ARMing HEP for the future

Energy Efficiency of WLCG sites (ARM vs. x86)

Emanuele Simili[1,*], *Gordon* Stewart[1], *Samuel* Skipsey[1], *Dwayne* Spiteri[1], *Albert* Borbely[1], and *David* Britton[1]

[1]School of Physics and Astronomy, University of Glasgow
Kelvin Building, University Avenue, Glasgow, G12 8QQ, United Kingdom

Abstract. We present a case for ARM chips as an alternative to standard x86 at WLCG sites to help reduce power consumption. New measurements are presented on the performance and energy consumption of two machines (one ARM and one x86), that were otherwise similar in specification and cost. The comparison was extended to a dual socket x86 node, representative of our site. These new results include the energy-efficiency and speed of single- and multi-threaded jobs; the effect of hyper-threading; and an initial look at clock throttling as a way of shaping power-load. We observe significantly lower power consumption and often slightly better performance on the ARM machine and, noting the increased availability of ARM software builds from all LHC experiments and beyond, we plan to install a 2k-core ARM cluster at our WLCG Tier2 site at Glasgow in the summer of 2023. This will enable testing, physics-validation, and eventually an ARM production environment that will inform and influence other WLCG sites in the UK and worldwide.

1 Introduction and Methodology

Following up on our initial study [1] on power consumption in high-energy physics (HEP), we have gathered new data and investigated solutions that could reduce the operational cost of the Worldwide LHC Computing Grid (WLCG [2]).

As previously noted, ARM chips have the potential to significantly reduce the energy needs of grid sites, but are not yet widely adopted as capacity hardware by the WLCG. Meanwhile, a lot of progress has been made on software compatibility (most LHC experiments have started to compile and validate their workloads on ARM CPUs), hardware availability (many vendors offer ARM enterprise options), and benchmarking tools (new HEPscore release and widespread adoption within the WLCG). We therefore gathered new measurements to better compare performance and energy consumption of arm64 CPUs with respect to x86_64 CPUs. We also investigated frequency throttling as another possible way of reducing power use.

1.1 Available Hardware

We compared the performance and power usage of two almost identical servers with very similar price tags. These featured identical chassis housing different processors: one an Ampere Altra [3] arm64 CPU and the other an AMD EPYC [4] x86_64 CPU. The comparison

[*]e-mail: emanuele.simili@glasgow.ac.uk

was extended to a dual socket x86 compute node, representative of the Glasgow Tier2 site. This node is part of a 2U quad-node chassis, but has a similar cost per node to the machines above.

All x86_64 CPUs support multi-threading. This means that the CPU can treat one physical core as two virtual cores, effectively doubling the number of parallel threads it can execute. AMD refers to this as Simultaneous MultiThreading (SMT), while Intel calls it Hyper-Threading (HT). The term hyper-threading is commonly used as a generic term for this feature.

The detailed specifications are as follows. Both x86 machines have two name tags, which will be used in plots to identify whether hyper-threading is enabled or not.

- Single-socket AMD EPYC server (x86): **AMD48c / AMD96ht**

  CPU: AMD EPYC 7643 48C/96T @ 2.3 GHz (TDP 225 W)
  RAM: 256 GB (16 × 16 GB) DDR4 3200 MHz (~2.7 GB / thread)
  SSD: 3.84 TB Samsung PM9A3 M.2

- Single-socket Ampere Altra server (ARM): **ARM80c**

  CPU: Ampere Altra Q80-30 80C @ 3.0 GHz (TDP 210 W)
  RAM: 256 GB (16 × 16 GB) DDR4 3200 MHz (~3.2 GB / thread)
  SSD: 3.84 TB Samsung PM9A3 M.2

- Dual-socket AMD EPYC server (x86): **2×AMD32c / 2×AMD64ht**

  CPU: 2 × AMD EPYC 7513, 32C/64T @ 2.6GHz (TDP 200 W)
  RAM: 512 GB (16 × 32 GB) DDR4 3200 MHz (~4 GB / thread)
  SSD: 3.84 TB Intel S4510 SATA

1.2 Power Measurements

Power metrics were collected remotely by using the open-source utility IPMItools [5], which enables remote management of server hardware via the Intelligent Platform Management Interface (IPMI) protocol. It can provide real-time power consumption and other metrics such as temperatures and voltages. A custom script was written to export power consumption, CPU load, clock frequency, and RAM usage every five seconds to a time-stamped CSV file.

Although this method is quite reliable, we noticed that the sampling interval tends to increase by a fraction of a second every step, and these delays accumulate during the benchmarking run. More rarely, especially when CPU load nears 100%, the exporter may skip an interval completely. To mitigate these irregularities and the coarseness of the sampling, totals and averages were calculated using the trapezoidal sum rule:

The total energy E_{IPMI} is calculated as: $E_{IPMI} = \sum_{i=1}^{N} \frac{(P_i + P_{i-1})}{2} \times (t_i - t_{i-1})$ and measured in kWh. The average power is measured in Watts and calculated as: $\langle P \rangle = \frac{E_{IPMI}}{T_{tot}}$, where the denominator is the total runtime: $T_{tot} = t_N - t_0$. Sampling interval and trapezoidal sum methodology were inspired by an unrelated, yet similar, study on power usage optimisation for lattice simulations on a super-computer [6].

1.3 IPMI Validation

The IPMI power measurements were validated using external power meters connected between the server and the power source. Each server has dual-redundant power supplies and two metered plugs were available; therefore, we tested three different configurations:

- single power supply to one metered plug (1p1m), while the other supply was disconnected.

- dual power supply to one metered plug (2p1m), using a split connector.
- dual power supply to two metered plugs (2p2m), taking the sum of the readings.

We ran each server in two different regimes: idle (`sleep`) and full load (`stress`). Each measurement lasted exactly one hour and the integrated power consumption from IPMI was compared to the metered plug reading(s).

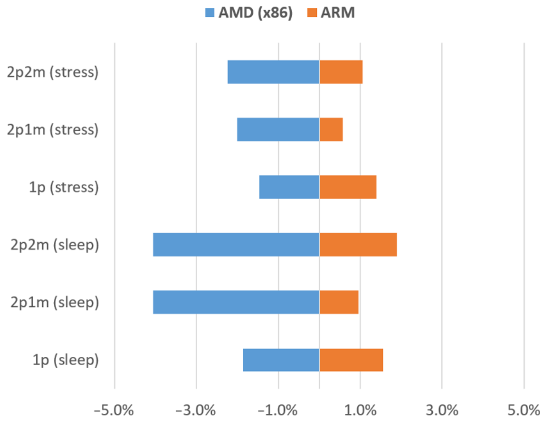

Figure 1: IPMI validation: relative difference between the total energy calculated from IPMI readings and the integrated energy usage provided by the external meter(s).

We performed this validation study on both single-socket machines (AMD96ht and ARM80c). Figure 1 shows the relative difference ($\Delta_E(\%) = \frac{E_{IPMI} - E_{plug}}{E_{IPMI}} \times 100$) between the integrated energy from remote IPMI metrics (E_{IPMI}) and the total energy reading from the metered plug (E_{plug}) at the end of the run. The measurements show a small but systematic difference between the power reported by the IPMI and that reported by the external meters, which has opposite sign on the two different test systems. On average, IPMI underestimates the energy used by the x86 (−2.6%) and overestimates the energy used by ARM (+1.2%). These systematic errors are believed to arise from unavoidable hardware differences (e.g., power loss in the motherboard, IPMI sensors calibration).

The highest relative difference we measured is ∼4% in idle and ∼2% on full load. Therefore, we take ±4% as the upper bound on systematic error for all our measurements, with the caveat that the x86 system may be using a bit more energy, and the ARM system a bit less, than indicated by the IPMI data. If true, this bias would reinforce, rather than degrade, our main conclusions presented below.

1.4 HEPscore Benchmarking Suite

The HEPscore benchmarking suite is a collection of containerised workloads from the major LHC experiments used to evaluate hardware performance for HEP. We run the latest version of the suite [7] to benchmark our servers. At the time of writing, the HEPscore suite had been officially released for both ARM and x86 with a total of seven workloads covering event generation, detector simulation, and track reconstruction tasks from experiments. The HEPscore software is now mature and stable, and is gradually being adopted as the standard benchmarking tool by WLCG sites.

The inclusion of power measurements whilst running the HEPscore benchmarking suite adds an important additional dimension to characterising hardware for the WLCG. Indeed, we propose that future hardware procurements take into consideration a combined metric of HEPscore / Watt as one of the key figures of merit. As HEPscore is directly proportional to the number of processed events per second (with some normalization constant), HEPscore / Watt is proportional to the number of processed events per unit of energy, which we can express as events / Joule or events / kWh.

2 HEPscore vs. Power

For this measurement, we ran the full HEPscore suite on all machines listed in section 1.1. The test was run twice on the two x86 machines, to measure performance with hyper-threading enabled and disabled. The ARM CPU has no such feature, and therefore can only run with its 80 physical cores (ARM80c). From the results shown in this section, we estimate that hyper-threading increases CPU performance by about 10% – 20% (one HT core is equivalent to 55% – 60% of a physical core, depending on the task).

We ran each benchmark three times and took the average score and power usage. The statistical error, calculated as the standard deviation of the three measurements, was below 1% and therefore dominated by the systematic error described in section 1.3.

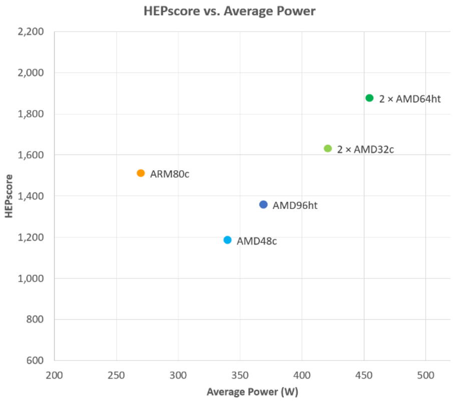

Figure 2: Comparison of the three machines (AMD, 2×AMD and ARM) on HEPscore and average power usage. Both x86 systems (i.e., those with AMD CPUs) have an extra data-point to show the performance gain when hyper-threading is enabled.

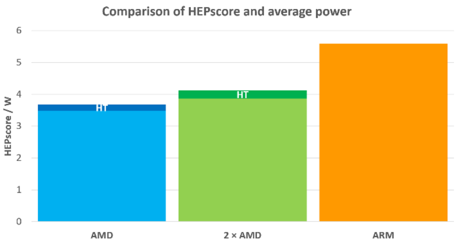

Figure 3: Comparison of the average power dissipation per unit HEPscore among the three machines (AMD, 2×AMD and ARM). The top part of the AMD bins shows the improvement in HEPscore / Watt when hyper-threading is enabled.

The plot in figure 2 compares the performance and power usage of our three machines on a 2-dimensional scale. Here we can see that our dual-socket x86 compute node has better performance than our single socket ARM server, but this comes at the cost of higher power consumption. When power dissipation is considered (figure 3), the ARM system outperforms both x86 servers in terms of HEPscore / Watt.

2.1 Thread-scan

To better characterise performance and power consumption of the multi-core CPUs, we repeated the above measurements as a function of the number of threads: we ran the HEPscore suite multiple times with a fixed number of threads per copy (either one or four, depending on the experiment) while increasing the number of copies executed simultaneously until the CPU was saturated. We ran this thread-scan on the two single socket machines: once on the ARM server (ARM80c), and twice on the AMD server to test performance with and without hyper-threading (AMD96ht and AMD48c respectively).

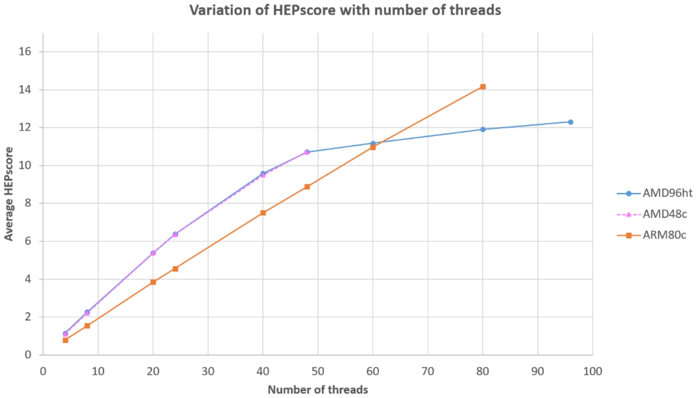

Figure 4: Thread-scan: average score with respect to the number of threads for the single socket AMD and ARM servers.

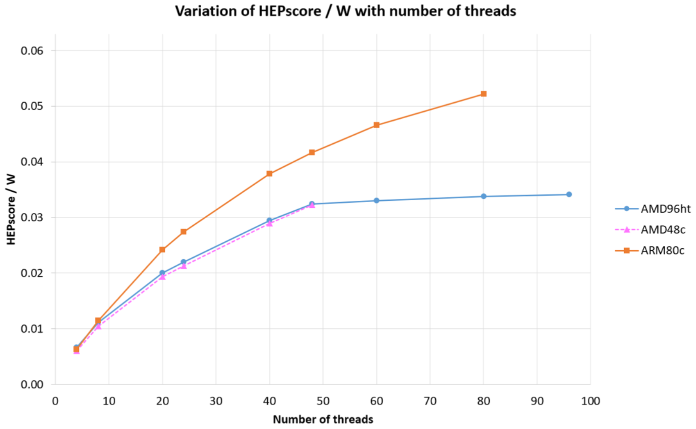

Figure 5: Thread-scan: average HEPscore / Watt with respect to the number of threads for the single socket AMD and ARM servers.

Figure 4 shows that, for a small number of threads, a single non hyper-threaded x86 core is faster than a single ARM core. Once the number of threads significantly exceeds the number of physical cores in the x86 CPU ($N_{threads} \gtrsim 60$), the greater physical core-count of the ARM CPU lends it a distinct advantage resulting in a higher overall score. When energy usage is taken into account (figure 5), the ARM server always gives the best score per unit power.

This study also shows that hyper-threading has no detrimental effect on performance when the number of threads is below the number of physical cores (the AMD96ht and AMD48c lines in figures 4 and 5 perfectly overlap). In conclusion, despite the better performance of a single physical x86 core, at full load the 80-core ARM system offers better performance and is more energy efficient than the single-socket AMD system.

2.2 Frequency Throttling

Reducing the CPU frequency could also save power, at the cost of a longer execution time. We looked at the power usage during execution of a single HEP workload for different frequency settings on the two single-socket test machines. The range of clock frequencies that can be selected varies from one model of CPU to another. One workload (CMS sim-digi) was used to avoid contaminating the measurement with the idling that happens between workloads during a full HEPscore run.

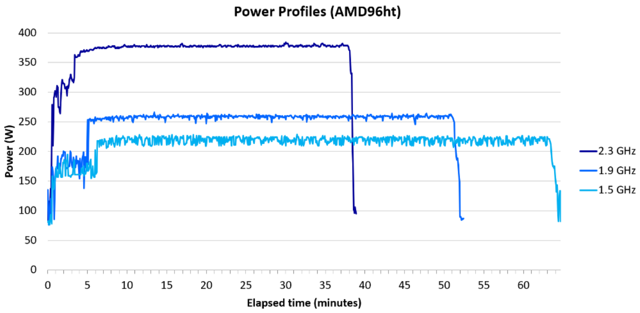

Figure 6: Frequency throttling: power profiles during the execution of a single workload for the 3 frequency settings available on the AMD machine.

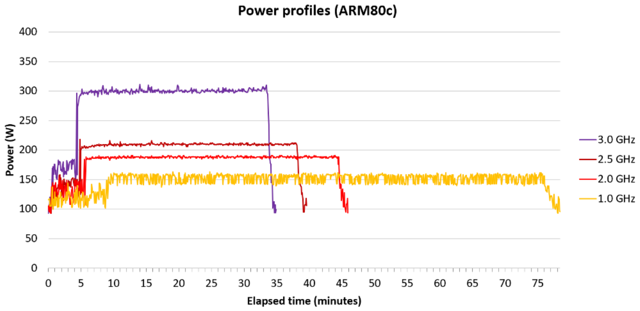

Figure 7: Frequency throttling: power profiles during the execution of a single workload for the 4 frequency settings available on the ARM machine.

The power profiles of different frequency settings for the ARM and AMD machines are shown in figure 6 and 7 respectively. From these plots we notice that, at lower frequencies, the job duration is increased but the average power usage reduced. In particular, at the maximum clock frequency the AMD machine uses about 380 W and the ARM about 300 W. Running the same job at a slightly lower frequency, the AMD uses a little over 250 W and the ARM a little over 200 W, which means in both cases a 33% reduction in energy usage, while the execution time increases of about 20% on the AMD and only 10% on the ARM.

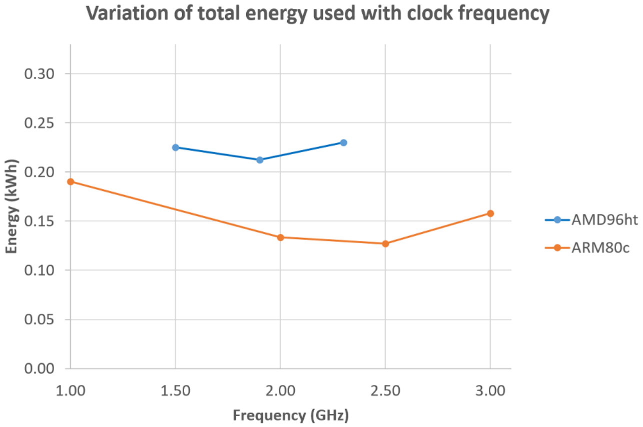

Figure 8: The total energy used for a given job vs. clock frequency shows a minimum at one step below the maximum allowed frequency on both test machines (AMD96ht and ARM80c).

By integrating the power usage, we see that the total energy (figure 8) varies modestly with clock speed and has a minimum value at an intermediate frequency on both machines. This indicates that, no matter the architecture, there is a beneficial trade-off in throughput for an overall decreased power consumption. WLCG sites can decide whether the one-third saving in energy is worth the 10-20% increase in time.

3 Conclusions and Outlook

With this study, we reinforce our previous claim on the better power efficiency of ARM servers for typical HEP workloads. With the HEPscore suite being adopted as the standard benchmarking tool by WLCG sites, it is the perfect time to begin considering energy efficiency as part of the overall performance evaluation. This will allow sites to make an informed choice about the most suitable hardware to purchase. The LHC experiments are now in the process of validating ARM builds of their software, and some non-LHC experiments such as Belle2 are also moving in this direction, demonstrating a willingness within the HEP community to support ARM resources.

3.1 Upcoming ARM farm at Glasgow

Subsequent to the conference, we took delivery of twelve dual-socket Mt. Collins servers containing Ampere Altra Q80-30 CPUs, kindly donated by Ampere Computing (USA)[1] following discussions about our work. This will mean that we will put on the grid 12 × 80 × 2 = 1,920 ARM cores. These will become part of the advertised resources of our Tier2 site; a

[1] https://amperecomputing.com/

separate batch queue will be created to expose these resources on the grid, such that they can be targeted by the experiments.

The increasing availability of ARM resources will further encourage experiments to build software appropriately in order to take advantage of them. Assuming our pilot site is successful, we plan to further expand our ARM farm. As the ARM architecture is evolving quickly, we want to investigate the various options that are becoming available, including the Ampere Altra Max and newer AmpereOne families of processors, and the NVIDIA Grace CPU.

References

[1] E. Simili et al., Power Efficiency in HEP (x86 vs. ARM), ACAT2022 submitted for publication in IOPscience Journal Of Physics: Conference Series (2023)

[2] The Worldwide LHC Computing Grid WLCG (2023), https://wlcg.web.cern.ch/

[3] Ampere™ Altra™ Multi Core Server Processors (2023), https://amperecomputing.com/processors/ampere-altra

[4] M. Mattioli, Rome to Milan, AMD Continues Its Tour of Italy, IEEE Micro **41-4**, 78-83 (2021), **doi:10.1109/MM.2021.3086541**

[5] D. Laurie, IPMItools: Intelligent Platform Management Interface (2023), https://github.com/ipmitool/ipmitool

[6] A. Portelli, Optimisation of lattice simulations energy efficiency (2022), **doi:10.5281/zenodo.7057319**

[7] D. Giordano et al., HEPiX Benchmarking Solution for WLCG Computing Resources, Comput Softw Big Sci **5**, 28 (2021)