utilizes the TensorFlow library to perform 19 sets of the AI test. ARM Graviton 3, with BFloat16 support, shows the highest results among all CPUs except for the Intel Icelake CPUs, which implement Intel Deep Learning Boost (Table 6). Graviton 2 and Ampere Altra perform similarly to the AMD CPUs and the older Intel CPUs. Energy efficiency wise, Ampere Altra and Fujitsu A64FX perform better than the Xeon Gold 6130. Not surprisingly, NVIDIA A100 GPUs show the fastest absolute result. It is 5.8 times faster than the fastest pure CPU solution and 5.7 times more energy efficient.

Similar to most previous tests, the newer Graviton 3 shows the fastest performance among ARM CPUs for the **ENZO** application. However, it is three times slower than the fastest x86 solution. The calculation was compromised by numeric instabilities during problem-solving, requiring code compilation without optimization for almost all systems. Most likely different versions of the compiler and building on different CPU architectures produce binaries with different efficiencies. Being three times slower than the fastest x86 solution, Graviton 3 is still two times faster than the worst-case scenario on the x86 platform. Ampere Altra shows similar performance to Graviton 3, and its energy efficiency falls similarly between x86 solutions. Fujitsu A64FX shows slow performance and energy inefficiency. Most likely, utilization of math-safe optimizations instead of omitting optimization will be helpful here.

5 CONCLUSIONS

The building and compiling experience on ARM platforms is very similar to that of traditional HPC systems. As tested by HPCC benchmark, numerical libraries implementing linear algebra and FFT routines support ARM CPUs well and the latter exhibit a solid performance. ARM machines shows solid performance in Gromacs, OpenFOAM, Tensorflow and NWChem applications. The ARM performance is comparable to x86 counterparts, and they often outperform previous generations of x86 CPUs (largely Amazon Graviton3). In ENZO, Amazon Graviton3 and Ampere Altra are within the

x86 systems performance. Fujitsu A64 FX and Ampere Altra are more energy efficient in GROMACS, NWChem and OpenFOAM than x86 CPUs.

From the performance, energy efficiency and software building point of view as of now for all tested applications, modern ARM CPUs provide a viable alternative to their x86 counterparts and not only as a cheaper option for the GPU gateway. Intel Skylake-X is a very robust architecture for scientific calculations, more than five years later since the initial release, it still competes well with modern CPUs.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under awards OAC 2137603 and OAC 1927880. This work used computational resources at SUNY University at Buffalo, SUNY Stony Brook University, XSEDE (award CCR120014), ACCESS (award CIS220121) and CloudBank (award 20220912-mms).

REFERENCES

[1] ACCESS. 2022. Advanced Cyberinfrastructure Coordination Ecosystem: Services Support. https://access-ci.org/.

[2] E. Aprà, E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. van Dam, Y. Alexeev, J. Anchell, V. Anisimov, F. W. Aquino, R. Atta-Fynn, J. Autschbach, N. P. Bauman, J. C. Becca, D. E. Bernholdt, K. Bhaskaran-Nair, S. Bogatko, P. Borowski, J. Boschen, J. Brabec, A. Bruner, E. Cauët, Y. Chen, G. N. Chuev, C. J. Cramer, J. Daily, M. J. O. Deegan, T. H. Dunning, M. Dupuis, K. G. Dyall, G. I. Fann, S. A. Fischer, A. Fonari, H. Früchtl, L. Gagliardi, J. Garza, N. Gawande, S. Ghosh, K. Glaesemann, A. W. Götz, J. Hammond, V. Helms, E. D. Hermes, K. Hirao, S. Hirata, M. Jacquelin, L. Jensen, B. G. Johnson, H. Jónsson, R. A. Kendall, M. Klemm, R. Kobayashi, V. Konkov, S. Krishnamoorthy, M. Krishnan, Z. Lin, R. D. Lins, R. J. Littlefield, A. J. Logsdail, K. Lopata, W. Ma, A. V. Marenich, J. Martin del Campo, D. Mejia-Rodriguez, J. E. Moore, J. M. Mullin, T. Nakajima, D. R. Nascimento, J. A. Nichols, P. J. Nichols, J. Nieplocha, A. Otero-de-la Roza, B. Palmer, A. Panyala, T. Pirojsirikul, B. Peng, R. Peverati, J. Pittner, L. Pollack, R. M. Richard, P. Sadayappan, G. C. Schatz, W. A. Shelton, D. W. Silverstein, D. M. A. Smith, T. A. Soares, D. Song, M. Swart, H. L. Taylor, G. S. Thomas, V. Tipparaju, D. G. Truhlar, K. Tsemekhman, T. Van Voorhis, Á. Vázquez-Mayagoitia, P. Verma, O. Villa, A. Vishnu, K. D. Vogiatzis, D. Wang, J. H. Weare, M. J. Williamson, T. L. Windus, K. Woliński, A. T. Wong, Q. Wu, C. Yang, Q. Yu, M. Zacharias, Z. Zhang, Y. Zhao, and R. J. Harrison. 2020. NWChem: Past, present, and future. *The Journal of Chemical Physics* 152, 18 (2020), 184102. https://doi.org/10.1063/5.0004997 arXiv:https://doi.org/10.1063/5.0004997

[3] G. L. Bryan, M. L. Norman, B. W. O'Shea, T. Abel, J. H. Wise, M. J. Turk, D. R. Reynolds, D. C. Collins, P. Wang, S. W. Skillman, B. Smith, R. P. Harkness, J. Bordner, J.-h. Kim, M. Kuhlen, H. Xu, N. Goldbaum, C. Hummels, A. G. Kritsuk, E. Tasker, S. Skory, C. M. Simpson, O. Hahn, J. S. Oishi, G. C. So, F. Zhao, R. Cen, Y. Li, and The Enzo Collaboration. 2014. ENZO: An Adaptive Mesh Refinement Code for Astrophysics. *The Astrophysical Journal Supplement Series* 211, Article 19 (April 2014), 19 pages. https://doi.org/10.1088/0067-0049/211/2/19

[4] Cloud Native Computing Foundation. 2016. Prometheus. https://prometheus.io.

[5] Wael Elwasif, Sergei Bastrakov, Spencer H. Bryngelson, Michael Bussmann, Sunita Chandrasekaran, Florina Ciorba, M. A. Clark, Alexander Debus, William Godoy, Nick Hagerty, Jeff Hammond, David Hardy, J. Austin Harris, Oscar Hernandez, Balint Joo, Sebastian Keller, Paul Kent, Henry Le Berre, Damien Lebrun-Grandie, Elijah MacCarthy, Verónica G. Melesse Vergara, Bronson Messer, Ross Miller, Sarp Oral, Jean-Guillaume Piccinali, Anand Radhakrishnan, Osman Simsek, Filippo Spiga, Klaus Steiniger, Jan Stephan, John E. Stone, Christian Trott, René Widera, and Jeffrey Young. 2022. Early Application Experiences on a Modern GPU-Accelerated Arm-based HPC Platform. https://doi.org/10.48550/ARXIV.2209.09731

[6] T. R. Furlani, B. I. Schneider, M. D. Jones, J. Towns, D. L. Hart, S. M. Gallo, R. L. DeLeon, C. Lu, A. Ghadersohi, R. J. Gentner, A. K. Patra, G. Laszewski, F. Wang, J. T. Palmer, and N. Simakov. 2013. Using XDMoD to facilitate XSEDE operations, planning and analysis. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE '13)*. ACM, 8. https://doi.org/10.1145/2484762.2484763

[7] Todd Gamblin, Matthew P. LeGendre, Michael R. Collette, Gregory L. Lee, Adam Moody, Bronis R. de Supinski, and W. Scott Futral. 2015. The Spack Package Manager: Bringing order to HPC software chaos. In *Supercomputing 2015 (SC'15)*. Austin, Texas. http://tgamblin.github.io/pubs/spack-sc15.pdf

[8] Andrey Ignatov. 2019. AI Benchmark Alpha - open source python library for evaluating AI performance of various hardware platforms, including CPUs, GPUs and TPU. https://ai-benchmark.com/alpha.

[9] Mateusz Jarus, Sébastien Varrette, Ariel Oleksiak, and Pascal Bouvry. 2013. Performance Evaluation and Energy Efficiency of High-Density HPC Platforms Based on Intel, AMD and ARM Processors. In *Energy Efficiency in Large Scale Distributed Systems*, Jean-Marc Pierson, Georges Da Costa, and Lars Dittmann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 182–200.

[10] Carsten Kutzner, Szilárd Páll, Martin Fechner, Ansgar Esztermann, Bert L. de Groot, and Helmut Grubmüller. 2015. Best bang for your buck: GPU nodes for GROMACS biomolecular simulations. *Journal of Computational Chemistry* 36, 26 (2015), 1990–2008. https://doi.org/10.1002/jcc.24030 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.24030

[11] Jahanzeb Maqbool, Sangyoon Oh, and Geoffrey C. Fox. 2015. Evaluating ARM HPC clusters for scientific workloads. *Concurrency and Computation: Practice and Experience* 27, 17 (2015), 5390–5410. https://doi.org/10.1002/cpe.3602 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.3602

[12] Simon McIntosh-Smith, James Price, Tom Deakin, and Andrei Poenaru. 2019. A performance analysis of the first generation of HPC-optimized Arm processors. *Concurrency and Computation: Practice and Experience* 31, 16 (2019), e5110. https://doi.org/10.1002/cpe.5110 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.5110 e5110 cpe.5110.

[13] Rahul Bapat Mike Bennett, Bryan Gartner. 2022. Discovering the performance and efficiency of Ampere Altra cloud-native processors for HPC workloads. In *The Arm HPC User Group (AHUG) symposium at SuperComputing (SC 2022).* https://github.com/arm-hpc-user-group/sc22-ahug-symposium/blob/53f01146b13504a99eb24db5e673ad85ead7b388/presentations/06-mbennett-ampere-perf-ahug-sc22.pdf

[14] Michael Norman, Vince Kellen, Shava Smallen, Brian DeMeulle, Shawn Strande, Ed Lazowska, Naomi Alterman, Rob Fatland, Sarah Stone, Amanda Tan, Katherine Yelick, Eric Van Dusen, and James Mitchell. 2021. CloudBank: Managed Services to Simplify Cloud Access for Computer Science Research and Education. In *Practice and Experience in Advanced Research Computing* (Boston, MA, USA) *(PEARC '21).* Association for Computing Machinery, New York, NY, USA, Article 45, 4 pages. https://doi.org/10.1145/3437359.3465586

[15] Jeffrey T. Palmer, Steven M. Gallo, Thomas R. Furlani, Matthew D. Jones, Robert L. DeLeon, Joseph P. White, Nikolay Simakov, Abani K. Patra, Jeanette Sperhac, Thomas Yearke, Ryan Rathsam, Martins Innus, Cynthia D. Cornelius, James C. Browne, William L. Barth, and Richard T. Evans. 2015. Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources. *Computing in Science & Engineering* 17, 4 (7 2015), 52–62. https://doi.org/10.1109/MCSE.2015.68

[16] Szilárd Páll, Artem Zhmurov, Paul Bauer, Mark Abraham, Magnus Lundborg, Alan Gray, Berk Hess, and Erik Lindahl. 2020. Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS. *The Journal of Chemical Physics* 153, 13 (2020), 134110. https://doi.org/10.1063/5.0018516 arXiv:https://doi.org/10.1063/5.0018516

[17] Nikolay A. Simakov, Robert L. DeLeon, Joseph P. White, Thomas R. Furlani, Martins Innus, Steven M. Gallo, Matthew D. Jones, Abani Patra, Benjamin D. Plessinger, Jeanette Sperhac, Thomas Yearke, Ryan Rathsam, and Jeffrey T. Palmer. 2016. A Quantitative Analysis of Node Sharing on HPC Clusters Using XDMoD Application Kernels. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale* (Miami, USA) *(XSEDE16).* ACM, New York, NY, USA, Article 32, 8 pages. https://doi.org/10.1145/2949550.2949553

[18] Nikolay A. Simakov, Martins D. Innus, Matthew D. Jones, Joseph P. White, Steven M. Gallo, Robert L. DeLeon, and Thomas R. Furlani. 2018. Effect of Meltdown and Spectre Patches on the Performance of HPC Applications. *CoRR* abs/1801.04329 (2018). arXiv:1801.04329 http://arxiv.org/abs/1801.04329

[19] Nikolay A Simakov, Joseph P White, Robert L DeLeon, Steven M Gallo, Matthew D Jones, Jeffrey T Palmer, Benjamin Plessinger, and Thomas R Furlani. 2018. A Workload Analysis of NSF's Innovative HPC Resources Using XDMoD. *arXiv preprint arXiv:1801.04306* (2018).

[20] Nikolay A Simakov, Joseph P White, Robert L DeLeon, Amin Ghadersohi, Thomas R Furlani, Matthew D Jones, Steven M Gallo, and Abani K Patra. 2015. Application kernels: HPC resources performance monitoring and variance analysis. *Concurrency and Computation: Practice and Experience* 27, 17 (2015), 5238–5260.