Evaluation of ARM based system for HPC workloads, a case study.

Maciej Pawlik[1]***, Klemens Noga***[1]***, Maciej Czuchry***[1]***, Jacek Budzowski***[1]***,***
Łukasz Flis[1]***, Patryk Lasoń***[1]***, Marek Magryś***[1]***, Michał Sterzel***[1]

ARM is a well known CPU architecture, up to this point, used primarily in portable devices. Application of ARM CPUs was mostly motivated by features which include low power consumption and flexible licensing model. Over the recent years ARM has grown to feature a mature software ecosystem and performance comparable to CPUs used in stationary devices like workstations and servers. According to the community, ARM has potential to address some of challenges which arose in the field of High Performance Computing. In the current day HPC, vendors and researchers alike are working on lowering power consumption, increasing performance per dollar and increasing density. This paper presents experience from a case study, which was done at ACC Cyfronet UST, where we evaluated ARM based system designed to provide computing services. The evaluation consisted of integrating the test system with a running HPC cluster and subjecting it to typical workloads. Integration allowed us to test system's compatibility with supporting services like storage and networking. Our experiences demonstrate that storage based on Lustre filesystem needs substantial changes in configuration to work with ARM servers properly. The computing performance was measured and proved to be comparable to that offered by most popular vendors. While we experienced some difficulties, it was shown that it is possible to build and run an HPC cluster based on ARM architecture.

Keywords: HPC, ARM, benchmarking, evaluation.

Introduction

Recent advancements in computing technology provided opportunities for testing performance of new architectures in HPC environments. Especially big amount of memory channels and vast memory throughput of recent ARM systems could lead to high performance in memory-bound applications. Furthermore, addition of Scalable Vector Extension (SVE) technology and other HPC focused enchantments to Armv8-A microarchitecture contribute to closing the CPU performance gap versus modern x86 systems in real HPC applications[4]. Several vendors, among them Fujitsu with A64FX chip and Cavium with ThunderX2 show that more and more market-ready solutions could be integrated by HPC vendors such as Cray, HPE, Atos, Fujitsu and Gigabyte into commercially available production systems. Due to size limitations, this abstract contains only selected results and omits some minor technical details.

1. Related work

There is ongoing work aiming to provide a fully fledged HPC system based on ARM architecture like evaluation of available ARM CPUs[6] which can give insight of progress made over the years. The most recent work done by Jackson et al.[2] focuses on evaluating the maturity of the whole ecosystem of hardware and software. Other works include the efforts to address discovered issues, some of which include: improving virtualization[1], cooperation with accelerators[5] and the need for performance[3].

[1] ACC Cyfronet UST, Krakow, Poland

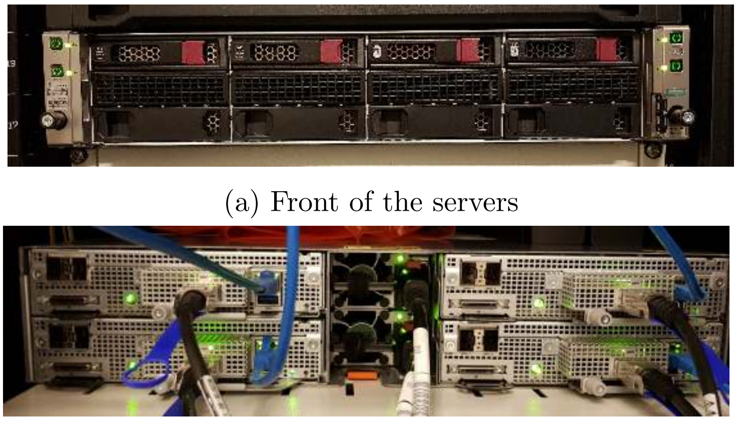
2. Test systems

Presented results are based on computations done on ARM based *test system* and Intel based *reference or production system*. We used a Apollo 70 chassis equipped with four compute nodes, each with two sockets occupied with ThunderX2 CN9980 CPUs as the test system. Single ThunderX2 CN9980 has 64 cores which run at base 2.2GHz frequency. Each node is equiped with 256GB of RAM and a dedicated network interface based on Mellanox ConnectX5 EDR. Our main compute cluster, an Apollo 8000 based on nodes containing dual Xeon E5-2680v3 CPUs clocked at 2.50GHz, 128GB of RAM and a ConnectX3 FDR Infiniband adapter, served as the reference system. In both cases network was realized as Infiniband with IP over IB extension, storage was provided as a NFS share.

ARM based systems provide unique features, which clearly separate them from common it hardware applied in HPC installations. One of such features is variable page size. The default page size for ARM is 64KB and 4KB for x86. Larger page size allows for managing larger amounts of memory, while it introduces larger overheads if allocated chunks of memory don't exceed 64KB. Additionally, page size can be adjusted based on specific needs. Additionally the ThunderX2 series provide 4-way Simultaneous Multi Threading (SMT), which results in exposing each core as having 4 separate threads. This enables greater utilization of compute pipeline as the processor can parallelize the execution and achieve greater execution unit utilization.

2.1. Integration with existing infrastructure

The test system comes in a standard 2U form factor, which allows for easy installation in a rack. Server front and back are depicted in Fig. 1.



(a) Front of the servers

(b) Back of the servers

Figure 1. Chassis of the test system.

Network adapters and connectors are identical and inter operable with those found on x86 systems. The next step was to install a dedicated operating system, in this case it was a Centos7 dedicated for aarch64, which is not an official port of the system. This might cause some compatibility problems, as it lacks the official support of Centos Project. Fortunately, the next Centos8 fully supports the ARM architecture. Mounting a lustre file system proved to be a challenge, which is still not fully addressed. During the operation Lustre client negotiates with the server, what parameters should be used to access the file system. Some of those parameters, namely message size, are based on page size, which serves as a smallest unit of data, based on assumption that both server and client share a common page size. It is possible to tune server parameters in such way, that an x86 server will cooperate with arm client, but this in turn breaks compatibility with x86 clients. At this point in time, with current stable version of Lustre, it is impossible to have a shared Lustre storage for both x86 and arm clients.

3. Evaluation results

Evaluation was performed by executing a standard test suite of applications, which are common to our production cluster. The test suite was recompiled with a tool chain dedicated for ARMv8 architecture. It was assumed that each test set should be build and run with the best set of tools available for a given platform.

One of the components included in the set is the Linpack benchmark, which gives a good approximation of available raw performance. Moreover the result can be easily predicted and the benchmark can serve as a test for basic performance.

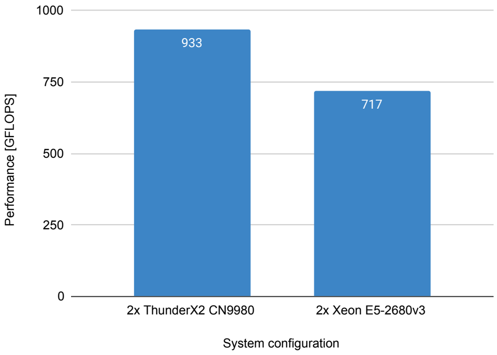


Figure 2. Linpack benchmark performance.

Table 1. Linpack results

CPU	theoretical performance	measured performance	efficiency
2x ThunderX2 CN9980	1280 GFLOPS	933 GFLOPS	73%
2x Xeon E5-2680v3	960 GFLOPS	717 GFLOPS	75%

Fig. 2 presents obtained results in form of graph while Tab. 1 contains numerical values. Results demonstrate the advantage of ThunderX2 over compared Xeon processor. This advantage is the result of core count and clock of the cores, where TuhnderX2 core is capable of performing 8 double precision operations per cycle, while Xeon's is able to perform 16 by using AVX.

The application test set contains scientific applications such as Gaussian, GROMACS, NAMD and CP2K. The latter was chosen to perform initial testing and verification of the system. As mentioned in our tests addressed overall performance, performance per watt and scalability. We focused on features which are unique to the system, where SMT is one of them. Fig. 3 presents run time of CP2K job in relation to thread count on a single test node. One can observe a clear relation of increased thread count to lower run time, with exception of the last configuration of 256 threads. This can be explained by the fact that managing and synchronization cost out weights the gains and possibly some system related processed are deprived of CPU time, which lead to overall lower performance. Overall, performance provided by Intel system

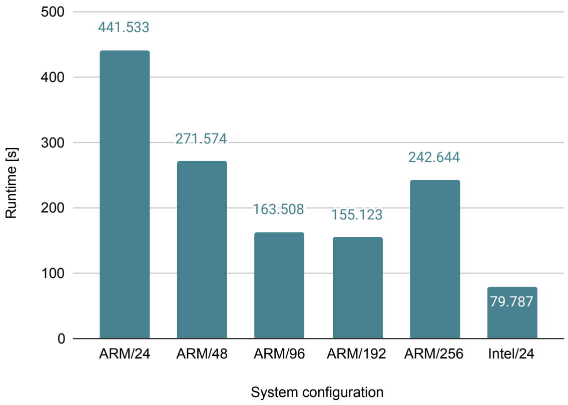


Figure 3. CP2K job run time. System configuration includes CPU vendor and thread count configuration. Intel result is supplied as reference.

is significantly higher, what can be explained that the application is lacking optimizations for ARM architecture.

4. Conclusions and Future Work

Tested Apollo70 system based on ThunderX2 CPUs demonstrated that it is capable of running scientific applications and providing acceptable performance. Discrepancies between synthetic benchmarks and efficiency obtained in real world scientific applications can be explained as difficulties with adopting scientific code to ARM architecture. Results obtained in this study can be further improved by expanding the test suite and verifying other parameters which include the power management governor, ICI Speed and base system software versions.

References

1. Dall, C., Nieh, J.: KVM/ARM: the design and implementation of the Linux ARM hypervisor. Acm Sigplan Notices 49(4), 333–348 (2014)

2. Jackson, A., Turner, A., Weiland, M., Johnson, N., Perks, O., Parsons, M.: Evaluating the ARM ecosystem for high performance computing. In: Proceedings of the Platform for Advanced Scientific Computing Conference. pp. 1–11 (2019)

3. McIntosh-Smith, S., Price, J., Deakin, T., Poenaru, A.: Comparative benchmarking of the first generation of HPC-optimised ARM processors on Isambard. Cray User Group 5 (2018)

4. McIntosh-Smith, S., Price, J., Poenaru, A., Deakin, T.: Scaling results from the first generation of ARM-based supercomputers. Cray User Group (2019)

5. Montella, R., Giunta, G., Laccetti, G., Lapegna, M., Palmieri, C., Ferraro, C., Pelliccia, V., Hong, C.H., Spence, I., Nikolopoulos, D.S.: On the virtualization of CUDA based GPU remoting on ARM and x86 machines in the GVirtuS framework. International Journal of Parallel Programming 45(5), 1142–1163 (2017)

6. Wang, W., Dey, T.: A survey on ARM Cortex A processors. Retrieved March (2011)