Toward Faster and More Efficient Training on CPUs Using STT-RAM-based Last Level Cache

Alexander Hankin[1], Maziar Amiraski[1], Karthik Sangaiah[2], Mark Hempstead[1]

[1]Tufts University, Medford, MA, USA

[2]Drexel University, Philadelphia, PA, USA

hankin@ece.tufts.edu, maziar@ece.tufts.edu, ks499@drexel.edu, mark@ece.tufts.edu

1. Abstract

Artificial intelligence (AI), especially neural network-based AI, has become ubiquitous in modern day computing. However, the training phase required for these networks demands significant computational resources and is the primary bottleneck as the community scales its AI capabilities. While GPUs and AI accelerators have begun to be used to address this problem, many of the industry's AI models are still trained on CPUs and are limited in large part by the memory system. Breakthroughs in NVM research over the past couple of decades has unlocked the potential for replacing on-chip SRAM with an NVM-based alternative. STT-RAM is an especially attractive replacement for SRAM in the last-level cache due to its density, low leakage, and most notably, endurance. Research into Spin-Torque Transfer RAM (STT-RAM) has explored the impact of trading off volatility for improved write latency.

2. Introduction

- Neural networks are often trained and deployed offline and inference decisions sent to a client device over a wireless network.
- The training phase is the major bottleneck in the way of ubiquitous, completely-distributed artificial intelligence (AI) on all kinds of devices.
- Some training is still done on the CPU, for example, in federated learning.
- Non-volatile memories (NVMs) may seem counter-intuitive given the frequency of writes in training and the notorious write characteristics of NVMs.
- However, the training phase has qualities that are well suited to NVMs, particularly Spin-Torque Transfer RAM (STT-RAM).

3. Motivation

- STT-RAM-based cache [1], [2] has higher density and lower static power than SRAM, but write energy and latency is larger.
- Relax the non-volatility characteristic of STT-RAM in order to reduce STT-RAM write latency and energy consumption.
- The relationships that govern this trade-off can be modeled by Equations 1–3 [6]:

$$\Delta \propto \frac{V \cdot HK \cdot Ms}{T} \quad (1)$$

$$t_{retention} \propto Ce^{\Delta} \quad (2)$$

$$I_C(t_{write}) = A \cdot (JC_0 + \frac{C}{t^{\gamma}_{write}}) \quad (3)$$

where:
Δ =thermal factor, V=volume, H_k=in-plane anisotropy field, M_s=saturation magnetization, T=absolute temperature [K], I_C=current, A=cross sectional area of the free layer, J_{C0}=critical current density at zero temperature, and C,γ=fitting constants

- Training is a good candidate for exploiting this tradeoff because of the high reuse as well as its robustness to error
- Also, training requires large working set sizes and is memory bound, so STT-RAM can enable significant increases in memory density at a low energy cost.

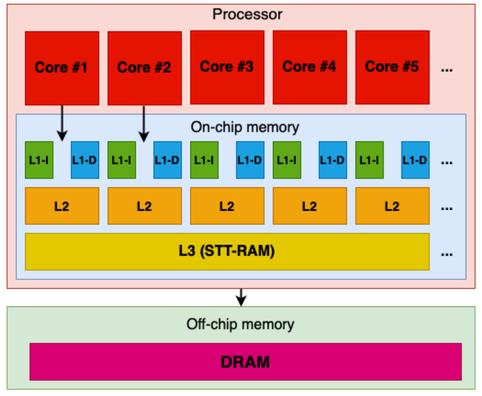

Figure 1: Topology of a CPU with STT-RAM-based last level cache (in yellow).

4. Exploration of a Low-Retention STT-RAM-based Last Level Cache

- Using an open source x86 simulator, Sniper [9], we created a simulation testbed of a CPU model based on a 14nm Intel Skylake processor running in turbo mode. We replace the SRAM-based LLC with an NVM-based LLC [7] keeping the physical area the same.
- Figure 2 shows the distances between a cache line write and a subsequent read for a common fully-connected neural network. Most distances are between 1 and 100 ns.
- To determine optimal retention time, we form a statistical distribution for retention failure as a function of LLC retention time (Table I).
- One STT-RAM model with a retention time of 1s and the other with 10ms. The forward pass of the training phase is quite sensitive to STT-RAM retention time. Reducing retention from 1s to 1ms results in two orders of magnitude reduction in probability of a retention failure. On the other hand, backpropagation is not significantly sensitive to differences in retention time.
- To see how retention errors affect accuracy of training, we inject errors into the model

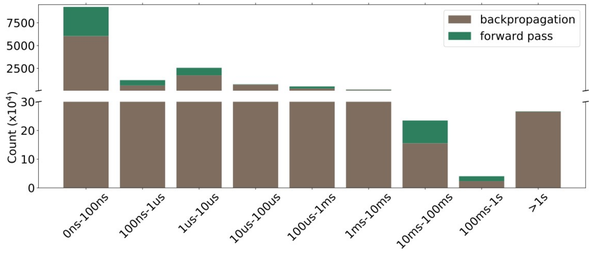

Figure 1: Histogram showing distribution of distances between cache line insertion and subsequent read.

	backpropagation	forward pass
$P_{retention\_failure}$(STTRAM_1s)	0.003	0.00002
$P_{retention\_failure}$(STTRAM_10ms)	0.002	0.002
$P_{retention\_failure}$(SRAM)	0.000	0.000

Table 1: Probability of retention failure for different memory technologies while running the DNN.

- We inject errors into the activations of lenet with error probabilities of 10^{-7}, 10^{-6}, and 10^{-5}:

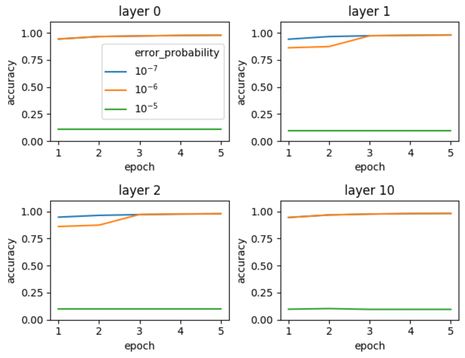

Figure 2: Error tolerance of lenet to injected activation errors. Error tolerance varies by layer.

- A 10^{-5} error probability does not allow the model to learn if applied to layer 0 (conv, 1024), layer 1(tanh, 4704), layer 2 (avgpool, 4704), or layer 10 (FC, 120).
- When coupled with an effective mitigation, a low-retention STT-RAM-based LLC can enable better training on CPUs.

5. References

[1] K. Korgaonkaret al., "Density tradeoffs of non-volatile memory as areplacement for sram based last level cache," in2018 ACM/IEEE 45thAnnual International Symposium on Computer Architecture (ISCA), 2018,pp. 315–327.
[2] A. Hankinet al., "Evaluation of non-volatile memory based last levelcache given modern use case behavior," in2019 IEEE InternationalSymposium on Workload Characterization (IISWC), 2019, pp. 143–154.
[3] P. Zhouet al., "Energy reduction for stt-ram using early write termina-tion," in2009 IEEE/ACM International Conference on Computer-AidedDesign - Digest of Technical Papers, Nov 2009, pp. 264–268.
[4] J. Ahnet al., "Write intensity prediction for energy-efficient non-volatilecaches," inInternational Symposium on Low Power Electronics andDesign (ISLPED), Sep. 2013, pp. 223–228.
[5] X. Wuet al., "Power and performance of read-write aware hybrid cacheswith non-volatile memories," in2009 Design, Automation Test in EuropeConference Exhibition, April 2009, pp. 737–742.
[6] C. W. Smullenet al., "Relaxing non-volatility for fast and energy-efficientstt-ram caches," in2011 IEEE 17th International Symposium on HighPerformance Computer Architecture, Feb 2011, pp. 50–61.
[7] A. Jogeret al., "Cache revive: Architecting volatile stt-ram caches forenhanced performance in cmps," inDAC Design Automation Conference2012, June 2012, pp. 243–252.
[8] Z. Sunet al., "Multi retention level stt-ram cache designs with adynamic refresh scheme," in2011 44th Annual IEEE/ACM InternationalSymposium on Microarchitecture (MICRO), Dec 2011, pp. 329–338.
[9] T. E. Carlsonet al., "Sniper: Exploring the level of abstraction forscalable and accurate parallel multi-core simulations," inInternationalConference for High Performance Computing, Networking, Storage andAnalysis (SC), Nov. 2011, pp. 52:1–52:12.