Comparing Cache Injection and Data Prefetching for I/O in CMPs

Edgar A. León and Arthur B. Maccabe
The University of New Mexico
Scalable Systems Lab

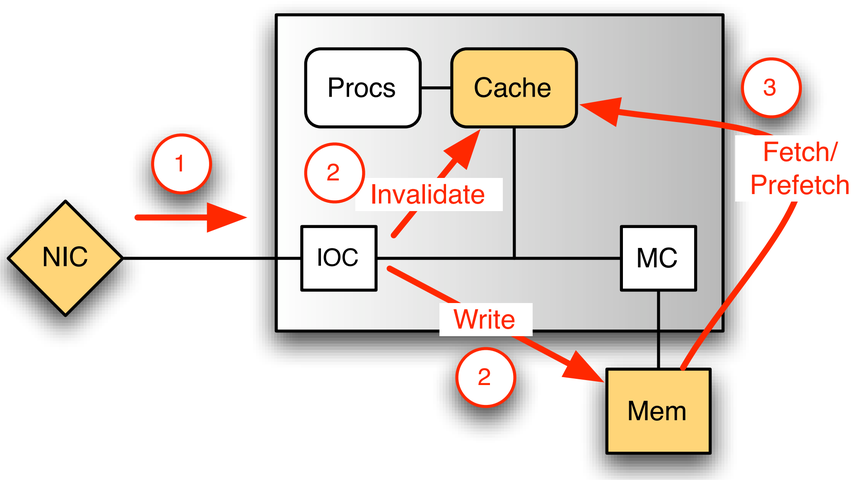
The Memory Wall

- The *memory wall*: disparity between processor and memory speeds.
- Memory wall adversely affects application performance, specially those limited by *memory bandwidth*.
- Examples: scientific computations (e.g., DNA matching), encryption, signal processing, some graphics applications, etc.
- *Prefetching* widely used to address this problem.
- *Cache injection* reduces memory latency and memory pressure for I/O.

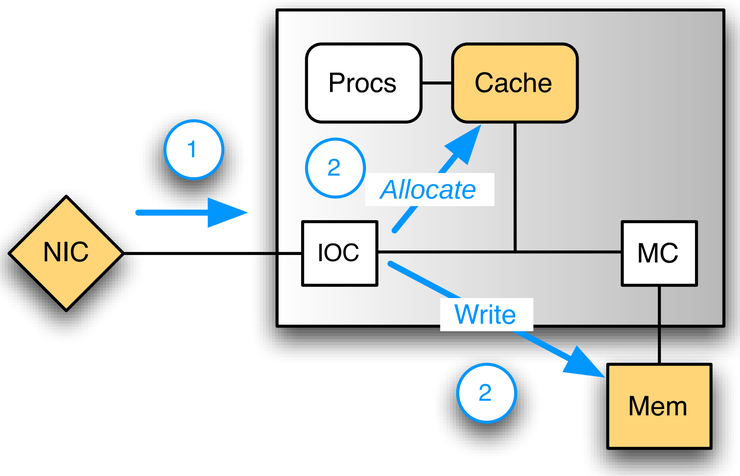
How does cache injection compare to prefetching?

What is Cache Injection?

- Producer-driven non-binding technique to place data from I/O DMA devices directly into cache.
  - Producer-driven: data transfer initiated by producer.
  - Non-binding: data is not bound to a particular cache block.



Current Architectures



Cache Injection Architecture

Comparing Cache Injection and Prefetching

	Data Prefetching	Cache Injection
Resource Usage	1) write to memory 2) fetch to cache	1) write to cache
	incurs memory latency and memory bandwidth usage	reduces memory latency and bandwidth usage
Fails When	data fetched too late	data injected too early
Applicability	general-purpose	limited to I/O data
Communication	consumer-driven	producer-driven

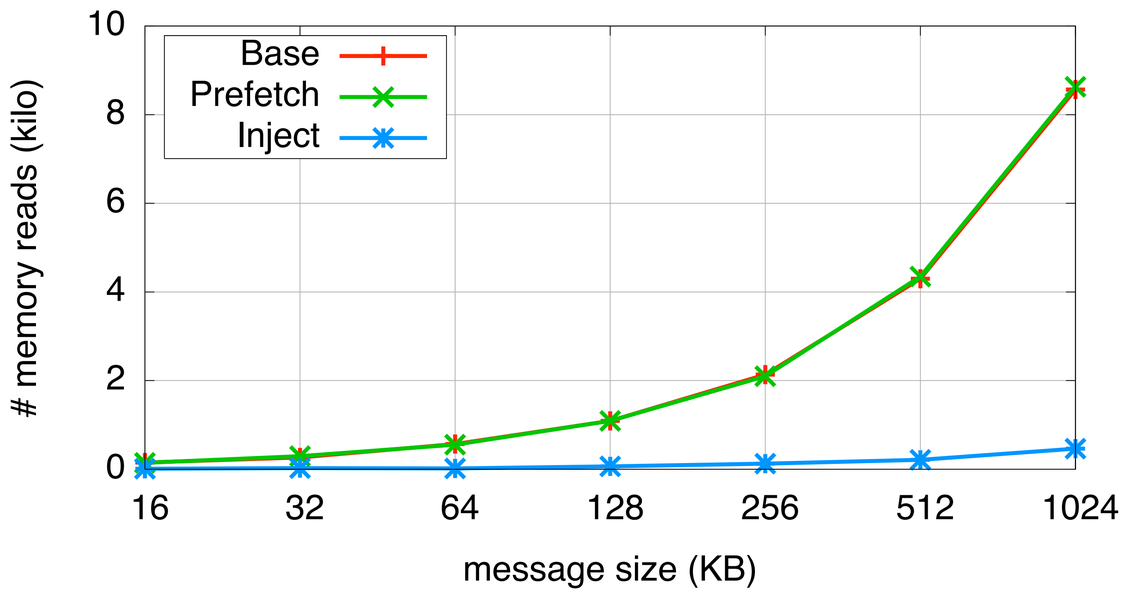
Cache injection reduces memory latency and memory pressure for I/O

Experimental Evaluation and Results

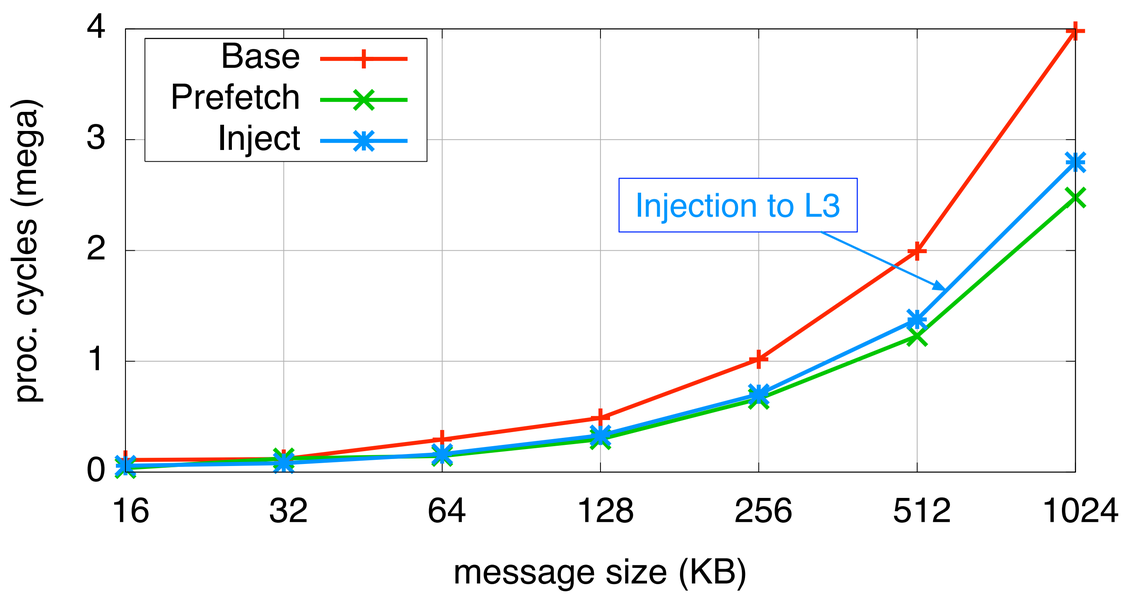
- Measure memory bandwidth utilization and execution time of an application using simulation.
- Application performs linear traversal of incoming network data followed by a reduction operation.

Simulation env.	Mambo: IBM PowerPC full-system simulator
OS	IBM K42 research OS.
Communication	OS-bypass, zero-copy UDP implementation
Processor freq.	1.65 GHz
L1 I/D cache	64KB/32KB, 2-way/4-way, 128B line
L2 cache	1.875MB, 3-slice, 10-way, 10 cycle latency
L3 cache	36MB, 3-slice, 12-way, 80 cycle latency
Main memory	512MB, 230 cycle latency

Memory Bandwidth



Run Time



Related Work

- Bohrer et al. Method and apparatus for accelerating I/O processing using cache injections. US Patent 6,711,650 B1, 2004.
- Huggahalli et al. Direct cache access for high bandwidth network I/O. ISCA 2005.

Conclusions and Future Work

- Cache injection outperforms prefetching in terms of memory bandwidth and compares well on execution time.
- Appropriate injection policies are necessary to leverage cache injection.
- Study injection policies based on OS, compiler, cache, and application information.

Acknowledgments

Orran Krieger, Michal Ostrowski, Lixin Zhang and Hazim Shafi from IBM.
The Scalable System Lab at UNM.
This work was supported by IBM and Intel.