Improve Media Processing on General-Purpose Processor

Bo Zhang, Guohui Wang, Jason Sedano

Rice University

Motivation

- Background
  - General-purpose processors execute a larger fraction of media processing applications today.
  - Media processing applications have significantly different characteristics from other general applications.
    - Little data reuse, low data dependence, high data parallelism
  - General-purpose processors show much worse performance on media applications compared against special media processors

- Problem: Can we leverage the features of media applications in data access, and run them on general purpose processor more efficiently by modifying memory subsystem?

What to do?

- Quantitatively characterize the media application behaviors with respect to cache usage.

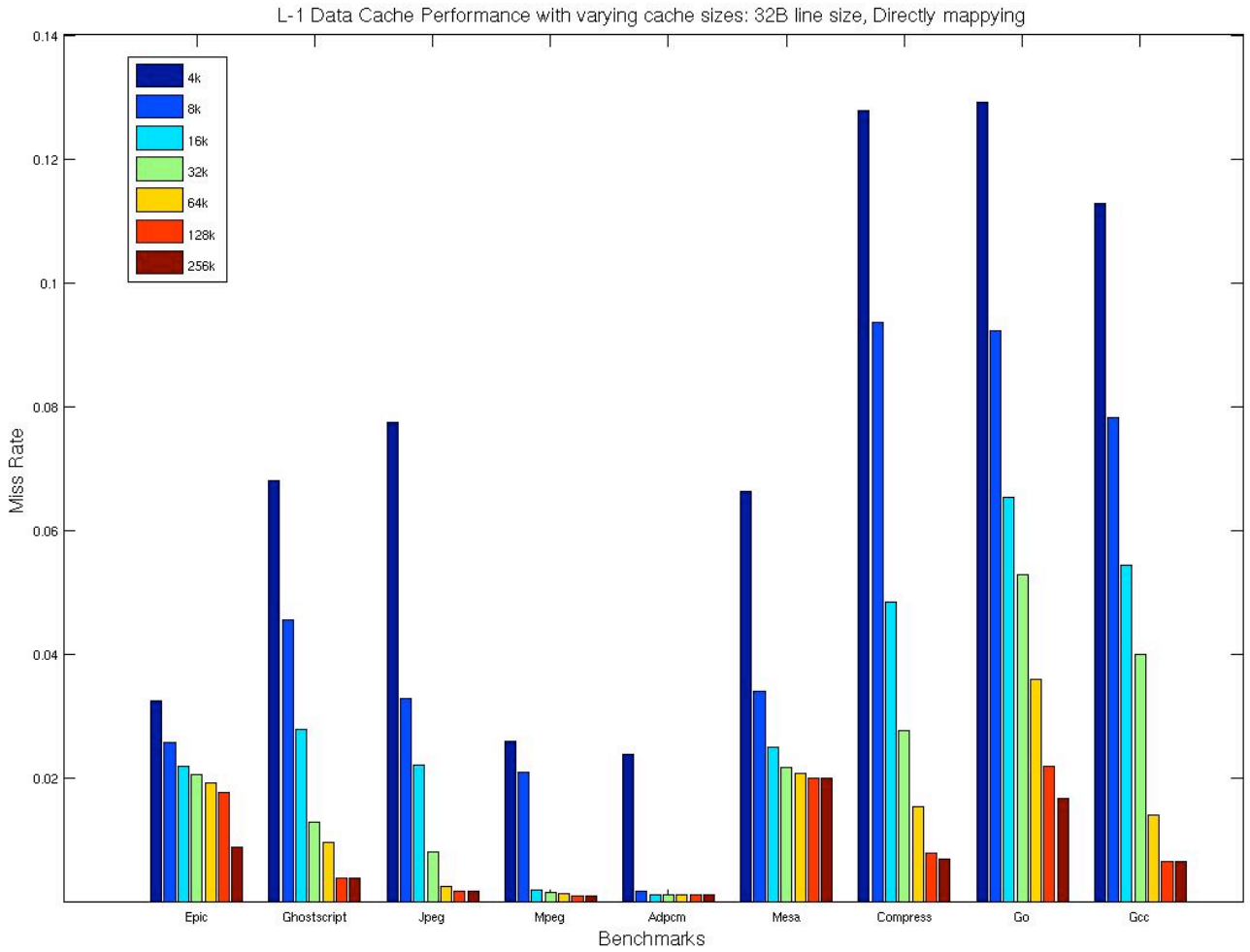
- Based on analytical understanding of media applications behaviors, we try to improve its performance by pre-fetch mechanism.

Mothodology

- Simulation
  - SimpleScalar version 3.0, PISA
  - On SPARC host machine
- Benchmarks used:
  - Media: ADPCM, JPEG MPEG2,Mesa, Ghostscript,Epic
  - Spec2000: GCC, Go, Compress
- We took into account the following variables and all possible combinations of varying sizes:
  - Number of sets: 64, 128, 256, 512, 1024, 2048, 4096, 8192
  - Block size(Bytes): 8, 16, 32, 64
  - Associativity: 1,2,4,8
  - Simulation combinations run : 144 per benchmark

L1 Data Cache Performance

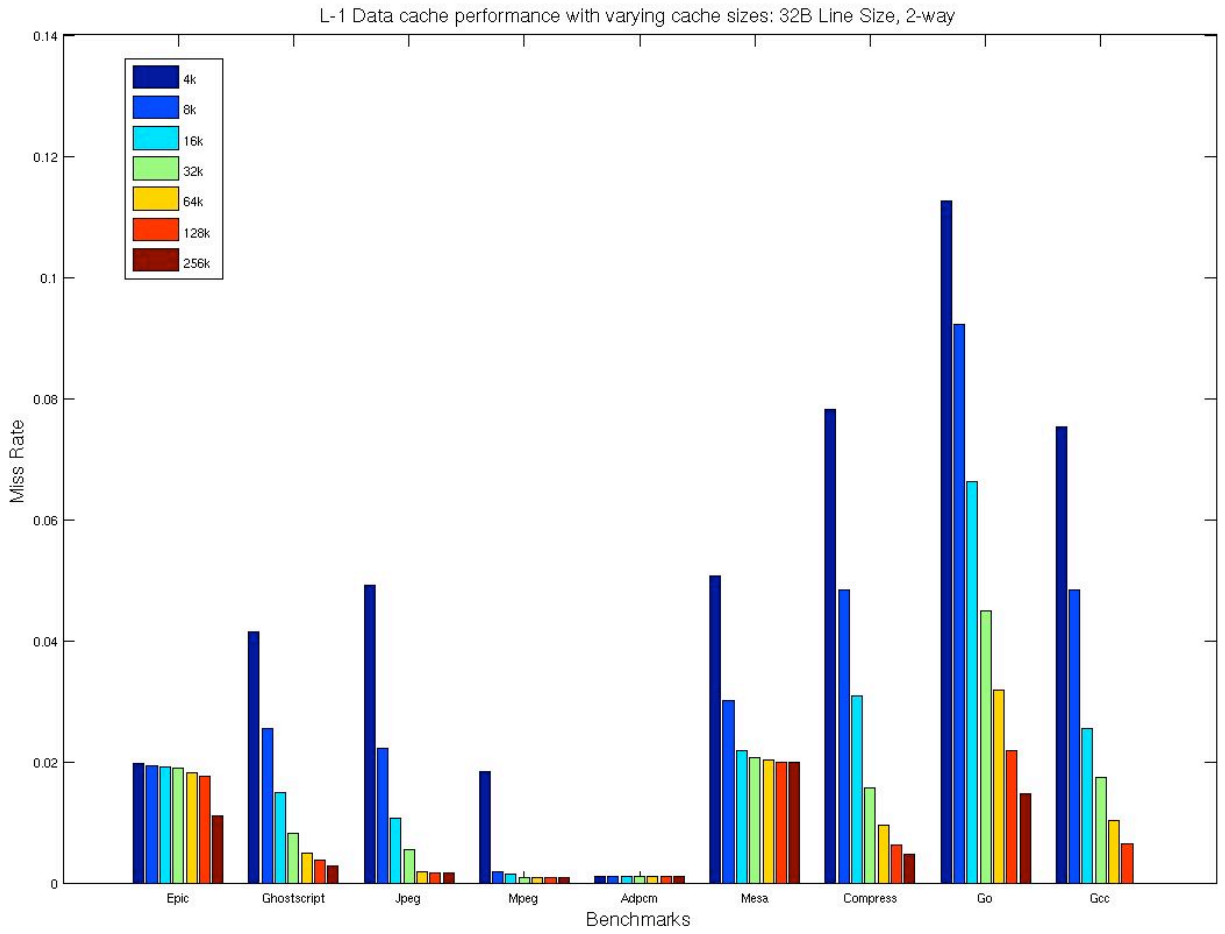
32B Line Size, directly mapping, varying cache sizes



Bc

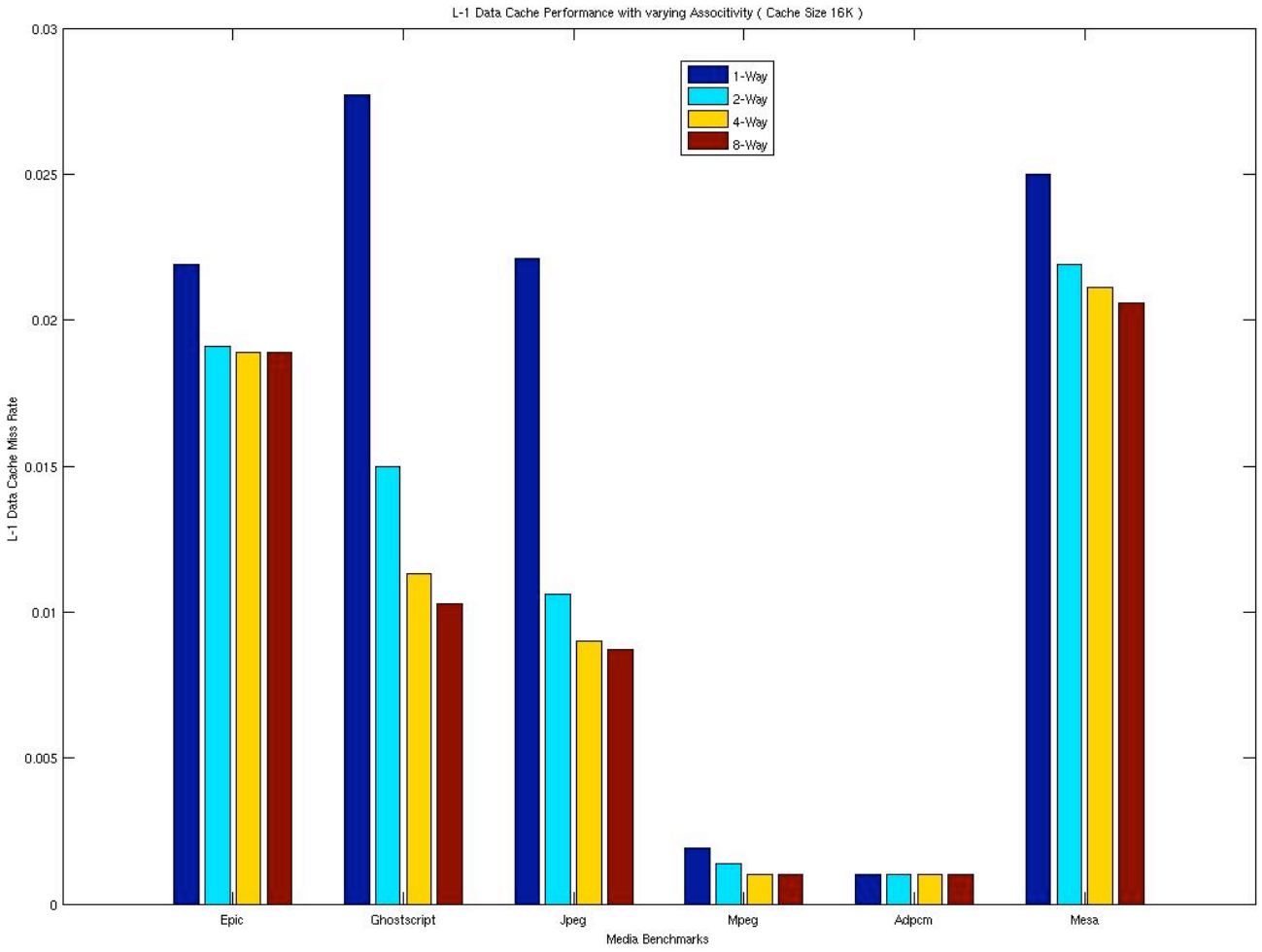
L1 Data Cache Performance

32B Line Size, 2-way, varying cache sizes



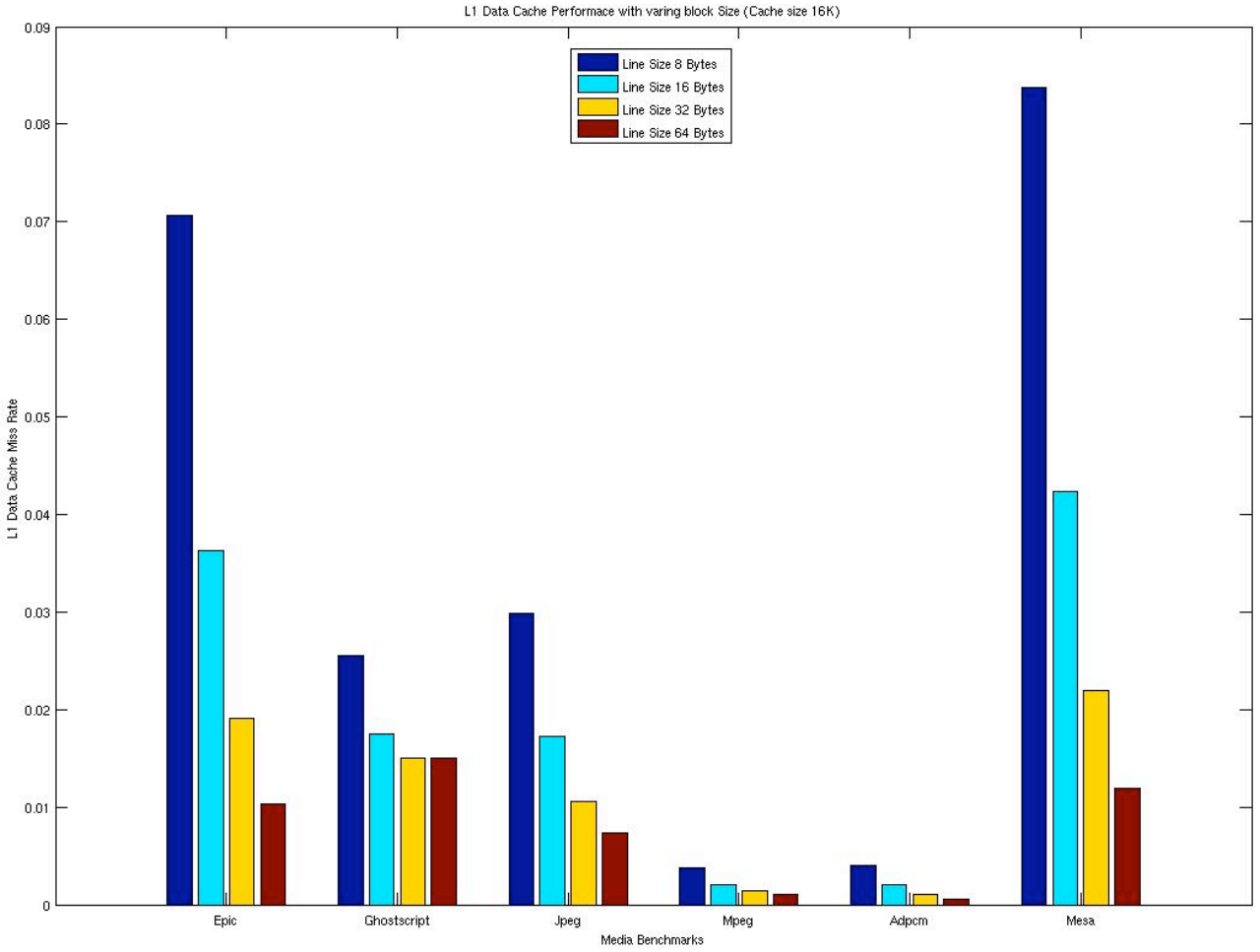
L1 Data Cache Performance

16KB: 32B Block Size, varying associativity



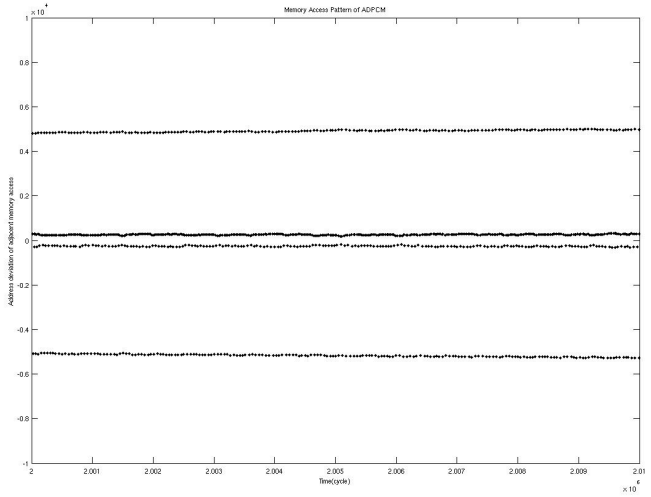
L1 Data Cache Performance

16KB: 2-way, varying line sizes

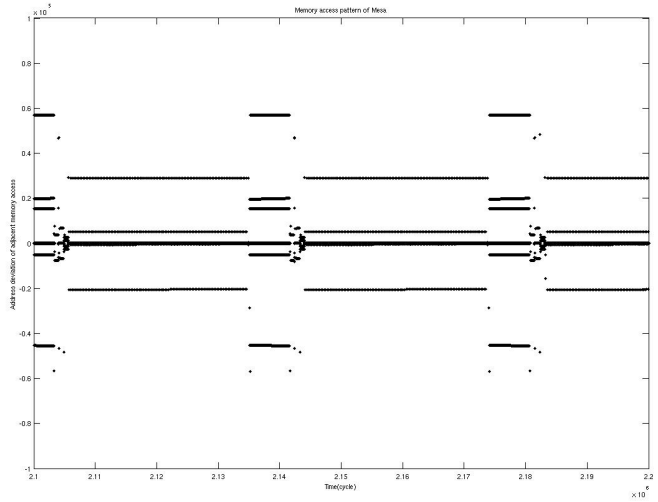


Memory Access Pattern

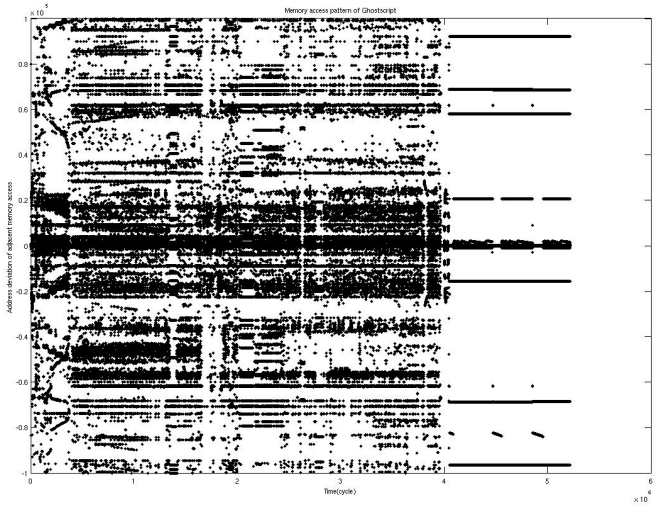
ADPCM



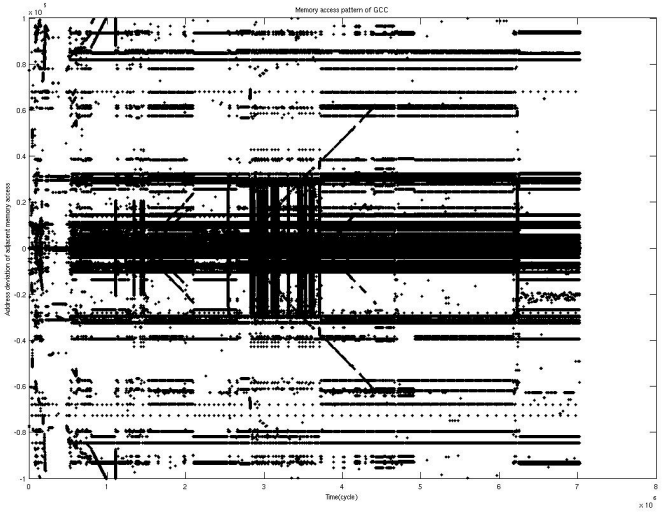
Mesa



Ghostscript



Gcc



B

Summary of Cache Performance

- Generally, media applications have better cache performance than SPEC 2000 programs.
- L-1 Cache performance of media applications is sensitive to line size: the bigger, the better
- Different media applications have significantly different cache performances
  - Image, video and audio applications have lower cache miss rate than 3D and document applications.
- Compared with SPEC programs, media applications have simpler memory access pattern.

Hypothesis

- By using pre-fetch, we can improve the cache performance of media application due to their regular memory access pattern.

Prefetch

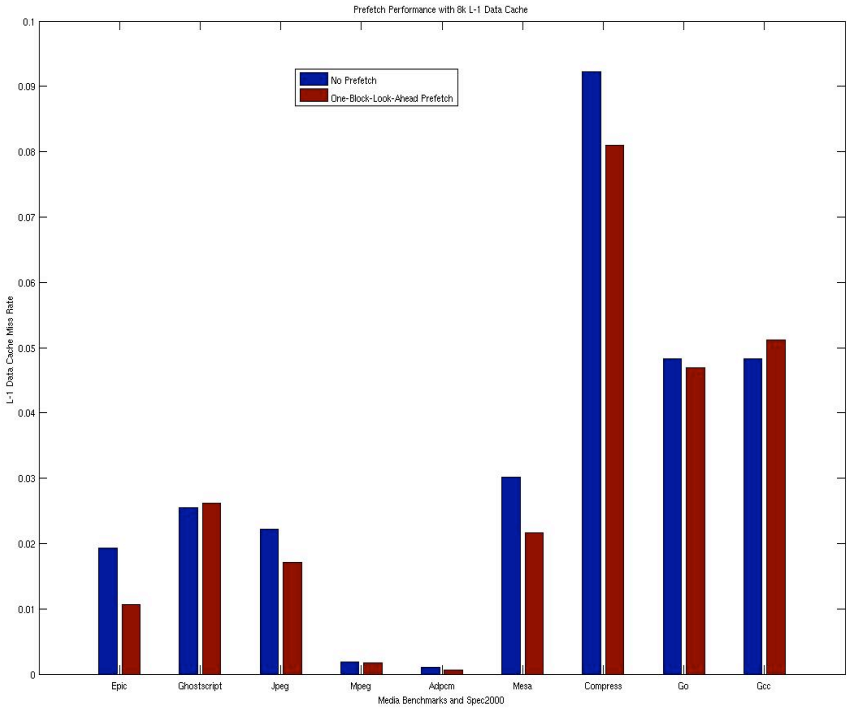
- One block look-ahead
- Different memory access model
  - Pipelined Memory: adopted originally by SimpleScalar
  - Sequential Memory Access: Modify Simplescalar
- Different priority policies in Sequential Memory Access model
  - FIFO
  - Missed Block First

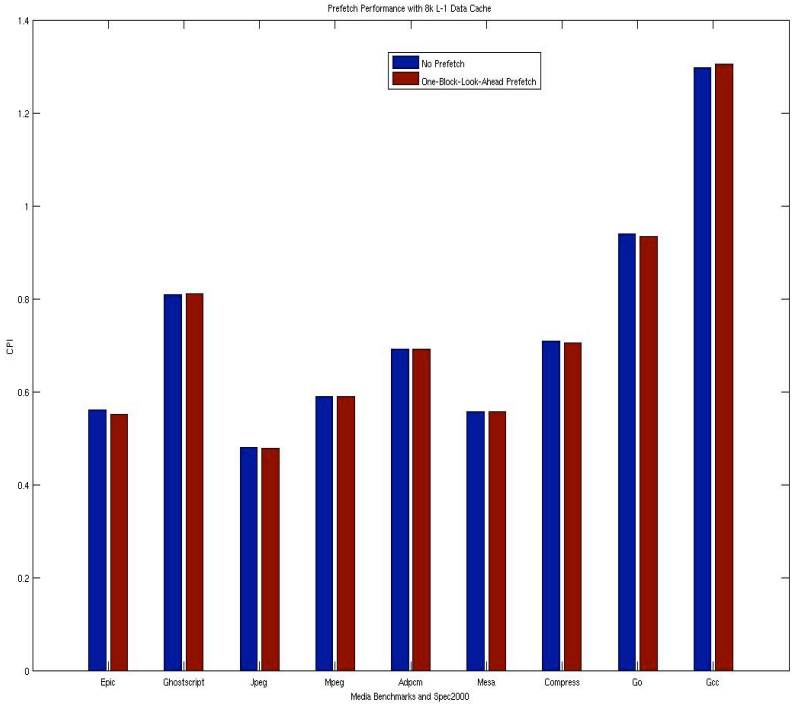
Simulation setting

- 8KB, 16KB, 32KB Data L1 cache, 2 way, Line size: 32 byte
- 256KB L2 cache, 4 way with 64 bytes line
- L1 hit latency: 1 cycle
- L2 hit latency: 6 cycles
- Memory access latency: 26 cycles
- Cache replacement: LRU

Evaluation: Pipelined Memory Model

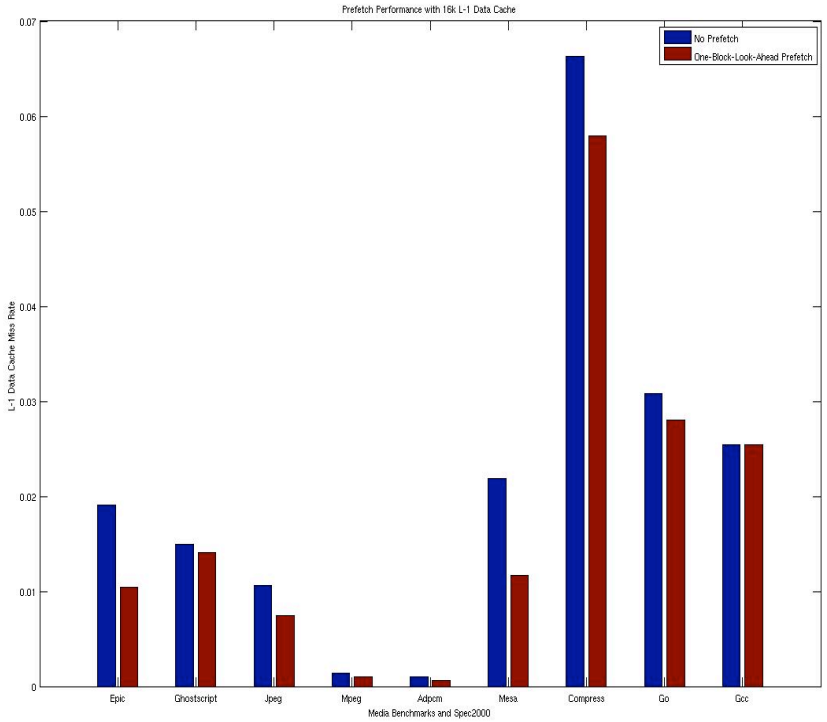
8KB: 32 Bytes Block, 2-way, Pipelined Memory

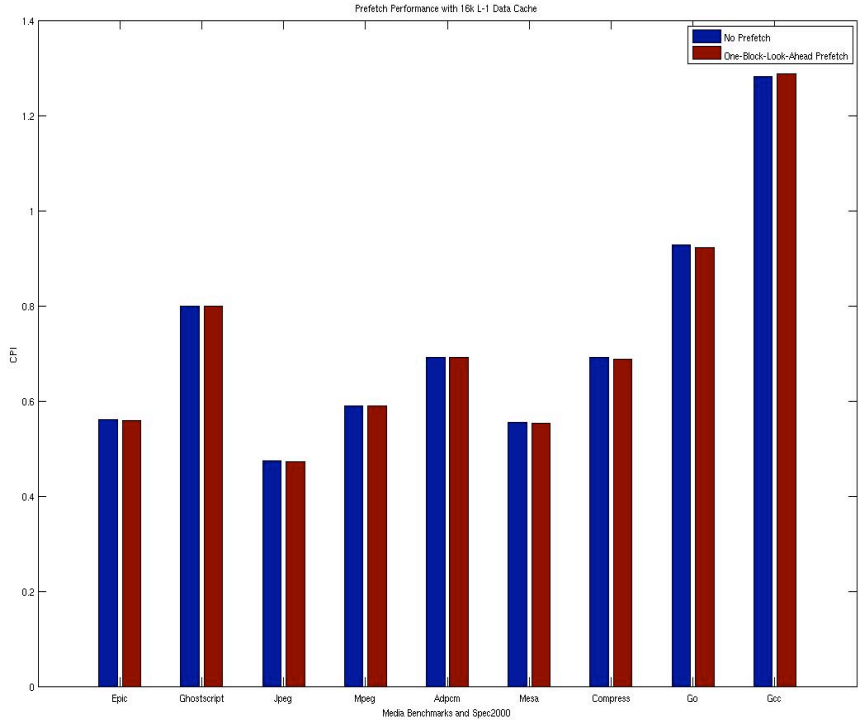




Evaluation: Pipelined Memory Model

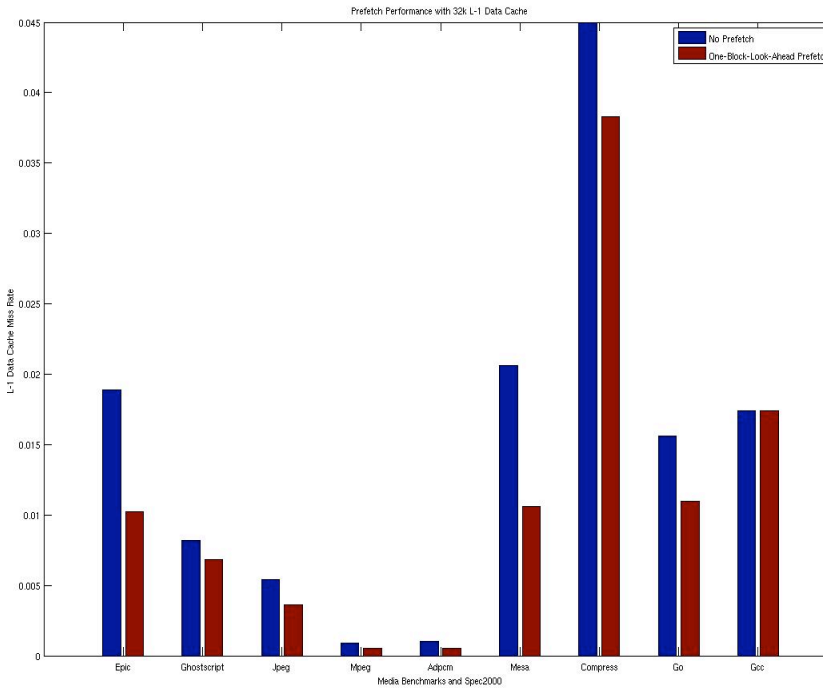
16KB: 32 Byte Block, 2-way, Pipelined Memory

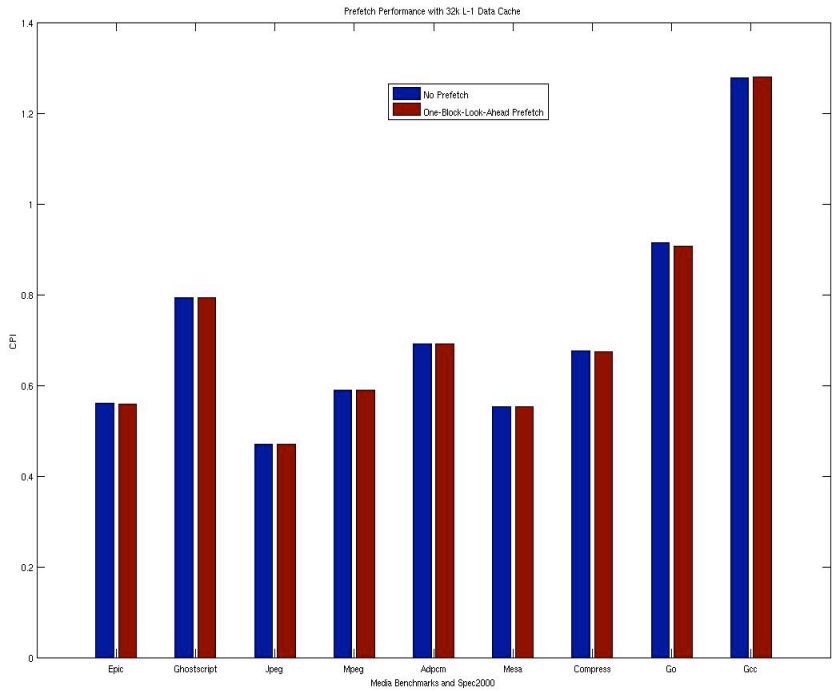




Evaluation: Pipelined Memory Model

32KB:32 Byte Block, 2-way, Pipelined Memory



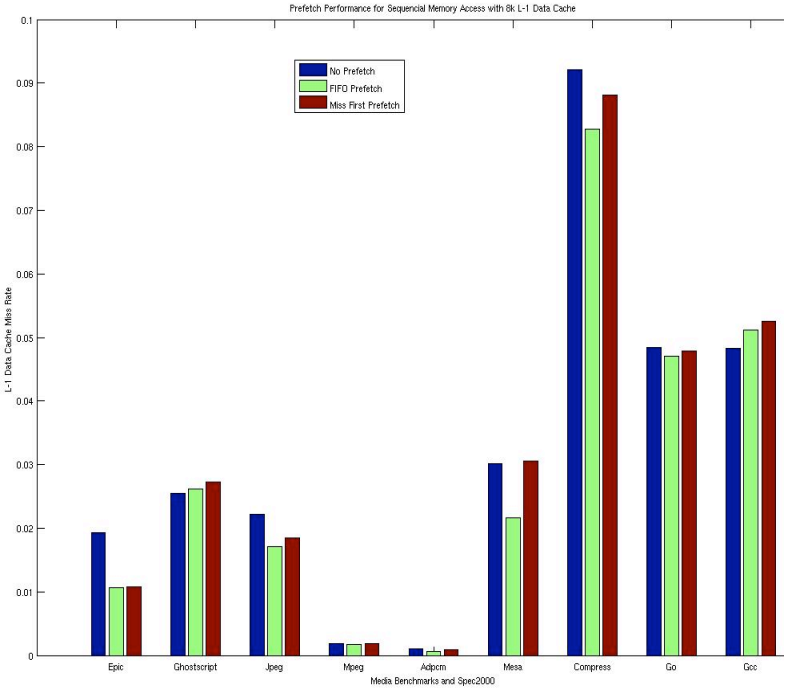


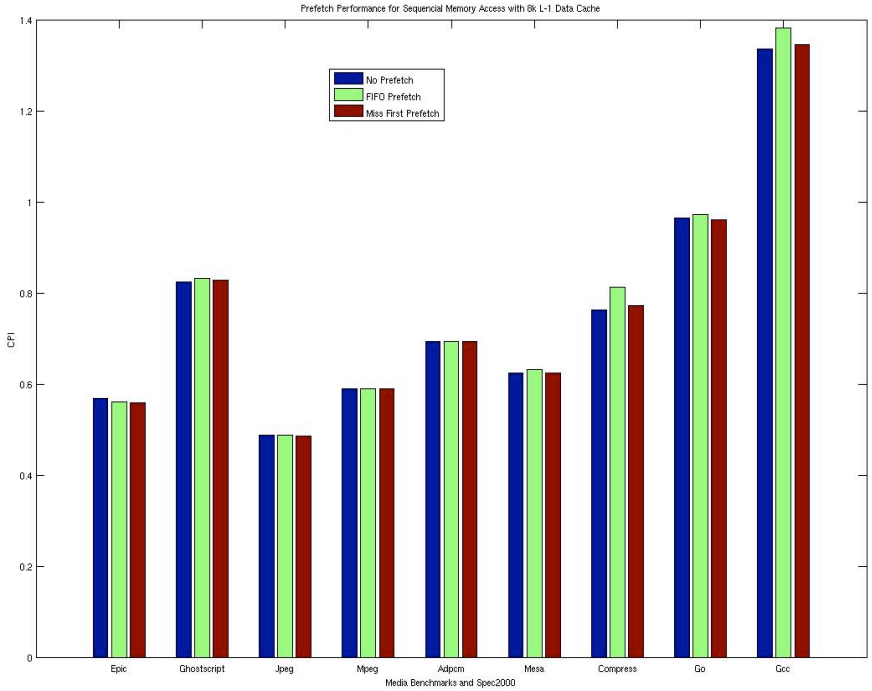
Summary-prefetch with pipelined memory

- Miss rates of Epic, Mpeg, Jpeg, ADPCM decrease by using the simple one-block-look-ahead prefetching.
  - Due to their regular memory access patterns.
- The miss rate of Ghostscript increases slightly.
  - Its memory access pattern is not very regular.
- CPI of media applications also decrease slightly

Evaluation: Sequential Memory Access

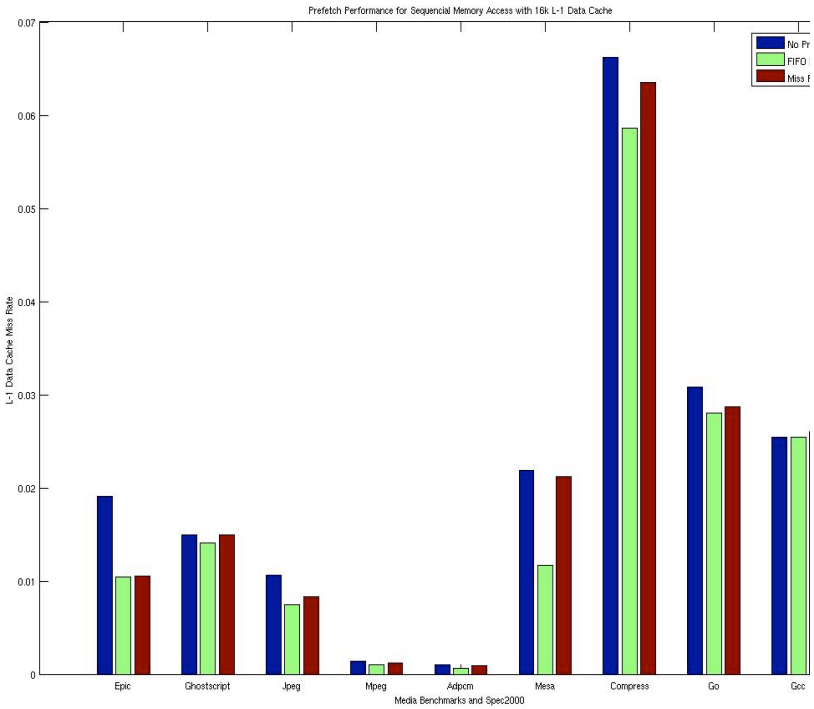
8KB: 32 Bytes Block, 2-way, Sequential Memory

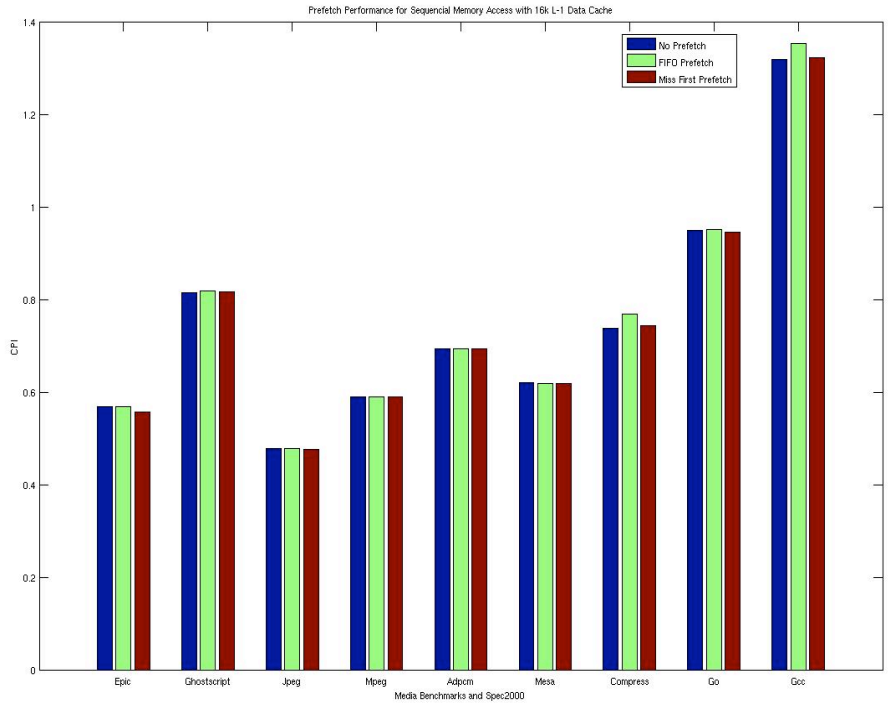




Evaluation: Sequential Memory Access

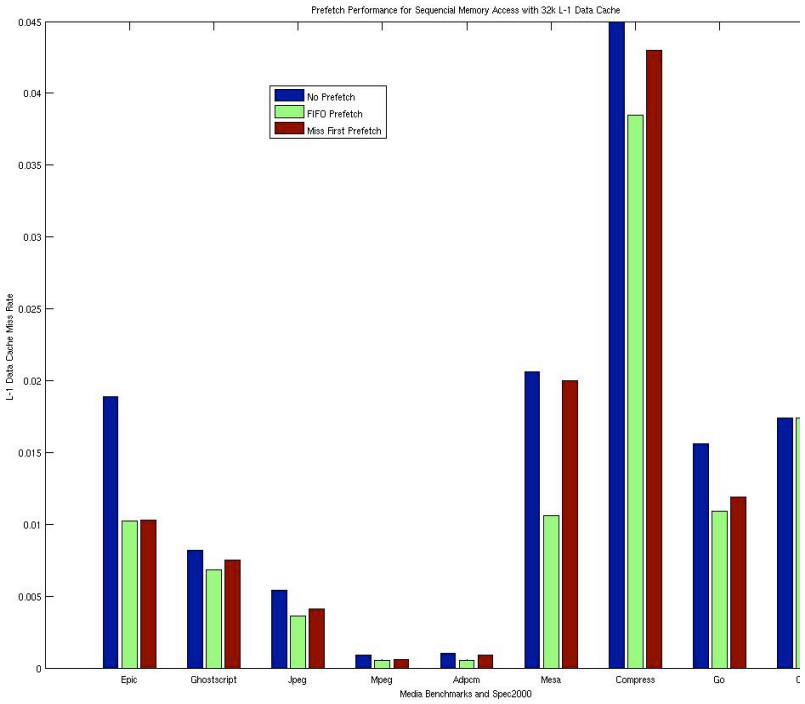
16KB:32 Bytes Block, 2-way, Sequential Memory

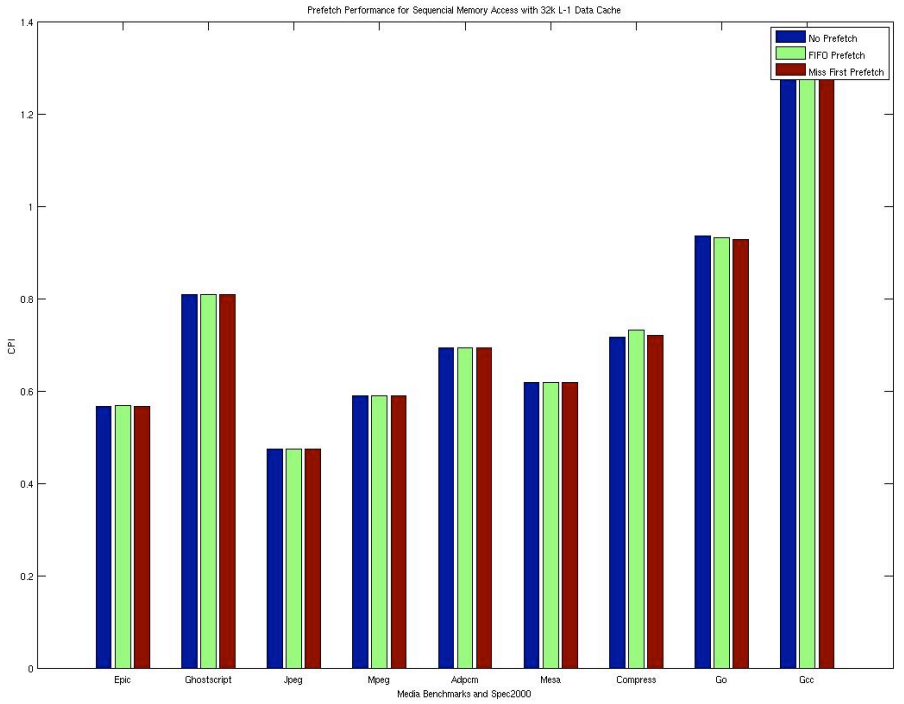




Evaluation: Sequential Memory Access

32KB: 32 Bytes Block, 2-way, Sequential Memory





Summary-prefetch with sequential memory

- For applications with regular memory access pattern, miss rate decreases.
- For applications without regular pattern, the simple prefetch scheme can bring more cache missing (data pollution).
- CPI can be increased because prefetching defers the data fetching after cache miss.

Conclusion

- Cache performance of media applications is not as bad as the common belief.
- Media applications demonstrate very different memory access pattern and cache performance.
- Using simple pre-fetch can decrease the miss rate for media applications, but there isn't big benefit in terms of CPI.

Future Work

- Deeply understanding of memory access pattern of media applications.
- Smart hardware and software prefetching techniques.
- Improving media application performance from computational unit.

- Thanks