Leveraging Page Size Information to Enhance Data Cache Prefetching

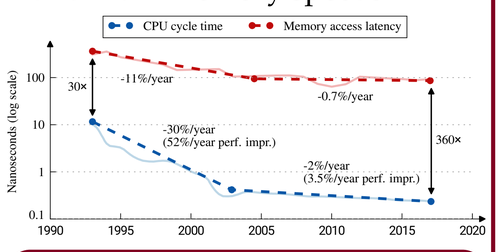
Georgios Vavouliotis[1,3], Gino Chancon[2], Lluc Alvarez[1,3], Paul V. Gratz[2], Daniel A. Jiménez[2], and Marc Casas[1,3]

[1]*Barcelona Supercomputing Center* [2]*Texas A&M University* [3]*Universitat Politècnica de Catalunya*

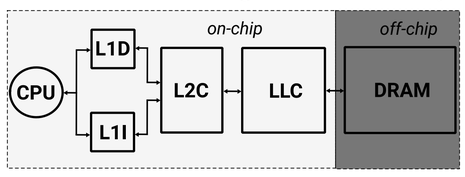
Memory Bottleneck

CPU vs RAM Speeds

- Discrepancy between processor and main memory speeds



- Architects use on-chip cache hierarchies to reduce the latency cost of accessing main memory



- On-chip caches partially reduce main memory accesses due to limited capacity

Cache Prefetching

Fundamental Idea

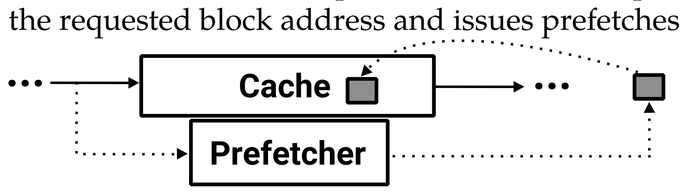
- Proactively fetch data blocks into the on-chip caches before they are explicitly requested

Why?

- HPC and Big Data workloads feature massive data footprints that do not fit in the on-chip caches

Cache Prefetching in Practise

- Prefetching can be applied on each cache level
- On a cache access, the prefetcher takes as input the requested block address and issues prefetches



- Today, all HPC chips employ various data cache prefetchers to capture heterogeneous access patterns (e.g., Intel IceLake, AMD Zen)

Methodology

Simulator

- ChampSim

Baseline

- No prefetching at any cache level

Simulated System

- L1D Cache: 48KB (12-way)
- L1I Cache: 32KB (8-way)
- L2 Cache: 512KB (8-way)
- L3 Cache: 2MB (16-way)
- DRAM: 8GB

Workloads

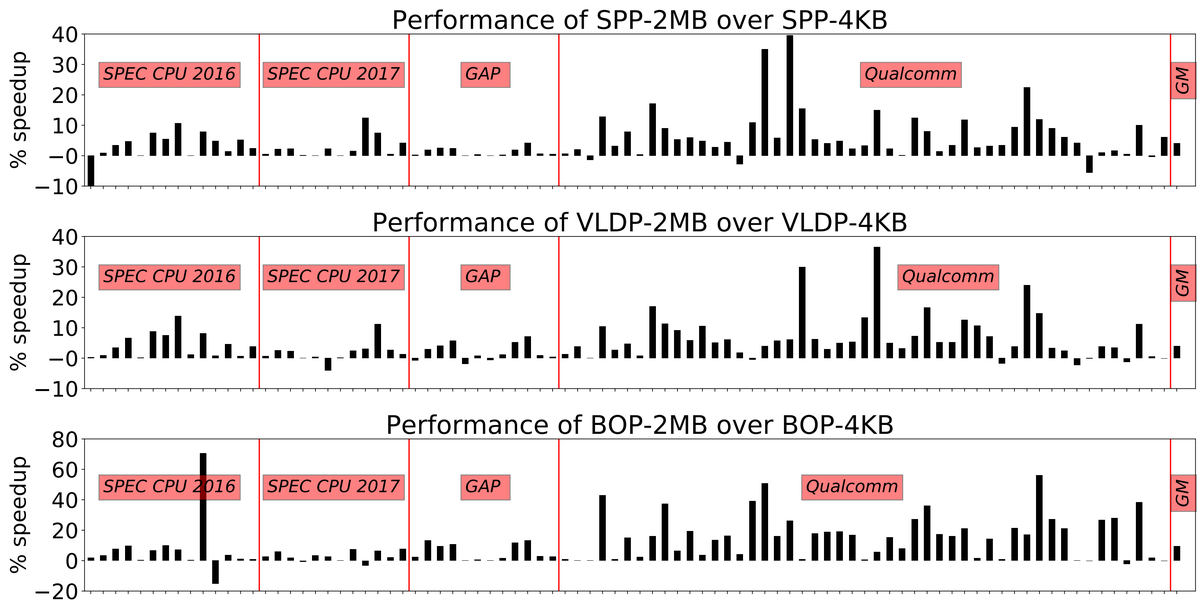
- SPEC CPU 2006
- SPEC CPU 2017
- Qualcomm traces provided for CVP-1
- GAP suite

Evaluated L2 Cache Prefetchers

- SPP [1]
- VLDP [2]
- BOP [3]

Potential Performance Gains

- System with only 2MB pages (>90% of the allocated pages are 2MB for these workloads)
- Evaluated versions of cache prefetchers
  - <Cache-Prefetcher>-4KB: stop prefetching at 4KB boundaries
  - <Cache-Prefetcher>-2MB: stop prefetching at 2MB boundaries



SPP-2MB, VLDP-2MB, and BOP-2MB improve geomean performance over SPP-4KB, VLDP-4KB, and BOP-4KB by **4.1%**, **4.0%**, and **9.4%**, respectively

Virtual Memory Sub-System

Overview

- All modern systems implement paging-based virtual memory
- Virtual and physical address spaces are split into pages
- Each memory access requires a virtual-to-physical address translation

Page Sizes

- The standard page size is 4KB in most systems
- To improve virtual memory management modern OSes provide support for larger page sizes (e.g., 2MB and 1GB pages)

Prior Work on Cache Prefetching & Opportunities

- Numerous cache prefetchers have been proposed in recent literature [1-3]
- All prior works propose cache prefetchers that use complex prefetching algorithms to capture more distinct patterns than the previously proposed designs

Common Aspects of All Cache Prefetchers

- All prior cache prefetchers are designed assuming 4KB pages
- All prior cache prefetchers do not permit prefetching beyond 4KB physical page boundaries because physical contiguity is not guaranteed

Opportunity for Improving Cache Prefetching

- Modern systems vastly use larger page sizes to reduce address translation overheads
- Physical pages are equally sized with the virtual pages (e.g., when a virtual page is 2MB the corresponding physical page is also 2MB)
- Intuitively, limiting cache prefetchers to prefetch within 4KB boundaries when larger page sizes are used results in sub-optimal performance gains

Physical Machine Measurements

- Evaluation of SPEC 2006, SPEC 2017, and GAP suites on an Intel Xeon machine
  - More than 90% of the allocated pages are 2MB pages for these workloads

Our Approach

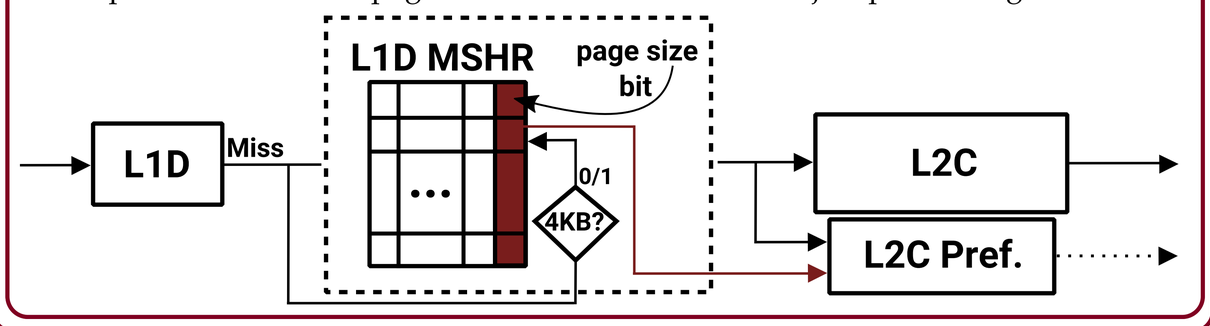
- Enhance the performance of all prior and new cache prefetchers by propagating the page size information
- Fake & safe beyond page boundaries cache prefetching
  - Cache prefetcher crosses 4KB boundaries when the block resides in a 2MB page
- More effective and timely cache prefetching

Propagating the Page Size Information

- Modern OSes concurrently support multiple page sizes (e.g., 4KB, 2MB)
- L1D Caches are VIPT –> On L1D misses page size is known
- L2C prefetchers are engaged on L1D misses

Page Size Propagation Scheme

- Enhance L1D MSHR with one bit, indicating the page size
- L2C prefetchers use the page size bit of L1D MSHR to adjust prefetching boundaries



Conclusions

- Propagating the page size to cache prefetchers provides significant performance gains
- Applicable to all prior and new cache prefetchers
- Potential impact on future industrial designs

References

[1] J. Kim et al., "Path confidence based lookahead prefetching", MICRO'16
[2] M. Shevgoor et al., "Efficiently prefetching complex address patterns", MICRO'15
[3] P. Michaud, "Best-offset hardware prefetching", HPCA'16