Linearly Compressed Pages: A Main Memory Compression Framework with Low Complexity and Low Latency

Gennady Pekhimenko

Advisers: Todd C. Mowry & Onur Mutlu

Carnegie Mellon

Executive Summary

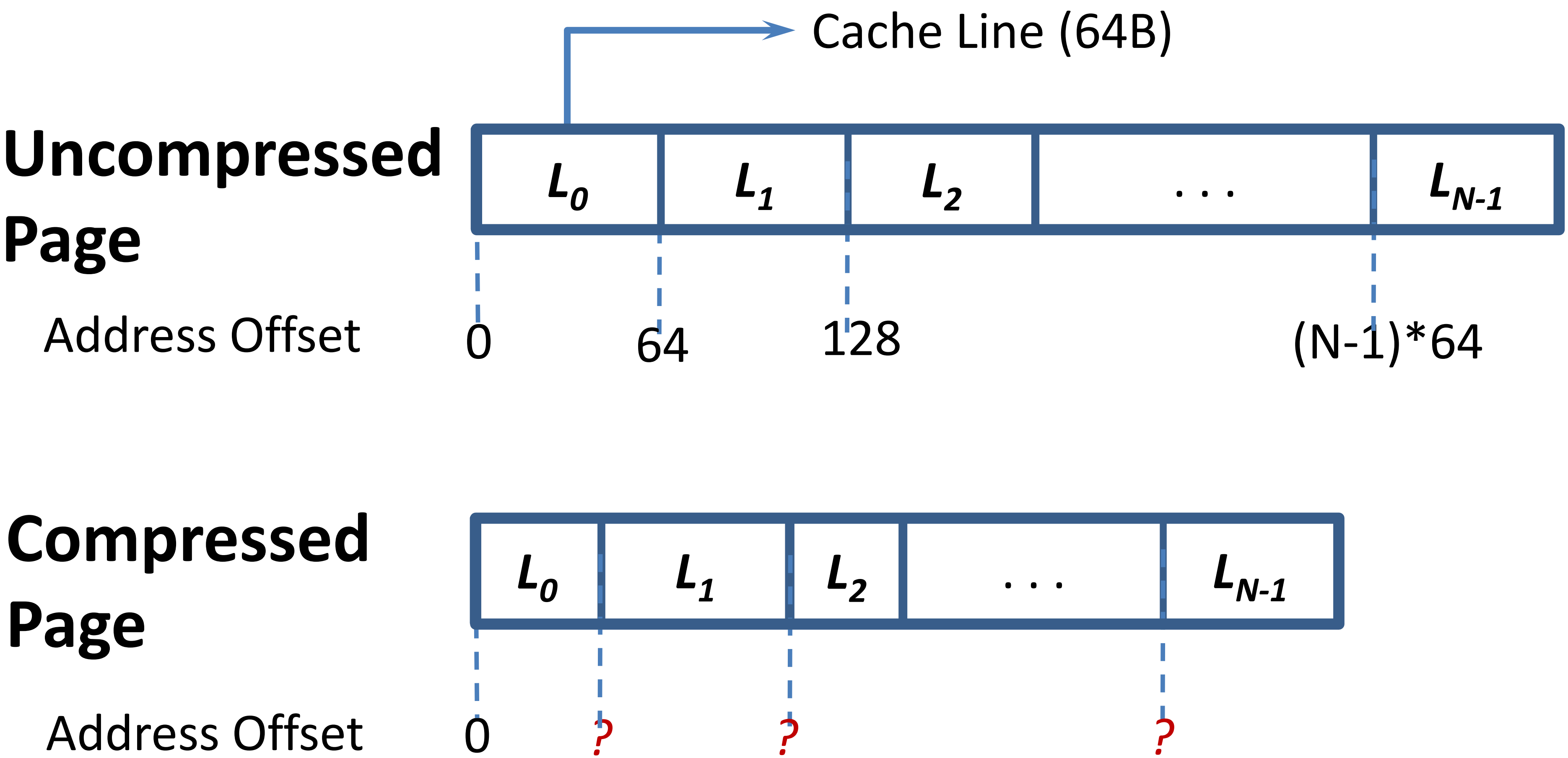
- Main memory is a limited shared resource
- **Observation**: Significant data redundancy
- **Idea**: Compress data in main memory
- **Problem**: How to avoid latency increase?
- **Solution**: Linearly Compressed Pages (LCP): fixed-size cache line granularity compression

1. Increases capacity (**69%** on average)
2. Decreases bandwidth consumption (**46%**)
3. Improves overall performance (**9.5%**)

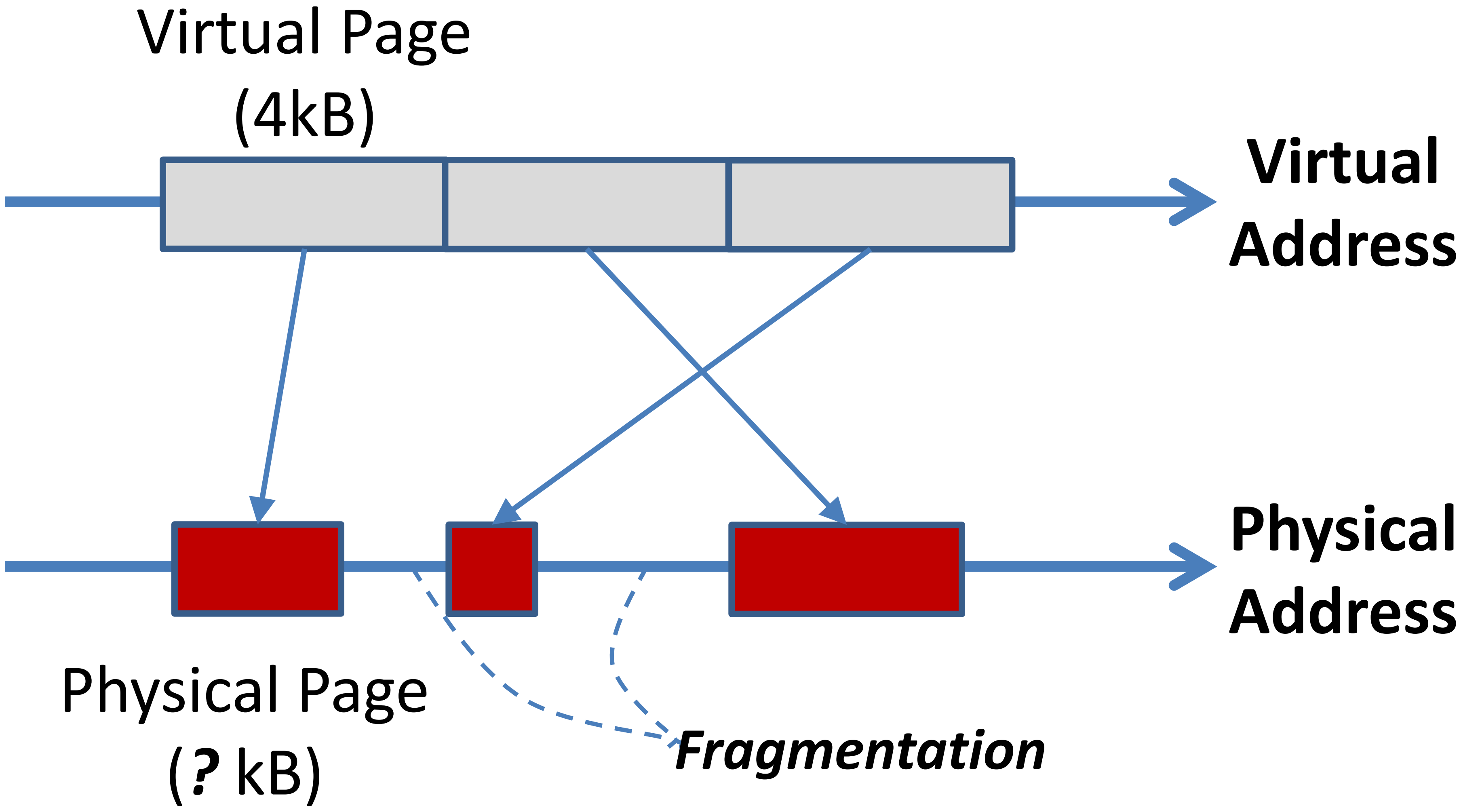
Challenges in Main Memory Compression

1. Address Computation
2. Mapping and Fragmentation
3. Physically Tagged Caches

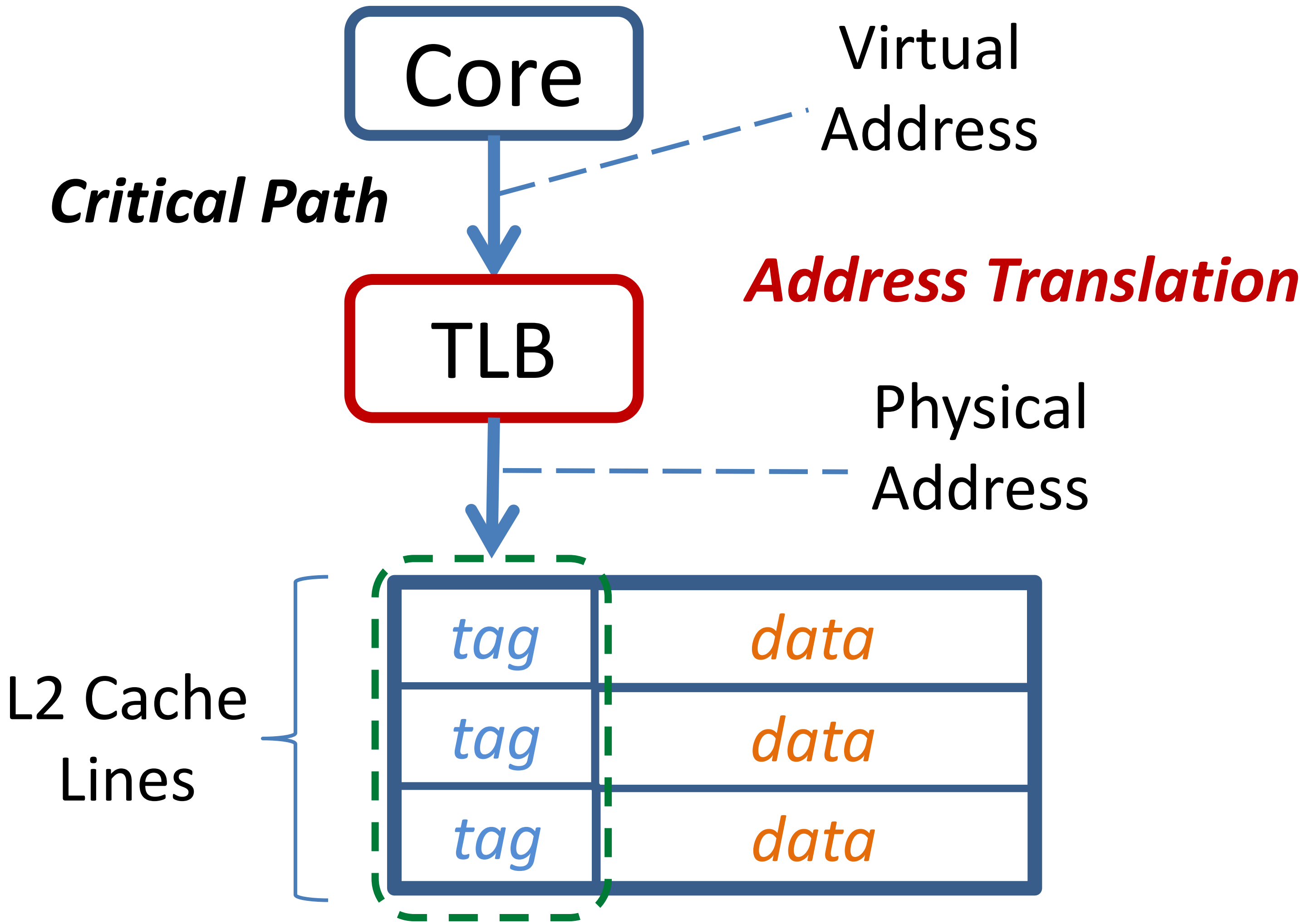
Address Computation



Mapping and Fragmentation



Physically Tagged Caches



Shortcomings of Prior Work

Compression Mechanisms	Access Latency	Decompression Latency	Complexity	Compression Ratio
IBM MXT *[IBM J.R.D. '01]*	✘	✘	✘	✔

Shortcomings of Prior Work

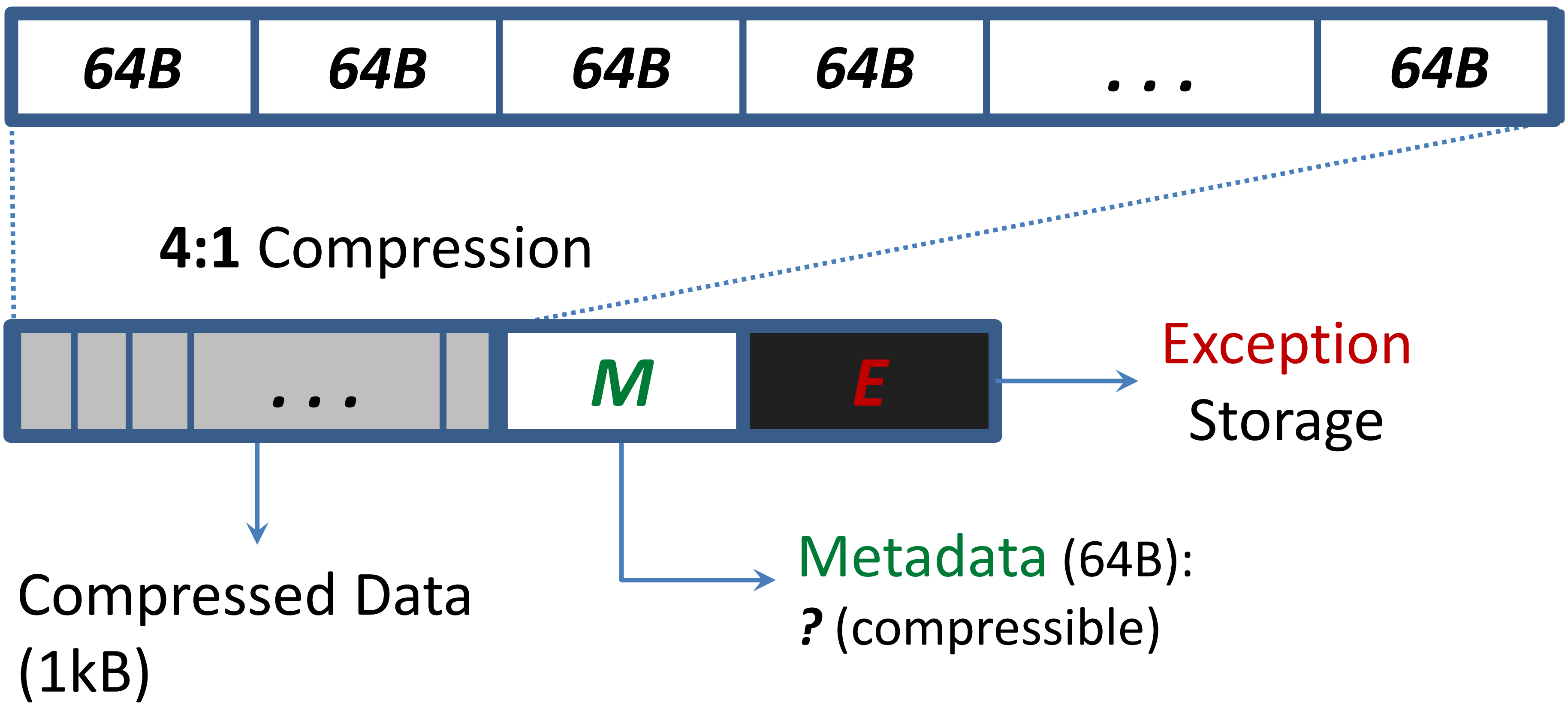
Compression Mechanisms	Access Latency	Decompression Latency	Complexity	Compression Ratio
IBM MXT *[IBM J.R.D. '01]*	✘	✘	✘	✔
Robust Main Memory Compression *[ISCA'05]*	✘	✔	✘	✔

Shortcomings of Prior Work

Compression Mechanisms	Access Latency	Decompression Latency	Complexity	Compression Ratio
IBM MXT *[IBM J.R.D. ’01]*	✘	✘	✘	✔
Robust Main Memory Compression *[ISCA’05]*	✘	✔	✘	✔
LCP: Our Proposal	✔	✔	✔	✔

Linearly Compressed Pages (LCP): Key Idea

Uncompressed Page (4kB: 64***64B**)

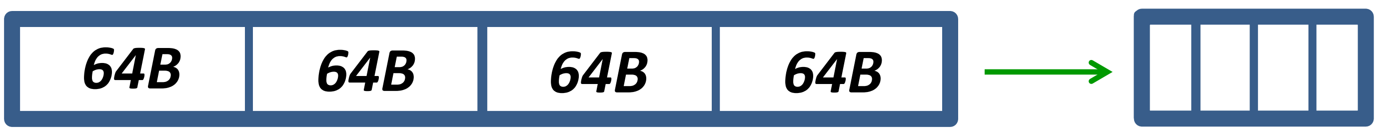


LCP Overview

- Page Table entry extension
  - compression type and size
  - extended physical base address
- Operating System management support
  - **4** memory pools (512B, 1kB, 2kB, 4kB)
- Changes to cache tagging logic
  - physical page base address + **cache line index** (within a page)
- Handling page overflows
- Compression algorithms: **BDI** *[PACT'12]* , **FPC** *[ISCA'04]*

LCP Optimizations

- **Metadata** cache
  - Avoids additional requests to metadata
- Memory bandwidth reduction:



1 transfer
instead of 4

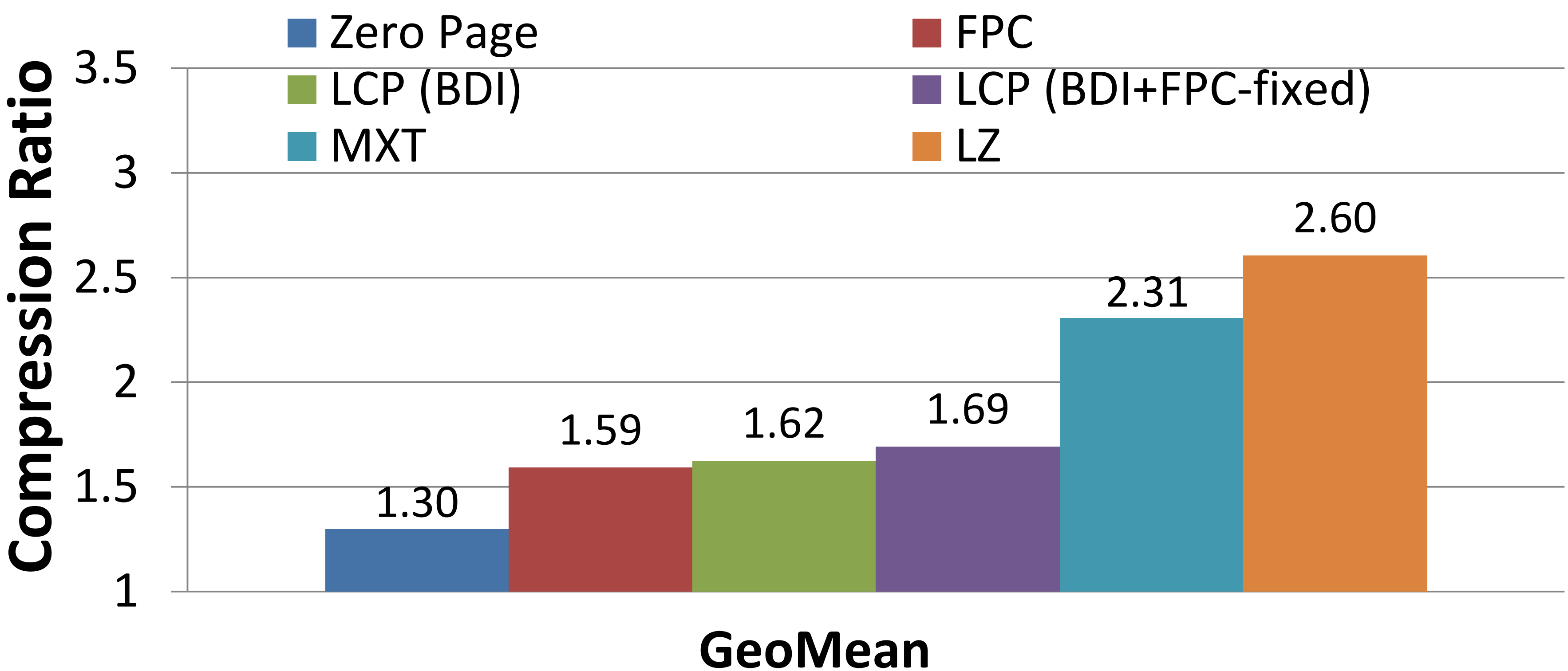
- Zero pages and zero cache lines
  - Handled separately in TLB (1-bit) and in metadata (1-bit per cache line)
- Integration with cache compression
  - BDI and FPC

Methodology

- **Simulator**
  - x86 event-driven simulators
    - Simics-based *[Magnusson+, Computer'02]* for CPU
    - Multi2Sim *[Ubal+, PACT'12]* for GPU
- **Workloads**
  - SPEC2006 benchmarks, TPC, Apache web server, GPGPU applications
- **System Parameters**
  - L1/L2/L3 cache latencies from CACTI *[Thoziyoor+, ISCA'08]*
  - 512kB - 16MB L2, simple memory model

Compression Ratio Comparison

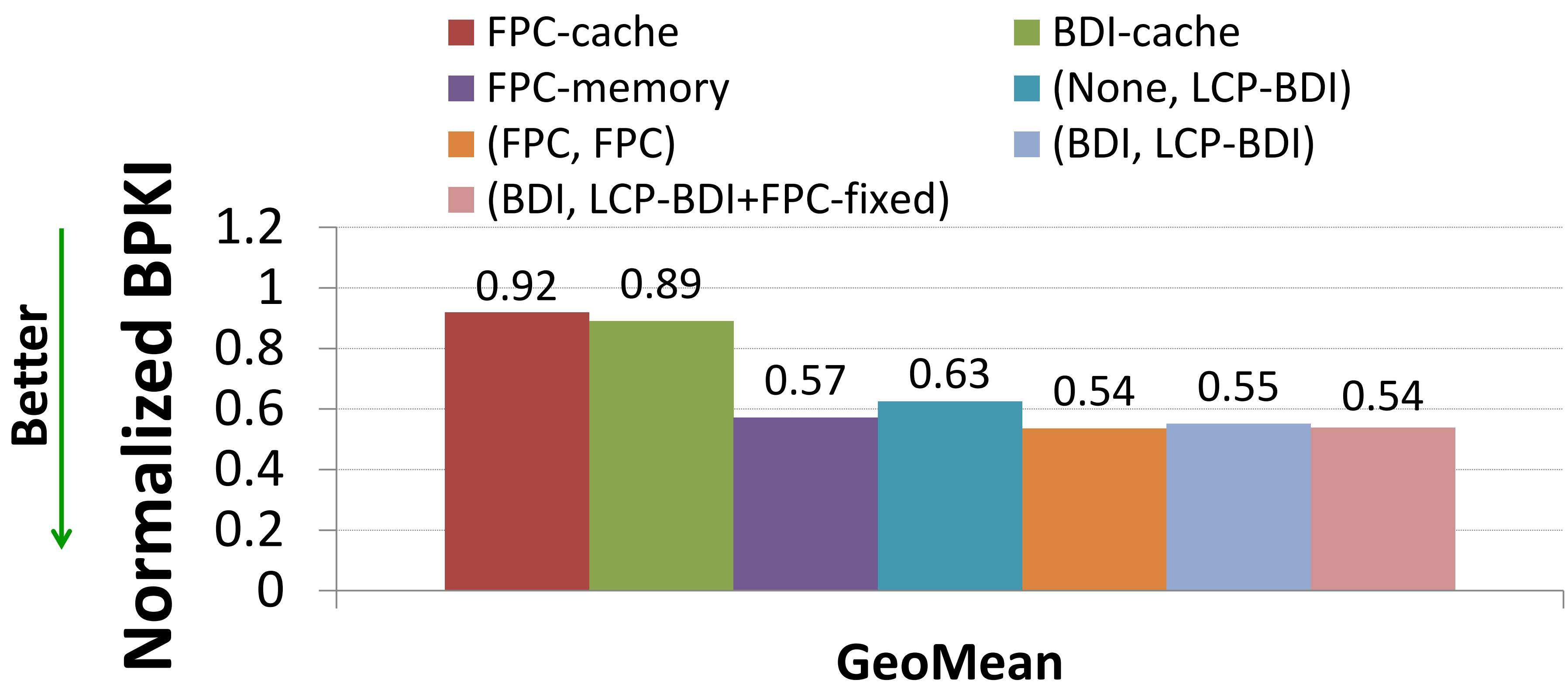
SPEC2006, databases, web workloads, 2MB L2 cache



LCP-based frameworks achieve competitive average compression ratios with prior work

Bandwidth Consumption Decrease

SPEC2006, databases, web workloads, 2MB L2 cache



LCP frameworks significantly reduce bandwidth (**46%**)

Performance Improvement

Cores	LCP-BDI	(BDI, LCP-BDI)	(BDI, LCP-BDI+FPC-fixed)
1	6.1%	**9.5%**	9.3%
2	13.9%	**23.7%**	23.6%
4	10.7%	**22.6%**	22.5%

LCP frameworks significantly improve performance

Conclusion

- A new main memory compression framework called **LCP(Linearly Compressed Pages)**
  - **Key idea: fixed size** for compressed cache lines within a page and **fixed compression algorithm** per page

- LCP evaluation:
  - Increases capacity (**69%** on average)
  - Decreases bandwidth consumption (**46%**)
  - Improves overall performance (**9.5%**)
  - Decreases energy of the off-chip bus (**37**%)

Linearly Compressed Pages: A Main Memory Compression Framework with Low Complexity and Low Latency

Gennady Pekhimenko

Advisers: Todd C. Mowry & Onur Mutlu

Carnegie Mellon