Banshee: Bandwidth-Efficient DRAM Caching via Software/Hardware Cooperation

Xiangyao Yu[†], Christopher J. Hughes[‡], Nadathur Satish[‡], Onur Mutlu[§], Srinivas Devadas[†]
[†] CSAIL, MIT, [‡] Intel Labs, [§] ETH Zurich
[†]{yxy, devadas}@mit.edu, [‡]{christopher.j.hughes, nadathur.rajagopalan.satish}@intel.com, [§]onur.mutlu@inf.ethz.ch

Review by Xian Kai Ng

Article: http://news.mit.edu/2017/new-high-capacity-data-caches-more-efficient-1023
Paper: https://arxiv.org/abs/1704.02677
[1]: https://www.altera.com/products/sip/memory.html

Summary:

Banshee aims to capitalize on in-package DRAM's quadrupled data throughput, using it as a cache for off-package DRAM. Higher throughput improves the performance of hot bandwidth-bound, highly parallel applications: Machine Learning, Sparse Linear Algebra, Quantum Simulations, etc. Its main propositions are PTE/TLB extension bits to record cache metadata, a hardware tag buffer, a modified cache layout for dram cache, and a "hardware-managed frequency-based replacement policy."

How it works:

Since in-package DRAM's inception ~2 years ago [1], researches have debated on its integration with off-package DRAM. Should we use the same or different addresses for in and off-package RAM? Use hardware-managed or software-managed caches? Banshee maps data with bits on the TLB and PTE like software-based implementations, but tags the data with cached bits and way bits to tell the memory controller which RAM to find the data in instead of using different addresses. With distinct addresses, in currently published implementations (as of Nov '17), every time a page is remapped from/to in to/from off-package RAM, the memory system needs to scrub off the old physical addresses and put new ones on all the on-chip caches that use physical addressing (usually L2 and below) and the PTE (which is itself in RAM or on-chip cache) and the TLB, creating high memory management traffic overhead. Bandwidth inefficiency. With Banshee, we still need to change the metadata bits on the TLB and PTEs with Banshee but this cost is amortized via the tag buffer.

The tag buffer is exactly that—a store of physical addresses and their corresponding cached, way, valid, and remap bits in the memory controller (MC). The paper's tag buffer visualization:

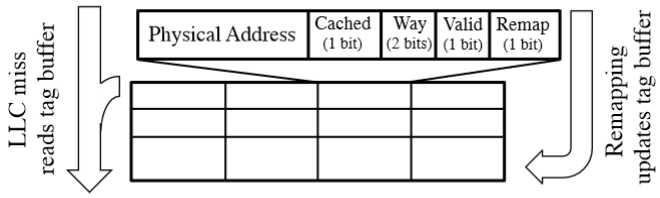

Figure 2: Tag buffer organization.

Every time a page is remapped from between the RAMs, we reflect this change on the tag buffer (no need to access RAM a second time or cache to update the tag bits there).

Let's explain what the different tags are:

-Physical address serves as the index

-Cached, Way, Valid bits are as in normal caches

-Remap bit of 1 tells the MC to rewrite those physical address's tags in the TLB/PTEs when the tag buffer is flushed.

The frequency-based replacement (FBR) refers to an algorithm that samples every so often (the sampling rate can be set, and scales with the recent miss rate), which pages are being accessed. It tracks the cached pages as well as candidate pages whose counter is also tracked, and may replace cached pages in in-package RAM. If a non-candidate, non-cached page is caught in sampling, it has a probability of 1 over (random candidate page's count) of replacing that candidate page as a new candidate. The FBR aims to reduce the frequency of moving pages on and off DRAM cache as it clogs the RAM bandwidth channel. Banshee's FBR also includes a threshold where candidates' count has to be victims' count plus the threshold to trigger replacement; it prevents thrashing. Because of the sampling and FBR, it takes a while for the tag buffer to fill up, and we flush it (and thus scrub cache, TLB and PTEs) only when it reaches ~70% full, so the bandwidth overhead is amortized to almost nothing. Small sized tag buffers also mean low memory space overhead, and allows the use of fast memory at low cost.

Applications:

On a benchmark involving PageRank, geometric mean calculations, triangle counting, graph500, etc, Banshee outperforms previous designs, with PageRank performance being 3x faster than no cache, and even somehow exceeding in-package-only RAM. The authors attribute this to Banshee's having two channels for main memory. Other applications like quantum mechanics simulation libquantum had pure in-package beating Banshee, but Banshee still performs 3x faster than no cache. And going pure in-package will be expensive. Intel's Knight's Landing will come with 16GB on-package RAM max, which is nothing for highly parallel industry applications, and I suspect there would be close to thousand-dollar markups—an order of magnitude higher than off-package RAM. So, an effective balance between cost and performance like Banshee can step in to make cash money, like in Bitcoin.

This increased speed should attract many developers in blazing hot fields today including machine learning which can be highly parallelized, and thus bandwidth-bound more so than compute-bound, and one I'm personally invested in: Cryptocurrency mining. Bitcoin mining is embarrassingly parallel so high-throughput, fast RAM means fat profits. Indeed, Intel and Micron (they worked on the Knight's Landing chip and Intel Labs contributed to this paper) have seen surging stock prices this year; investors are optimistic that the new hardware tech and associated low-level software optimizations can boost the inflating industry of AI, crypto, and Big Data (also bandwidth-bound—I mean, it's called big data i.e. a lot of data coming in).

Banshee may be found before 2020 in our self-driving cars collecting data for the fleet, in enthusiasts' or investors' homes or data centers mining the latest cryptocurrency (BansheeCoin?), in personalization calculators processing billions of people's preferences—ad servers—and in 16K 120Hz video producers' machines. Though Moore's law for computing may be halting, distributed computing and multi-core technology can satisfy the compute requirements, so speeding up memory is now the bottleneck in sustaining Moore's Law's results in technological advancement. Indeed, Banshee could directly help sustain Moore's law as the computing paradigm shifts to quantum. Quantum physics simulations sped up by faster RAM can supplement research in quantum computing. Banshee + in-package RAM => better computers, better AI => more GDP. I think this will be a hit; I'm betting on it with over half my savings in MU and INTC.