Where are the Joules? Energy Demand Analysis of Heterogeneous Memory Technologies

Thomas Preisner
Friedrich-Alexander-Universität Erlangen-Nürnberg

Dustin Tien Nguyen
Friedrich-Alexander-Universität Erlangen-Nürnberg

Manuel Vögele
Ruhr-Universität Bochum

Matthias Szymanski
Ruhr-Universität Bochum

Timo Hönig
Ruhr-Universität Bochum

Rüdiger Kapitza
Friedrich-Alexander-Universität Erlangen-Nürnberg

Wolfgang Schröder-Preikschat
Friedrich-Alexander-Universität Erlangen-Nürnberg

Abstract

New heterogeneous memory architectures are emerging with the increasing availability of different memory technologies. The growing number of byte-addressable memory technologies (e.g., NVRAM and CXL) make the selection of components for concrete systems increasingly difficult, as each memory technology has specific advantages and disadvantages. Currently, decisions are made based on a variety of performance measurements and recommendations with respective implementation guidelines. However, traditional performance metrics often overlook the unique energy-demand dynamics of each specific memory technology, especially with diverse memory access patterns and workloads.

This paper highlights the importance of memory utilization with different technologies and their environmental impact (i.e., energy demand). To this end, we present meBench – a benchmark generator that systematically quantifies the (non-uniform) energy demand of different memory types. We reveal the relevance of the generated results by contrasting them with the assumptions made in previous research on carbon-aware memory allocation.

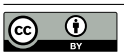
CCS Concepts

• **Computer systems organization → Heterogeneous (hybrid) systems**; • **Hardware → Power estimation and optimization**.

Keywords

Memory Technologies, Energy Consumption, Benchmarking, Heterogeneous Systems, DRAM, Non-Volatile Memory, Optane, CXL

ACM Reference Format:
Thomas Preisner, Dustin Tien Nguyen, Manuel Vögele, Matthias Szymanski, Timo Hönig, Rüdiger Kapitza, and Wolfgang Schröder-Preikschat. 2025. Where are the Joules? Energy Demand Analysis of Heterogeneous Memory Technologies. In *4th Workshop on Heterogeneous Composable and Disaggregated Systems (HCDS '25), March 30, 2025, Rotterdam, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3723851.3723857



HCDS '25, Rotterdam, Netherlands

ACM ISBN 979-8-4007-1470-2/25/03
https://doi.org/10.1145/3723851.3723857

1 Introduction

The continuing rise in global energy consumption [9, 25] is a major challenge for our environment and economy. In this context, data centers and the momentum associated with artificial intelligence (AI), in particular, contribute a non-negligible part [21] to this. According to the International Energy Agency (IEA), their share of global energy consumption currently amounts to approximately 2% and is estimated to further double by 2026 [9]. Making matters worse, data centers are also exhausting power grids regionally. At the same time, some of these grids are already limiting their growth [10]. To mitigate this, cloud providers are investing in, or have already implemented, on-site power generation [23, 27]. However, in order to solve the structural problems in the long term, we need to reduce the energy demand of computer systems and data centers.

System memory, particularly dynamic random access memory (DRAM), represents a significant factor in server infrastructure and accounts for up to 50% of total energy consumption in modern computing systems [7, 19, 32]. Modern memory technologies include several DRAM alternatives with varying properties, enabling system heterogeneity. In addition, there are new memory *types* such as non-volatile random access memory (NVRAM) [15, 31], high bandwidth memory (HBM) [22] as well as other upcoming memory technologies such as the memory-semantic solid state drives (SSDs), which uses the emerging compute express link (CXL) standard [26]. Careful planning of a system's memory composition prior to its procurement, and memory placement strategies for existing systems, reduce the energy consumption and, therefore, carbon emissions [18].

Understanding memory technologies' power consumption patterns is crucial for aiding this intricate decision-making process. However, the performance benchmarks available today [2, 11, 14] are inadequate for this purpose because they do not take into account varying loads or even idle consumption. As a result, and in order to provide tools for answering this question, we propose meBench, a tool to trace the energy behavior of different memory technologies. meBench generates minimal but configurable synthetic benchmarks and allows for selective examination of energy consumption behavior for specific memory access patterns without requiring expertise with workload internals. Instead, meBench

quantifies performance data and provides energy measurement results for individual workloads that are more practical than theoretical numbers from vendor datasheets.

The contributions of this paper are as follows: We propose meBench, a practical tool to analyze the performance and energy demand of different memory technologies. With meBench, we aim to answer the following core questions that help system designers make use of different memory types. 1.) Do performance optimizations consistently reduce energy consumption? 2.) Which memory technology consumes more energy under typical workloads? We solicit further research on systems with heterogeneous memory by supplying actual measurement data as generated with meBench.

The rest of the paper is organized as follows: Section 2 discusses related work on benchmarking different memory types. Next, Section 3 provides the background with regard to the system's architecture and the unique differences of a selection of memory technologies initially supported by our benchmarking tool. Section 4 gives further insights into design and implementation of meBench. Continuing, Section 5 discusses the results and utilizes these to refine existing theoretic research. Finally, we summarize our results in Section 6.

2 Related Work

There is only limited research addressing the topic of energy consumption of different memory technologies available for server systems. Specifically, in 2015, Vandierendonck et al. compared the energy consumption of the non-volatile spin-transfer-torque random access memory (STT-RAM) and resistive random access memory (RRAM) with DRAM and proposed an energy model to help with design decisions regarding energy efficiency [30]. In addition, they concluded that hybrid memory hierarchies can help improve energy efficiency depending on the respective memory size. However, none of the evaluated NVRAM solutions are available as of today, and their energy model is mostly theoretical and not supported by actual measurements.

Vogelsang concretely addressed DRAM's energy consumption characteristics and also proposed an energy consumption model while fully considering DRAM's inner workings [32]. However, Ghose et al. argue that, due to its complexity, DRAM's consumption cannot be modeled accurately only using the manufacturer's specifications and instead supplement their own energy model with additional measurements [7]. In this regard, they further state that the energy required for reads and writes of DRAM correlates with the data being written. Continuing, Katsaragakis et al. explicitly measured the energy consumption of Intel Optane using a selection of B+ tree indexing implementations designed for persistent memory [17]. They identify read-heavy workloads to be generally less energy-demanding than write-heavy workloads. In addition, Peng et al. and Weiland et al. both noted, that Intel Optane exhibits a high idle energy consumption in comparison to DRAM [24, 33].

Other research from the field of memory technologies mainly focuses on measuring performance characteristics and creating performance models of various available or upcoming technologies. For instance, there exist various generic benchmarks such as, e.g., Intel's Memory Latency Checker [14], Intel's Performance Counter Monitor [11] or SPEC [2]. Furthermore, Izraelevitz et al., Yang et al. as well as Zhang and Swanson analyzed basic performance characteristics of Intel Optane Persistent Memory – a byte-addressable NVRAM [16, 35, 36]. As such, they all observed the memory's bandwidth and latency under varying loads, access patterns, and operations in order to provide meaningful insights at the micro and macro level, which can then be used to purposefully optimize software for DRAM respectively Intel Optane Persistent Memory. Analogously, Friesel et al. provided a performance model for evaluating and optimizing workloads towards HBM [6].

With regard to the relatively new but uprising CXL standard, Wu et al. provided a generic performance evaluation emphasizing different memory topologies enabled by the technology [34]. In contrast, Liu et al. present SupMario, a characterization framework for systematically analyzing and modeling memory performance [20]. The work also addresses latency and throughput of CXL, which is inherited from its underlying PCIe hardware interface. In addition, they proposed a performance model allowing for performance predictions of concrete workloads in order to provide recommendations for software optimizations. Last but not least, Tang et al. also analyzed performance on concrete CXL memory expansions, effectively formulating latency, throughput, and cost models [28].

However, almost all of the aforementioned work focus on measuring the energy consumption behavior inherent to the respective technologies. As such, we currently have to mostly rely on theoretic energy models using data provided by the respective manufacturers – if available. Consequently, this paper aims to provide a workload generator to comprehensively measure the energy consumption behavior of current and emerging memory technologies.

3 Fundamentals

In this section, we will first go over relevant constraints defined by the underlying architecture that need to be regarded for the different workloads to be generated. Afterwards, we will further go into detail about the subtle differences between the memory technologies supported by our tool and how they do affect their respective energy consumption.

3.1 Architectural Specifics

With regard to our intended goal, this subsection provides some basic information about architectural specifics as a consequence of targeting x86-64 as the underlying platform. In addition to the constraints defined by the instruction set architecture (ISA), we also need to regard non-uniform memory access (NUMA) as one further potential factor influencing energy consumption in the same way it influences performance.

3.1.1 x86-64. Depending on the underlying CPU's ISA, different assembly instructions are available to perform memory accesses. Intel's x86-64 provides a multitude of instructions for different memory access granularity [12]. Specifically, Intel's x86-64 provides instructions for accessing memory in granularities of 1 (BYTE), 2 (WORD), 4 (DWORD), 8 (QWORD) and 16 (DQWORD) bytes length. With ISA extensions such as advanced vector extensions (AVX), even coarser granularities of up to 64 bytes are possible. For all these instructions, both aligned and unaligned variants exist.

In addition, there exist variants of most of these instructions that allow to load or store directly to memory, bypassing caches. Under

x86-64, they are referred to as non-temporal loads respectively stores. They allow optimizations such as preventing the loading of rarely used data to pollute the caches or, with NVRAM in mind, to directly write back to memory in order to avoid potential data losses or data inconsistencies.

x86-64 also provides the instruction `clflush` for explicitly flushing the cache line[1] associated with a given address. This instruction can be used to ensure that modified data, at any level of the cache hierarchy, is written back to memory [12].

3.1.2 Non-Uniform Memory Architecture. As a result of NUMA, memory accesses can vary in cost depending on their destination [5]. Commonly, this can be observed on multi-processor hardware where every CPU can access the entire memory whilst only being directly connected to a subset thereof. As a consequence, some memory is more local than others, affecting its latency and bandwidth. In addition, with heterogeneous memory architectures in mind, NUMA increases in relevance as different memory technologies also possess different access characteristics, which is further reinforced by the upcoming CXL standard (see Section 3.2.3) [3].

3.2 Memory Technologies

As mentioned in the introduction, there is a multitude of different memory technologies available nowadays. However, due to availability, we currently only support the generation of workloads for DRAM and Intel Optane Persistent Memory as exemplary NVRAM technology as a starting point. As such, we will briefly outline the specifics thereof. Nonetheless, we kept our tool as generic as possible to easily incorporate benchmarking of other memory types, like the emerging CXL-based memory.

3.2.1 DRAM. Being installed in virtually every server, and especially every x86-64-based system, DRAM is a commodity main memory technology. Internally, it comprises multiple matrices containing cells of capacitors and transistors. Each row is interleaved over multiple DRAM chips and ensues the granularity of read and write operations. This is typically 64 bytes and also equals the cache line size. The capacitors' electric charge represents binary 1s or 0s and, due to the inherent leakage of capacitors, require frequent refresh cycles in order to maintain their stored state. Consequently, DRAM is volatile as stored information will be lost without a continuous power supply. Furthermore, reading a row's contents is a destructive process as it entails measuring the capacitors' current electric charge by discharging them. Subsequently, a write operation is required in order to restore their previous state [7, 32].

3.2.2 Intel Optane. In contrast to DRAM, Intel's Optane Persistent Memory – specifically the 200 series – is marketed as non-volatile, byte-addressable NVRAM [15–17, 35]. It is based on the 3D-XPoint architecture and can be categorized as a new memory tier between DRAM and block-based storage. As such, it exhibits higher latencies with simultaneously greater available storage capacity than DRAM. However, due to its non-volatile nature, Intel Optane logically does not rely on periodic operations to retain its contents. Additionally, Intel Optane provides persistency guarantees by means of an asynchronous DRAM refresh (ADR) domain, which denotes that any CPU-issued write operation that reaches it will endure a power failure. 3D-XPoint has an access granularity of 256 bytes – smaller write operations are stored in a designated write-combining buffer to mitigate write amplification. Last but not least, Intel Optane offers two operation modes. On the one hand, in *Memory Mode*, it uses the existing DRAM as cache to hide access latencies whilst effectively expanding available main memory. Here, non-volatility is not granted. In the other mode, namely *App Direct*, Intel Optane is directly accessible (DAX) as persistent memory device without additional, volatile cache. This mode also allows optional interleaving memory across DIMMs for potential speedup.

3.2.3 CXL. CXL [3] is an emerging communication protocol that can be used to attach memory devices to a CPU over PCIe. CXL's requirements toward the attached memory are much more relaxed than those of traditional memory connectors: Memory may be physically placed far away from the CPU, DRAM of different generations can be attached to the same system, and the creation of entirely new classes of memory devices becomes possible. This includes devices such as Samsung's CMM-H, which is able to act as large persistent main memory and as an SSD simultaneously [26].

CXL is still in its infancy, and few CXL devices are available. However, in the future, it will allow the creation of systems utilizing highly heterogeneous memory, requiring informed decisions by system builders.

4 meBench

meBench is a tool that can generate highly configurable workloads. These workloads can then be used to independently analyze a multitude of factors and their effect on energy consumption. In the following, we outline our considerations regarding choices of configurability and benchmark design with our available hardware and compatibility with future ones in mind.

4.1 Configurability

In meBench, every workload configuration is designated to perform only one single task in order to reduce any confounding factors tainting the measurement result. In addition, the available configuration options were chosen based on the hardware nuances of the x86-64 ISA and the initially supported DRAM and NVRAM. Nonetheless, meBench is built to allow easy integration with upcoming technologies, e.g., CXL.

System Topology: With NUMA in mind, meBench requires the configuration of the system topology, including a set of usable CPU cores for at least one NUMA domain. This is necessary for creating worker threads and pinning them to specific CPU cores. Depending on the memory to be benchmarked, this topology can also be enhanced by setting explicit files to be memory-mapped. This enables support for any direct access (DAX)-capable hardware, such as Intel Optane in *App Direct* mode.

Memory Type: Selects the memory type to be stressed. Depending on its value, we allocate or map memory defined in the system topology.

Memory per Thread: Defines the amount of memory exclusively allocated per thread. Naturally, the total amount of

[1] x86-64 uses cache lines with 64 bytes granularity.

allocated memory cannot exceed the size of the installed memory (including some buffer for the operating system).

NUMA Distance: As NUMA affects performance depending on the distance, the workload can be configured to enforce the usage of either *near* or *far* accesses.

Load/Store Ratio: Defines the number of consecutive load respectively store operations that are performed repeatedly.

Access/Chunk Size: Due to the availability of different access granularities, we divide the allocated memory into chunks. This option can then be used to define how many bytes per chunk are accessed, allowing to simulate basic stride access. See Section 3.1.1 for the supported memory granularities.

Access Pattern: As memory prefetching improves the performance of sequential operations, this option enforces access to memory chunks either *sequentially* or *randomly*.

Cache Optimizations: Allows substitution of load/store operations with their non-temporal counterparts. Alternatively, this option can also be used to enforce cache line flushes after loads and stores.

Duration: Defines the runtime of a single workload execution.

4.2 Benchmark Design

Conceptually, we kept the code for the workloads minimal and modular. For this reason, it is written in C to avoid unexpected performance penalties as a result of abstractions. The configuration for a specific workload is supplied as a C header file and validated at compile time via C preprocessor macros. Furthermore, the use of preprocessor macros prevents the introduction of additional runtime overhead.

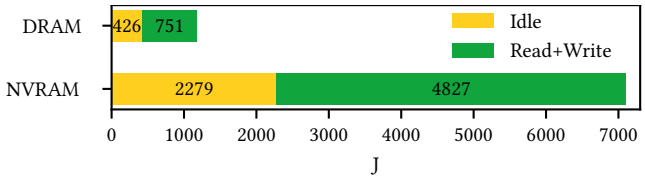
Payload contents for writes are randomized to prevent biases concerning the distribution of stored 0s and 1s. For this reason, meBench employs *drandr48_r*, a simple pseudorandom generator, initialized with a constant, but configurable seed to maintain reproducibility. This random generator is then used to initialize further thread-local random generators to ensure deterministic payloads and *random* access patterns (if configured).

Last but not least, to enable comparability of results over a set of executed workloads, they return the total number of accessed bytes upon completion.

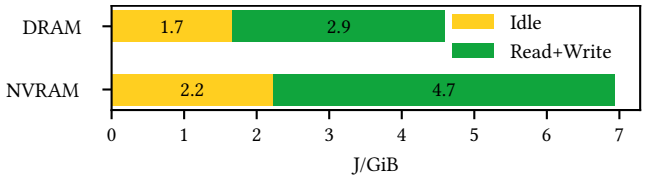
5 Evaluation

We executed a multitude of workloads generated with meBench on an NVRAM-equipped server system running a minimal Debian 12 installation with a 6.1.0 Linux kernel. The system is equipped with two Intel Xeon Gold 6330 processors, each with 28 cores and hyper-threading enabled for a total of 112 logical cores. It contains 256 GB DRAM, consisting of 8 32 GB DDR4 DIMMs. In addition, our system also contains 1024 GB of NVRAM, composed of 8 128 GB Intel Optane Persistent Memory 200 DIMMs. The NVRAM is configured in *App Direct* mode with interleave enabled to decouple its usage from DRAM[2]. We utilize Intel's running average power limit (RAPL) [13] in conjunction with PERF[3] to take energy measurements. It is capable of gauging the energy consumption of the installed memory with some limitations with regard to its accuracy [1, 4], but is sufficient for an initial assessment of meBench.

However, we estimate that future hardware will further improve in accuracy. Furthermore, we turned off all background services to prevent any potential noise from distorting our results. For the following measurements, unless otherwise noted, we always employ both CPUs with 16 threads each for mixed operations without cache optimizations whilst exclusively using near memory and 16-byte access and chunk sizes.



(a) Total energy consumptions per memory technology.



(b) Energy consumption per installed GiB.

Figure 1: Comparison of energy consumption in idle and during an exemplary workload specifically caused by DRAM respectively NVRAM.

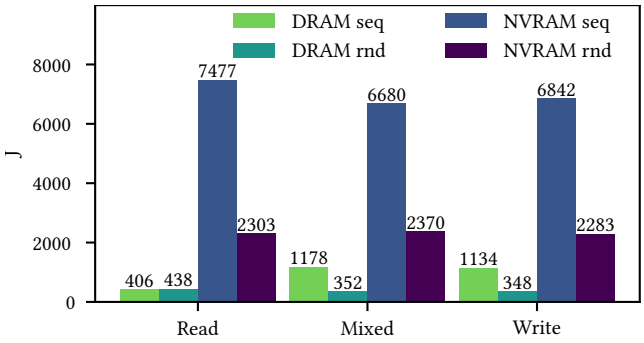
5.1 Idle Measurements

Initially, we measured our system's idle energy consumption with and without NVRAM being installed multiple times over 80 s to determine respective baselines. With these values, we are able to distinguish the actual energy consumption of each technology in idle and under load, as visualized in Figure 1a. Given the installed memory capacity of each technology, this results in an effective consumption of 1.7 J/GiB for DRAM and 2.2 J/GiB for NVRAM (cf. Figure 1b). Here, it is evident that Intel Optane consumes significantly more energy per GiB than DRAM, both at rest and even more so under load. This observation matches earlier findings by Katsaragakis et al. [17].

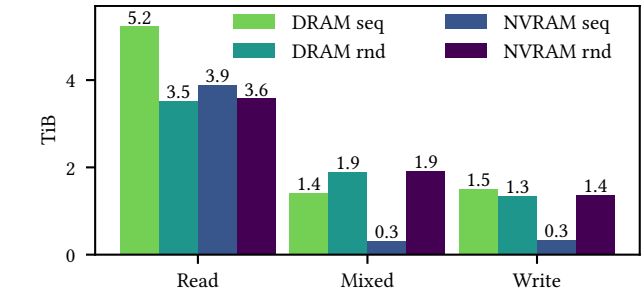
5.2 Access Type and Patterns

Regarding the ratio between read and write operations in conjunction with either sequential or random accesses to the respective memory, we notice that – as before – DRAM requires several magnitudes less energy than NVRAM, as visible in Figure 2a. In parts, this also correlates with Intel Optane's worse performance, resulting in fewer bytes written during the benchmarks' executions (cf. Figure 2b). Interestingly, this performance penalty is especially severe for sequential mixed and write-only workloads. Consequently, our NVRAM generally requires more energy per memory access,

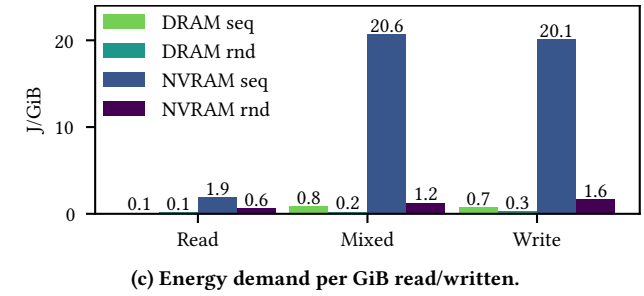
[2]Due to limitations of x86-64 and firmware, NVRAM can not be used without having an identical number of DRAM DIMMs installed.
[3]https://perfwiki.github.io/main/



(a) Distinctive energy consumption.



(b) Number of TiB accessed.



(c) Energy demand per GiB read/written.

Figure 2: Comparison between DRAM and NVRAM depending on access type and access pattern.

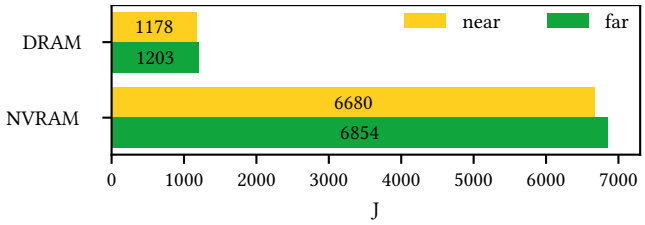
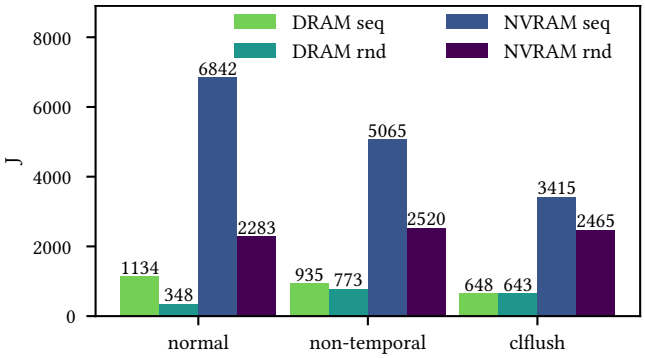
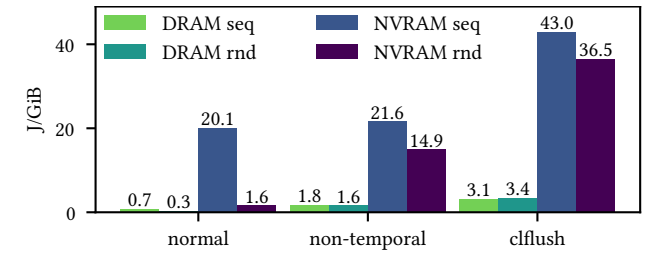


Figure 3: Comparison of energy consumption of DRAM and NVRAM between near and far NUMAs.



(a) Distinctive energy consumption.



(b) Energy demand per GiB written.

Figure 4: Comparison of energy consumptions as a result of optionally flushing or skipping caches during writes.

as shown in Figure 2c. However, we must note that read-only and generally random accesses are approximately in the same range.

5.3 NUMA Distance

As expected, NUMA does not significantly affect the energy demand for DRAM and NVRAM as seen in Figure 3. However, please note that we are only observing the raw consumption of the memory itself; the CPU may still be penalized due to the incurred slowdown.

5.4 Cache Optimizations

Since neither DRAM nor Intel Optane supports non-temporal loads, we are only examining write-only workloads for now. As in the previous comparison, the influence of sequential and random access patterns on the total energy consumption can be observed. At first glance, sequential accesses are generally more expensive for normal and non-temporal stores than for cache line flushes. Similarly, random accesses seem more taxing in combination with non-temporal stores (cf. Figure 4a). However, looking at the energy demand in relation to the written data, as in Figure 4b, normal accesses are the most and cache line flushes are the least efficient. For normal, random accesses, Intel Optane approaches the efficiency of DRAM. In general, uncached write accesses are predictably more taxing and ensuring consistent non-volatility costly.

5.5 Case Study: Carbon-Aware Allocations

In their SIGENERGY publication, Köhler et al. demonstrate how memory placement decisions can influence the carbon emissions of a given system [18]. Due to the unavailability of accurate power usage data from measurements or data sheets, the authors use

estimated numbers in their case study. In this section, we will verify the assumptions made in that work for DRAM and Optane[4] to highlight the importance of measuring the actual energy usage on the target system prior to any decision-making.

The work assumes DRAM consumes 0.4 W/GB[5], independently of how it is being used. This assumption does not hold, as we have shown that writing into DRAM – the operation used in the work – can consume more energy than reading or idling. Additionally, our measurements indicate that they overestimate the energy consumption quite significantly, which, in reality, is only ≈0.02 W/GB.

The assumptions for Optane need to be corrected, too. The work assumes that Optane only consumes negligible amounts of energy while idle. However, our measurements reveal that its power consumption is quite significant. Consequently, for calculating the carbon emissions of Optane, it is not sufficient to only take the number of bytes written into account, as done in the work. Instead, the lifetime of an allocation must factor into the calculation to account for the high idle power usage of Optane. Additionally, the energy used by a write operation cannot be represented by a flat cost per GB written. Our data shows that Optane's energy usage for writing heavily depends on the write operation's access pattern.

Repeating the calculations of Köhler et al. leads to a significantly different result: While the CO_2 emissions reported for DRAM stay roughly in the same ballpark, the incorrect assumptions about Optane's idle energy consumption lead to a massive underestimation of the carbon emissions by Optane. With the number we have measured, DRAM outperforms Optane in all workloads presented in their work.

6 Conclusion and Future Work

Heterogeneous memory architectures present us with a multitude of challenges. Both the vast spectrum of options and the inherent environmental implications of each technology complicate the selection of efficient hardware. To provide vital information for the selection process, we present meBench, a synthetic workload generator to benchmark non-uniform energy demands systematically. We demonstrate the fine-granular insights meBench provides by benchmarking diverse workloads on DRAM and Optane. The results show that Optane, Intel's NVRAM implementation, consumes non-negligible amounts of energy, even while idling. Consequently, the assumptions used in previous research do not hold and need to be reiterated.

Initially, meBench only provides built-in support for DRAM and DAX-capable memory such as Optane. However, the improving availability of CXL may necessitate the addition of support for new memory classes, such as CMM-H, which is facilitated by meBench's modular design.

Given the measurement inaccuracies of RAPL on current hardware, we are planning to extend meBench to support other interesting architectures, e.g., Apple's ARM-based CPUs [8]. Furthermore, we intend to implement more access patterns, e.g., reverse sequential access or gather-scatter.

Complementing the focused benchmarks provided by meBench, additional work needs to be done on benchmarks investigating heterogeneous systems in their entirety, in order to provide the best insights possible to decision-makers. Moreover, the significantly increased energy demand observed during sequential memory accesses requires further root cause analysis.

Finally, future research should also factor in embodied carbon with regard to memory technologies to gauge their respective environmental impact.

Availability

The source code is available under open source license on GitHub at https://github.com/i4/meBench.

Acknowledgments

The author order corresponds to the SDC (sequence determines credit) model: that is, the n th author receives the $1/n$ th share of the credit of the whole impact [29]. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project numbers 465958100, 501993201 and 502228341 and by the Bundesministerium für Bildung und Forschung (BMBF) with project SUSTAINET-inNOvAte 16KIS2262.

References

[1] Lukas Alt, Anara Kozhokanova, Thomas Ilsche, Christian Terboven, and Matthias S. Mueller. 2024. An Experimental Setup to Evaluate RAPL Energy Counters for Heterogeneous Memory. In *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering* (London, United Kingdom) *(ICPE '24)*. Association for Computing Machinery, New York, NY, USA, 71–82. https://doi.org/10.1145/3629526.3645052

[2] James Bucek, Klaus-Dieter Lange, and Jóakim v. Kistowski. 2018. SPEC CPU2017: Next-Generation Compute Benchmark. In *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering* (Berlin, Germany) *(ICPE '18)*. Association for Computing Machinery, New York, NY, USA, 41–42. https://doi.org/10.1145/3185768.3185771

[3] Inc. Compute Express Link Consortium. 2024. *Compute Express Link™ (CXL™) – Specification.* Revision 3.2, Version 1.0.

[4] Spencer Desrochers, Chad Paradis, and Vincent M. Weaver. 2016. A Validation of DRAM RAPL Power Measurements. In *Proceedings of the Second International Symposium on Memory Systems* (Alexandria, VA, USA) *(MEMSYS '16)*. Association for Computing Machinery, New York, NY, USA, 455–470. https://doi.org/10.1145/2989081.2989088

[5] Ulrich Drepper. 2007. Memory part 4: NUMA support. (17 Oct 2007). https://lwn.net/Articles/254445/ Last visited: 2025-02-04.

[6] Birte Friesel, Marcel Lütke Dreimann, and Olaf Spinczyk. 2024. Performance Models for Task-based Scheduling with Disruptive Memory Technologies. In *Proceedings of the 2nd Workshop on Disruptive Memory Systems* (Austin, TX, USA) *(DIMES '24)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3698783.3699376

[7] Saugata Ghose, Abdullah Giray Yaglikçi, Raghav Gupta, Donghyuk Lee, Kais Kudrolli, William X. Liu, Hasan Hassan, Kevin K. Chang, Niladrish Chatterjee, Aditya Agrawal, Mike O'Connor, and Onur Mutlu. 2018. What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study. *Proc. ACM Meas. Anal. Comput. Syst.* 2, 3, Article 38 (Dec. 2018), 41 pages. https://doi.org/10.1145/3224419

[8] Paul Hübner, Andong Hu, Ivy Peng, and Stefano Markidis. 2025. Apple vs. Oranges: Evaluating the Apple Silicon M-Series SoCs for HPC Performance and Efficiency. arXiv:2502.05317 [cs.AR] https://arxiv.org/abs/2502.05317

[9] IEA. 2024. Electricity 2024. https://www.iea.org/reports/electricity-2024 Licence: CC BY 4.0.

[10] IEA. 2024. Electricity Mid-Year Update - July 2024. https://www.iea.org/reports/electricity-mid-year-update-july-2024 Licence: CC BY 4.0.

[11] Intel. 2022. Intel™ Performance Counter Monitor - A Better Way to Measure CPU Utilization. https://www.intel.com/content/www/us/en/developer/articles/tool/performance-counter-monitor.html. Last visited: 2025-02-28.

[12] Intel®. 2024. *Intel® 64 and IA-32 Architectures Software Developer's Manual.* Volume 2 (2A, 2B, 2C, & 2D): Instruction Set Reference, A-Z.

[4] We cannot verify the data for Viking NVDIMMs, as we do not have access to those DIMMs.

[5] We use GB in this section as that is the unit used by Köhler et al.

[13] Intel®. 2024. *Intel® 64 and IA-32 Architectures Software Developer's Manual*. Volume 3 (3A, 3B, 3C, & 3D): System Programming Guide, Part.

[14] Intel. 2024. Intel™ Memory Latency Checker. https://www.intel.com/content/www/us/en/developer/articles/tool/intelr-memory-latency-checker.html. Last visited: 2025-02-28.

[15] Intel Corporation. 2022. Achieve Greater Insight from Your Data. https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/optane-persistent-memory-200-series-brief.html. Last visited: 2025-01-31.

[16] Joseph Izraelevitz, Jian Yang, Lu Zhang, Juno Kim, Xiao Liu, Amirsaman Memaripour, Yun Joon Soh, Zixuan Wang, Yi Xu, Subramanya R Dulloor, et al. 2019. Basic Performance Measurements of the Intel Optane DC Persistent Memory Module. *arXiv preprint arXiv:1903.05714* (2019).

[17] Manolis Katsaragakis, Christos Baloukas, Lazaros Papadopoulos, Verena Kantere, Francky Catthoor, and Dimitrios Soudris. 2022. Energy Consumption Evaluation of Optane DC Persistent Memory for Indexing Data Structures. In *2022 IEEE 29th International Conference on High Performance Computing, Data, and Analytics (HiPC)*. 75–84. https://doi.org/10.1109/HiPC56025.2022.00022

[18] Sven Köhler, Benedict Herzog, Henriette Herzog, Manuel Vögele, Lukas Wenzel, Andreas Polze, and Timo Hönig. 2024. Carbon-Aware Memory Placement. *SIGENERGY Energy Inform. Rev.* 4, 3 (Sept. 2024), 39–45. https://doi.org/10.1145/3698365.3698372

[19] Yanjing Li, Onur Mutlu, Donald S. Gardner, and Subhasish Mitra. 2010. Concurrent autonomous self-test for uncore components in system-on-chips. In *Proceedings of the 28th VLSI Test Symposium (VTS'10)*. 232–237.

[20] Jinshu Liu, Hamid Hadian, Hanchen Xu, Daniel S. Berger, and Huaicheng Li. 2024. Dissecting CXL Memory Performance at Scale: Analysis, Modeling, and Optimization. arXiv:2409.14317 [cs.OS] https://arxiv.org/abs/2409.14317

[21] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2023. Estimating the carbon footprint of BLOOM, a 176B parameter language model. 24, 1, Article 253 (Jan. 2023), 15 pages.

[22] Micron. 2021. HBM3E – The industry's fastest, highest-capacity high-bandwidth memory (HBM) to advance generative AI innovation. https://www.micron.com/products/memory/hbm/hbm3e. Last visited: 2025-01-31.

[23] Sebastian Moss. 2024. Three Mile Island nuclear power plant to return as Microsoft signs 20-year, 835MW AI data center PPA. (20 September 2024). https://www.datacenterdynamics.com/en/news/three-mile-island-nuclear-power-plant-to-return-as-microsoft-signs-20-year-835mw-ai-data-center-ppa/ Last visited: 2025-01-25.

[24] Ivy B. Peng, Maya B. Gokhale, and Eric W. Green. 2019. System Evaluation of the Intel Optane Byte-addressable NVM. In *Proceedings of the International Symposium on Memory Systems* (Washington, District of Columbia, USA) *(MEMSYS '19)*. Association for Computing Machinery, New York, NY, USA, 304–315. https://doi.org/10.1145/3357526.3357568

[25] Hannah Ritchie, Pablo Rosado, and Max Roser. 2020. Energy Production and Consumption. *Our World in Data* (2020). https://ourworldindata.org/energy-production-consumption.

[26] Samsung. 2024. Samsung CXL Solutions - CMM-H. https://semiconductor.samsung.com/news-events/tech-blog/samsung-cxl-solutions-cmm-h/. Last visited: 2025-01-31.

[27] Zachary Skidmore. 2024. AWS reaffirms commitment to nuclear data center despite regulatory rejection. (4 November 2024). https://www.datacenterdynamics.com/en/news/aws-reaffirms-commitment-to-nuclear-data-center-despite-regulatory-rejection/ Last visited: 2025-01-25.

[28] Yupeng Tang, Ping Zhou, Wenhui Zhang, Henry Hu, Qirui Yang, Hao Xiang, Tongping Liu, Jiaxin Shan, Ruoyun Huang, Cheng Zhao, et al. 2024. Exploring Performance and Cost Optimization with ASIC-Based CXL Memory. In *Proceedings of the Nineteenth European Conference on Computer Systems*. 818–833.

[29] Teja Tscharntke, Michael E Hochberg, Tatyana A Rand, Vincent H Resh, and Jochen Krauss. 2007. Author sequence and credit for contributions in multiauthored publications. (2007). https://doi.org/10.1371/journal.pbio.0050018

[30] Hans Vandierendonck, Ahmad Hassan, and Dimitrios S. Nikolopoulos. 2015. On the Energy-Efficiency of Byte-Addressable Non-Volatile Memory. *IEEE Computer Architecture Letters* 14, 2 (2015), 144–147. https://doi.org/10.1109/LCA.2014.2355195

[31] viking. 2021. Product Brief: DDR4 NVDIMM-N. https://www.vikingtechnology.com/wp-content/uploads/2021/05/VikingTechnology_NVDIMM_ProductBrief.pdf. Last visited: 2025-01-31.

[32] Thomas Vogelsang. 2010. Understanding the Energy Consumption of Dynamic Random Access Memories. In *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. 363–374. https://doi.org/10.1109/MICRO.2010.42

[33] Michèle Weiland, Holger Brunst, Tiago Quintino, Nick Johnson, Olivier Iffrig, Simon Smart, Christian Herold, Antonino Bonanni, Adrian Jackson, and Mark Parsons. 2019. An Early Evaluation of Intel's Optane DC Persistent Memory Module and its Impact on High-Performance Scientific Applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, Colorado) *(SC '19)*. Association for Computing Machinery, New York, NY, USA, Article 76, 19 pages. https://doi.org/10.1145/3295500.3356159

[34] Jianbo Wu, Jie Liu, Gokcen Kestor, Roberto Gioiosa, Dong Li, and Andres Marquez. 2024. Performance Study of CXL Memory Topology. In *Proceedings of the International Symposium on Memory Systems*.

[35] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steve Swanson. 2020. An Empirical Guide to the Behavior and Use of Scalable Persistent Memory. In *18th USENIX Conference on File and Storage Technologies (FAST 20)*. 169–182.

[36] Yiying Zhang and Steven Swanson. 2015. A Study of Application Performance with Non-Volatile Main Memory. In *2015 31st Symposium on Mass Storage Systems and Technologies (MSST)*. 1–10. https://doi.org/10.1109/MSST.2015.7208275