COMPUTER ARCHITECTURE LETTERS

On The Energy-Efficiency of Byte-Addressable Non-Volatile Memory

Hans Vandierendonck, *Senior Member, IEEE*
Queen's University Belfast
E-mail: h.vandierendonck@qub.ac.uk

Ahmad Hassan
SAP Belfast
E-mail: ahmad.hassan@sap.com

Dimitrios S. Nikolopoulos, *Senior Member, IEEE*
Queen's University Belfast
E-mail: d.nikolopoulos@qub.ac.uk

Abstract—Non-Volatile Memory (NVM) technology holds promise to replace SRAM and DRAM at various levels of the memory hierarchy. The interest in NVM is motivated by the difficulty faced in scaling DRAM beyond 22 nm and, long-term, lower cost per bit. While offering higher density and negligible static power (leakage and refresh), NVM suffers increased latency and energy per memory access. This paper develops energy and performance models of memory systems and applies them to understand the energy-efficiency of replacing or complementing DRAM with NVM. Our analysis focusses on the application of NVM in main memory. We demonstrate that NVM such as STT-RAM and RRAM is energy-efficient for memory sizes commonly employed in servers and high-end workstations, but PCM is not. Furthermore, the model is well suited to quickly evaluate the impact of changes to the model parameters, which may be achieved through optimization of the memory architecture, and to determine the key parameters that impact system-level energy and performance.

1 Introduction

One of the advantages that byte-addressable non-volatile memories (NVM) bring to system design is their close-to-zero leakage power and absence of refresh power. These two types of loss will be jointly referred to as static power. Compared to current state-of-the-art DRAM technology, however, NVM incurs a higher to substantially higher dynamic energy as well as increased latency. This trend holds across most of the NVM contenders, and is valid for at least phase-change memory (PCM) and resistive memory (RRAM), while spin transfer torque memory (STT-RAM) may be faster than DRAM [3].

This paper explores the static vs. dynamic energy trade-off of byte-addressable NVM when used as a DRAM replacement or complement. Our approach uses first-order analytical performance and power models in order to make a number of observations on this trade-off which are generally not expressed in the mainstream computer architecture literature.

2 Methodology

We aim to model memory technologies using a small set of parameters that characterise access latency, access energy and background energy. While memory has many distinct parameters, such as precharge energy, array and row buffer read/write energy and latency, etc., we summarise those parameters in 3 key values:

- Average access latency (L, [CPU cycles]), taking into account statistics for read/write distribution and open page hit rate.
- Dynamic energy (E_d, [J]), the energy consumed on average when accessing memory, again representing the common case access.
- Static energy (P_s, [W/GB]), the background power consumed per GB of memory capacity. Background power is assumed constant over time and includes the refresh energy for DRAM.

Manuscript received May 26, 2014; revised July 20, 2014.

2.1 Performance and Energy Models

We assume a first order processor performance model that distinguishes between a baseline *cycles per instruction* (CPI_0), which is increased by a constant amount for every memory access made. We assume that the processor executes in total N instructions, and requires M memory accesses *per instruction*. The performance model is thus:

$$T(L) = \frac{N}{\phi}(CPI_0 + ML) \tag{1}$$

Here, ϕ is the CPU clock frequency. The average memory access latency L comprises demand fetches and cache line writebacks. We assume that 67% of memory accesses are reads in main memory, following industry practice [10]. Our model furthermore assumes that processes are latency-bound, a situation that appears to be true for modern server workloads [7]. Moreover, we assume parallelism in the memory system is increased to offset technological restrictions of NVM.

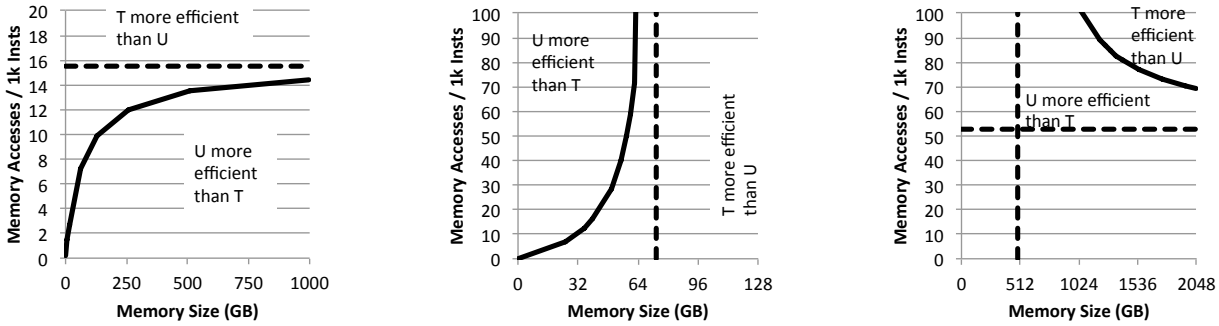
Our energy model includes dynamic energy ($E_{d,mem}$) of the memory, static power per GB of memory ($P_{s,mem}$) and processor power consumption (P_{cpu}).

$$E_{mem} = E_{d,mem}NM + (P_{s,mem}S + P_{cpu})T(L) \tag{2}$$

The parameter S represents the main memory size. Processor power consumption is modelled as a constant, the value of which depends on the processor, the program and on the ability of the memory to supply data to the processor. We are thus assuming that any performance degradation caused by NVM will impact CPU energy through an increase in execution time ($T(L)$) but not through a change in P_{cpu}.

2.2 Comparative Energy Model

The comparative energy model contrasts the energy consumption when using a memory technology T versus when using technology U. We vary T and U over DRAM, PCM, STT-RAM and RRAM in this paper.



(a) $M^\infty < 0, S^\infty > 0, \Delta E_s < 0$ (b) $M^\infty > 0, S^\infty < 0, \Delta E_s < 0$ (c) $M^\infty > 0, S^\infty > 0, \Delta E_s < 0$

Fig. 1. Possible trade-offs between memory technologies. The solid line describes workloads where $\Delta E = 0$. Depending on the technology parameters and on workload behavior, either U or T has lower energy. The dashed lines are the asymptotes S^∞ and M^∞. There is a 4th trivial scenario, where one technology outperforms another on all energy and performance metrics.

Assuming main memory is comprised either entirely of technology T or entirely of technology U, the energy difference $\Delta E = E_T - E_U$ is equal to:

$$\Delta E = \frac{N}{\phi}(\phi \Delta E_d M + CPI_0 \Delta P_s S + \Delta E_s M S + P_{cpu} M \Delta L) \quad (3)$$

where $\Delta E_d = E_{d,T} - E_{d,U}$, the difference in dynamic energy per memory access, $\Delta P_s = P_{s,T} - P_{s,U}$, the difference in static energy per gigabyte, and $\Delta E_s = P_{s,T} L_T - P_{s,U} L_U$, the difference in static energy attributable to the difference in memory access time.

We use the energy equation to characterize the workloads for which either technology T or U consumes less energy. We characterize workloads by two key properties: M, the memory access rate of the workload, and S, the memory footprint of the workload. Each (hypothetical) workload is depicted in a 2-dimensional space by assigning it to the point in the space where its X-coordinate corresponds to the value of S and its Y-coordinate corresponds to the value of M. Then, for each coordinate, we evaluate ΔE. For some workloads (values of M and S), ΔE is positive, while for other workloads it is negative.

We identify the boundary between workloads that favour one of the technologies T and U by solving the equation $\Delta E = 0$. There exist three non-trivial types of solutions to this equation, as depicted in Figure 1. The governing situation depends on the values of the energy and latency parameters.

Interestingly, in all three scenarios the boundary $\Delta E = 0$ is governed by its asymptotic behavior for infinitely large M and S. We calculate these asymptotes as:

$$M^\infty = -CPI_0 \Delta P_s / \Delta E_s \quad (4)$$

$$S^\infty = -(\phi \Delta E_d + P_{cpu} \Delta L) / \Delta E_s \quad (5)$$

The signs of the differences of the model parameters determine the signs of M^∞ and S^∞, which correlates with the three scenarios indicated in Figure 1. A fourth scenario is possible if all differences carry the same sign, in which case one memory technology trivially outperforms the other.

The scenarios will occur under different conditions. The first scenario (Figure 1(a)) occurs when ΔP_s and ΔE_s have the same sign (as $M^\infty < 0$), i.e., the static energy consumed by one technology is less than the other's and increases in access delay in the first technology do not offset the static energy gain. Moreover, the sign of S^∞ is positive, implying that the technology with less static energy has higher dynamic energy and delay. This is the common scenario when comparing byte-addressable non-volatile memories against DRAM.

The second and third scenarios (Figure 1(b), (c)) occur when the reduction in static energy is offset by increased access latency (ΔP_s and ΔE_s have different signs). Moreover, the increase in latency may be coupled with increase in memory access energy (Figure 1(b)), or not (Figure 1(c)).

The presence of the S^∞ threshold in scenario I may seem counterintuitive: when increasing the memory access rate while keeping memory size constant, shouldn't U consume more energy than T at some point? This is not necessarily the case, as increasing the memory access rate M increases the execution time (Equation 1), and thus also increases static energy. The model predicts that for particular values of dynamic and static energy, there is a situation where static energy grows faster with increasing memory access rate than dynamic energy does. Alternatively, one may restructure Equation 3:

$$\Delta E = \frac{N}{\phi}(\Delta E_s \, M \, (S - S^\infty) + CPI_0 \, \Delta P_s \, S + P_{cpu} \, \Delta L) \quad (6)$$

Thus, the sign of M in ΔE depends on whether $S > S^\infty$.

2.3 Energy-Delay Product

Similar observations can be made for the energy-delay product (EDP) as for energy, although now ΔEDP is a quadratic function of M:

$$\begin{aligned} \Delta EDP &= \frac{N^2}{\phi^2}(\Delta P_s \, S \, CPI_0^2 \\ &+ (\phi \, \Delta E_d + 2\Delta E_s \, S + 2P_{cpu} \, \Delta L) CPI_0 \, M \\ &+ (\phi \Delta EDP_d + \Delta EDP_s \, S + P_{cpu} \, \Delta^2 L) M^2) \end{aligned}$$

Here, $\Delta^2 L = L_{dram}^2 - L_{nvm}^2$, $\Delta EDP_d = E_{d,dram} L_{dram} - E_{d,nvm} L_{nvm}$ and $\Delta EDP_s = P_{s,dram} L_{dram}^2 - P_{s,nvm} L_{nvm}^2$.

We find four scenarios that are qualitatively the same as for energy, with thresholds:

$$M^\infty = -CPI_0(\Delta E_s \pm \sqrt{P_{s,dram} P_{s,nvm}} \Delta L) / \Delta EDP_s$$

$$S^\infty = -(\phi \, \Delta EDP_d + P_{cpu} \, \Delta^2 L) / \Delta EDP_s$$

M^∞ potentially has two values. Only positive values matter.

3 Comparison of DRAM vs. NVM

We compare DRAM and NVM using the methodology outlined above. Model parameter values for non-volatile memory

TABLE 1
Model parameters for non-volatile memory technologies.

	DRAM	PCM	STT-RAM	RRAM
E_d [J]	1.56e-8	3.69e-7	2.25e-7	2.39e-8
P_s [W/GB]	3.66e-1	3.66e-3	4.88e-2	7.31e-3
L [cy]	45.5	165	60.6	61.9
ΔE_d [J]	-	-3.54e-7	-2.09e-7	-8.25e-9
ΔP_s [W/GB]	-	3.62e-1	3.17e-1	3.58e-1
ΔE_s [J/GB]	-	1.60e+1	1.37e+1	1.62e+1
ΔL [cy]	-	-1.19e+2	-1.52e+1	-1.65e+1
ΔEDP_d [Js]	-	-6.02e-5	-1.29e-5	-7.68e-7
ΔEDP_s [Js]	-	6.56e+2	5.76e+2	1.77e+3
	Energy comparison			
S^{∞} [GB]	-	270.0	65.4	31.6
	EDP comparison			
S^{∞} [GB]	-	1340.2	130.9	75.1

technologies are indicative of how these future technologies may perform (see Table 1). The values are consistent with the literature on PCM [5], [9], STT-RAM [11] and RRAM [9]. Static PCM energy is rated at 1% of DRAM static energy [2]. We further assume a 2.1 GHz CPU clock, a baseline $CPI_0 = 1$ and a uniform 30 W CPU power consumption. The latter is based on the observations that CPU power may be approximated by linearly interpolating between idle and peak power using CPU load [10], that CPU load on servers is typically small, and that CPU idle power ratings are in the 10-20 W range [4], [8]. The memory bus is clocked at 800 MHz and is activated on both clock edges. The memory is organized in multiple banks and channels in order not to be bandwidth bound for the application domain where the above assumptions hold.

We validated the performance and energy model by executing a micro-benchmark with configurable MPKI. The benchmark uses pointer-chasing code through a fixed-size array to generate a predictable number of cache misses. The MPKI is modified by changing the array access pattern. The benchmark is simulated on a cycle-accurate processor simulator consisting of GEM5 [1] and the DRAMSim2 [13] main memory simulator. We modified DRAMSim2 to achieve the desired characteristics of PCM storage-class memory which include zero refresh power, low background power, higher dynamic energy and higher latency than DRAM. We evaluated the micro-benchmark for 5 different MPKI values between 2 and 125. By varying the sizes of PCM and DRAM for a given MPKI configuration, we could confirm the shape and position of the curve ΔE = 0.

3.1 Energy

We find that PCM, STT-RAM and RRAM behave as in the first scenario of Figure 1. Most importantly, the thresholds S^{∞} that we find for these technologies are relatively small (Table 1). In the case of STT-RAM, it is 65.4GB, implying that *any STT-RAM main memory larger than 65.4 GB consumes less energy than the equivalent amount of DRAM*. This memory capacity is common, if not small, for contemporary workstations and servers.

As the energy model is a first-order model, thresholds may not be exact. However, in practice M is restricted to values in the range $0 - 100$ MPKI. Thus, the main conclusion that a threshold exists is valid. The thresholds are quite loose for large values. One may define the value S^* where $\Delta E = 0$ and $MPKI = 100$, which is a practically meaningful bound. For PCM and the energy equation, S^* is 220 GB.

3.2 Energy-Delay Product

Plugging in values for the model parameters shows that the S^{∞} thresholds are larger for EDP than they are for energy. This is expected, as NVM increases delay, which must be made up by larger savings in static power.

The results (Table 1) show that PCM is not quite as promising a technology as STT-RAM and RRAM: A memory size of 1.34 TB is required to make PCM outperform DRAM on EDP. However, technologies such as STT-RAM and RRAM can outperform DRAM for memory sizes of 130.9 GB and 75.1 GB, respectively.

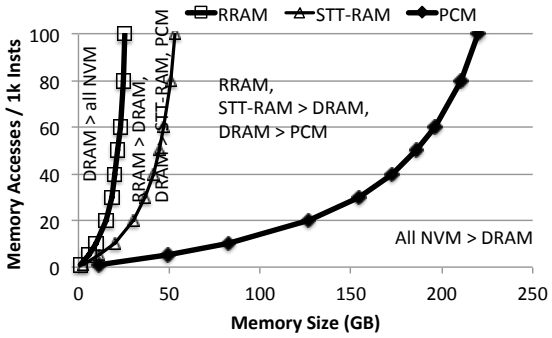


Fig. 2. Characterisation of workload space. The annotation X>Y indicates a region of workloads where memory technology X is more energy-efficient than technology Y.

3.3 NVM Latency Tolerance

The thresholds M^{∞} and S^{∞} depend on the memory access latencies L_{dram} and L_{nvm} through ΔL and ΔE_s. We can rewrite the thresholds in function of L_{dram} and ΔL (avoiding the use of L_{nvm}), e.g., for the energy equation:

$$S^{\infty} = -(\phi \Delta E_d + P_{cpu} \Delta L)/(\Delta P_s L_{dram} + P_{s,dram} \Delta L)$$

Note that $P_{cpu} > P_{s,dram}$, so S^{∞} increases in absolute value as ΔL grows. When ΔL becomes too large, S^{∞} becomes negative and the scenario changes (Figure 1). For a technology like PCM, the contribution of ΔL to S^{∞} is much larger than the contribution of ΔE_d. Consequently, minimising ΔL at the expense of ΔE_d increases the applicability of PCM.

If, however, NVM is optimised in isolation (i.e., P_{cpu} is assumed zero), then S^{∞} is minimized by minimizing ΔE_d. This assumption corresponds to optimizing a single memory characteristic in isolation. Such a solution would be sub-optimal at the system-level, where minimizing ΔL is more important than minimizing ΔE_d.

4 Hybrid Memory Systems

A hybrid main memory consists of S_{dram}GB of DRAM memory and S_{nvm} GB of non-volatile memory. A fraction μ of the memory accesses are directed to NVM, while a fraction $1 - \mu$ is directed to DRAM [12]. We assume reads and writes are distributed evenly across the memory types. We make no particular assumptions on whether memory traffic is distributed between DRAM and NVM by hardware or software.

We define energy per memory access $E_{d,hyb} = (1 - \mu)E_{d,dram} + \mu E_{d,nvm}$ and average memory access latency $L_{hyb} = (1 - \mu)L_{dram} + \mu L_{nvm}$. Static energy consumption is calculated as the weighed sum of static energy consumption in both DRAM and NVM.

The energy equation for hybrid memory is given as:

$$E_{hyb} = E_{d,hyb} MN + (P_{s,dram} S_{dram} + P_{s,nvm} S_{nvm} + P_{cpu}) T(L_{hyb})$$

TABLE 2
Asymptotes for hybrid memory system, assuming $S_{dram} = 8$ GB and $\mu = 50\%$.

	DRAM	PCM	STT-RAM	RRAM
	Energy comparison			
S^{∞}_{nvm}	-	143.9 GB	33.4 GB	17.2 GB
	EDP comparison			
S^{∞}_{nvm}	-	471.2 GB	59.0 GB	37.6 GB

We are interested in the comparison of a DRAM-only memory system versus a hybrid memory system. We define ΔE as the difference of energy consumed by a DRAM-only memory system of size $S_{dram} + S_{nvm}$ and average latency L_{dram} and a hybrid memory system with sizes S_{dram} and S_{nvm} and corresponding latencies L_{dram} and L_{nvm}. The model leads to similar observations as before, i.e., four distinct scenarios are possible, with values for the thresholds given as follows:

$$M^{\infty} = -\frac{CPI_0 \Delta P_s}{\Delta E_s + \frac{1-\mu}{\mu}\Delta P_s L_{dram}} \quad (7)$$

$$S^{\infty}_{nvm} = -\frac{\phi \Delta E_d + (P_{s,dram} S_{dram} + P_{cpu})\Delta L}{\Delta E_s + \frac{1-\mu}{\mu}\Delta P_s L_{dram}} \quad (8)$$

A similar analysis was performed to compare the EDP of a hybrid memory system with a DRAM-only system.

Hybrid memory hierarchies are very attractive from an energy-efficiency perspective (Table 2). A hybrid memory system with 8 GB of DRAM and at least 59.0 GB of STT-RAM is always more energy-efficient and has lower EDP than a DRAM-only system. But, again, we find that PCM is less promising.

5 Related Work

Hybrid memories are a commonly studied approach to address main memory energy consumption. Researchers often assume small memory sizes, e.g. 8 GB or less [6], [14], [15]. As such, dynamic energy consumption is important in these studies. In this paper, we have shown the existence of a threshold memory size S^{∞} above which NVM outperforms DRAM in terms of energy and EDP. As static energy dominates for large memory sizes, many of the proposed techniques to reduce NVM energy consumption become mute at scale, or they should be interpreted as a way of lowering the S^{∞} threshold.

This work does not optimize the performance of NVM. Performance must be optimised independently, typically through caching. Similarly, a common goal is to modify the NVM memory layout [3], [5], [9]. Our hybrid memory model may be applied to these studies by adjusting the parameters E_d, P_s and L to the optimised values. Note that asymmetric read/write memory latencies can be expressed in the parameter L by computing the appropriate average latency. The parameter μ can model caching. Conversely, the model may be applied to evaluate the benefits that optimisation of the model parameters may bring, select what parameters to optimise, or express constraints on the optimisation (e.g., $S^{\infty} < 64\,GB$).

6 Conclusion

We have presented an analytical model for comparing energy and energy-delay product of non-volatile and DRAM memory technologies. Workloads are characterised by memory size and access frequency. The model shows that byte-addressable NVM technologies are more energy-efficient than DRAM when the memory size exceeds a technology-dependent threshold. The model can be used to quickly evaluate the impact of changes to the memory architecture on energy, such as DRAM caching and write-avoidance schemes, by substituting improved values for the technology parameters. It can also indicate what memory system parameters to optimize.

The presented model applies to latency-sensitive workloads. In future work, bandwidth restrictions and read/write asymmetry will be addressed.

Acknowledgments

This work is supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013), grant agreement no. 327744.

References

[1] N. Binkert, et al, The GEM5 simulator. *SIGARCH Comput Arch News*, 2011, 39(2):1–7.

[2] E. Doller, "Forging a future in memory - new technologies, new markets, new applications," in *Hot Chips Tutorials*, 2010.

[3] X. Dong, C. Xu, Y. Xie, and N. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 7, pp. 994–1007, July 2012.

[4] D. Economou, S. Rivoire, and C. Kozyrakis, "Full-system power analysis and modeling for server environments," in *Workshop on Modeling Benchmarking and Simulation (MOBS)*, 2006, p. 8.

[5] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory as a scalable DRAM alternative," in *Proc. of the Intl. Symp. on Computer Architecture*, 2009, pp. 2–13.

[6] H. G. Lee and N. Chang, "Energy-aware memory allocation in heterogeneous non-volatile memory systems," in *Proc. of the Intl. Symp. on Low Power Electronics and Design*, 2003, pp. 420–423.

[7] K. T. Lim and D. Meisner and A. G. Saidi and P. Ranganathan and T. F. Wenisch, "Thin servers with smart pipes: designing SoC accelerators for memcached," in *Proc. of the Intl. Symp. on Computer Architecture*, 2013, pp. 36–47.

[8] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: Eliminating server idle power," in *Proc. of the Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, 2009, pp. 205–216.

[9] J. Meza, J. Li, and O. Mutlu, "Evaluating row buffer locality in future non-volatile main memories," Carnegie Mellon University, Tech. Rep. SAFARI Technical Report No. 2012-002, Dec. 2012.

[10] L. Minas and B. Ellison, "The problem of power consumption in servers," Jun. 2012, [Online]. Available: http://www.intel.com/intelpress/articles/rpcs1.htm.

[11] A. K. Mishra, X. Dong, G. Sun, Y. Xie, N. Vijaykrishnan, and C. R. Das, "Architecting on-chip interconnects for stacked 3D STT-RAM caches in CMPs," in *Proc. of the Intl. Symp. on Computer Architecture*, 2011, pp. 69–80.

[12] M. K. Qureshi, V. Srinivasan, J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," in *Proc. of the Intl. Symp. on Computer Architecture*, 2009, pp. 24–33.

[13] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, DRAMSim2: A Cycle Accurate Memory System Simulator. *Computer Architecture Letters*, 2011, pp. 16-19.

[14] D.-J. Shin, S. K. Park, S. M. Kim, and K. H. Park, "Adaptive page grouping for energy efficiency in hybrid PRAM-DRAM main memory," in *Proc. of the Research in Applied Computation Symp.*, 2012, pp. 395–402.

[15] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "A durable and energy efficient main memory using phase change memory technology," in *Proc. of the Intl. Symp. on Computer Architecture*, 2009, pp. 14–23.