Understanding The Effects of Wrong-path Memory References on Processor Performance

Onur Mutlu

Hyesoon Kim

David N. Armstrong

Yale N. Patt

The University of Texas at Austin

Motivation

- Processors spend a significant portion of execution time on the wrong path
  - 47% of all cycles are spent fetching instructions on the wrong path for SPEC INT 2000 benchmarks
- Many memory references are made on the wrong path
  - 6% of all data references are wrong-path references
  - 50% of all instruction references are wrong-path references
- We would like to understand the effects of wrong-path memory references on processor performance
  - The goal is to build hardware/software mechanisms that take advantage of this understanding

Questions We Seek to Answer

1. How important is it to correctly model wrong-path memory references?
    - How does memory latency and window size affect this?

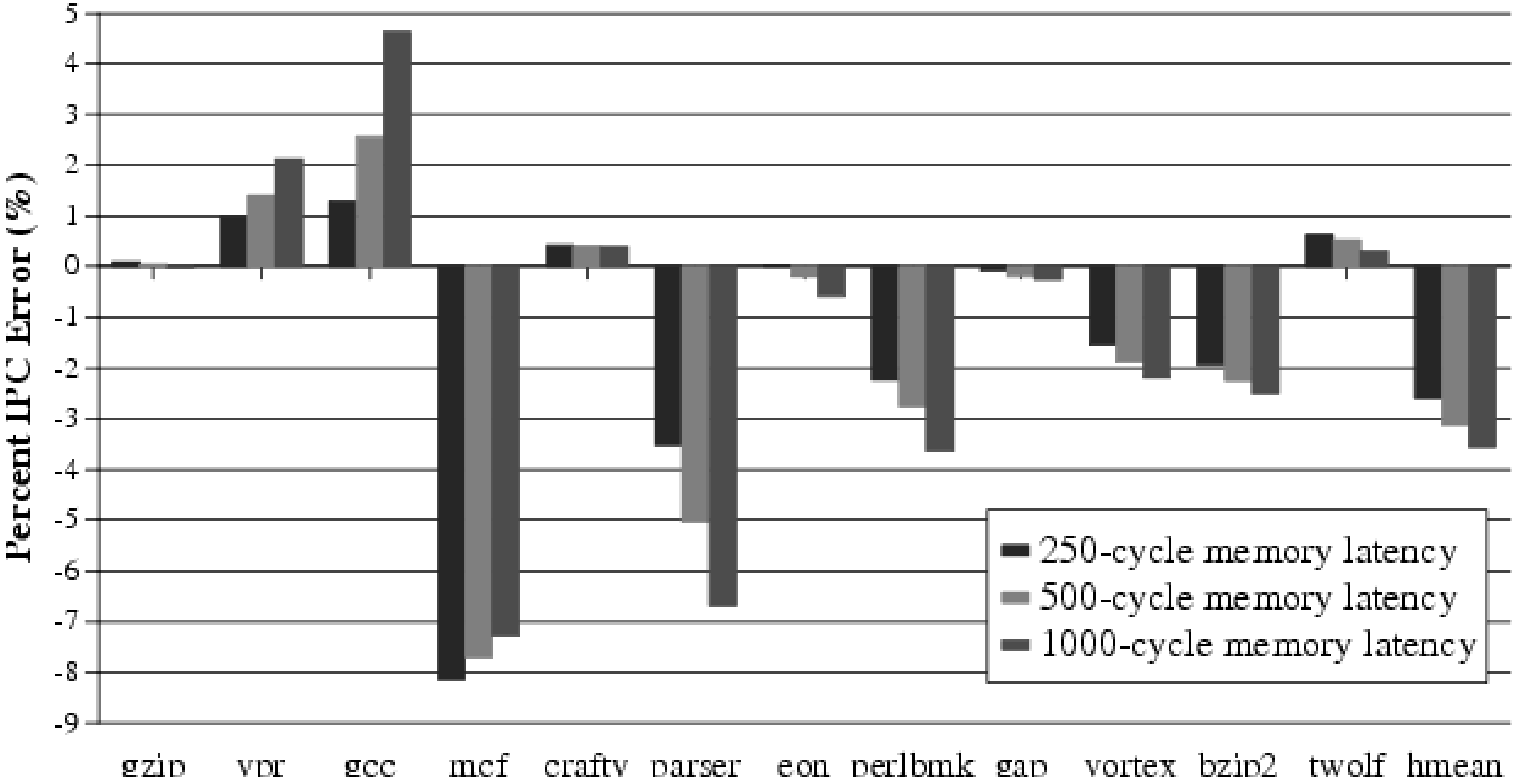
2. What is the relative significance of negative and positive effects of wrong-path memory references on performance?
    - Negative: Cache pollution and bandwidth/resource contention
    - Positive: Prefetching

3. What kind of code structures lead to the positive effects of wrong-path memory references?

Experimental Methodology

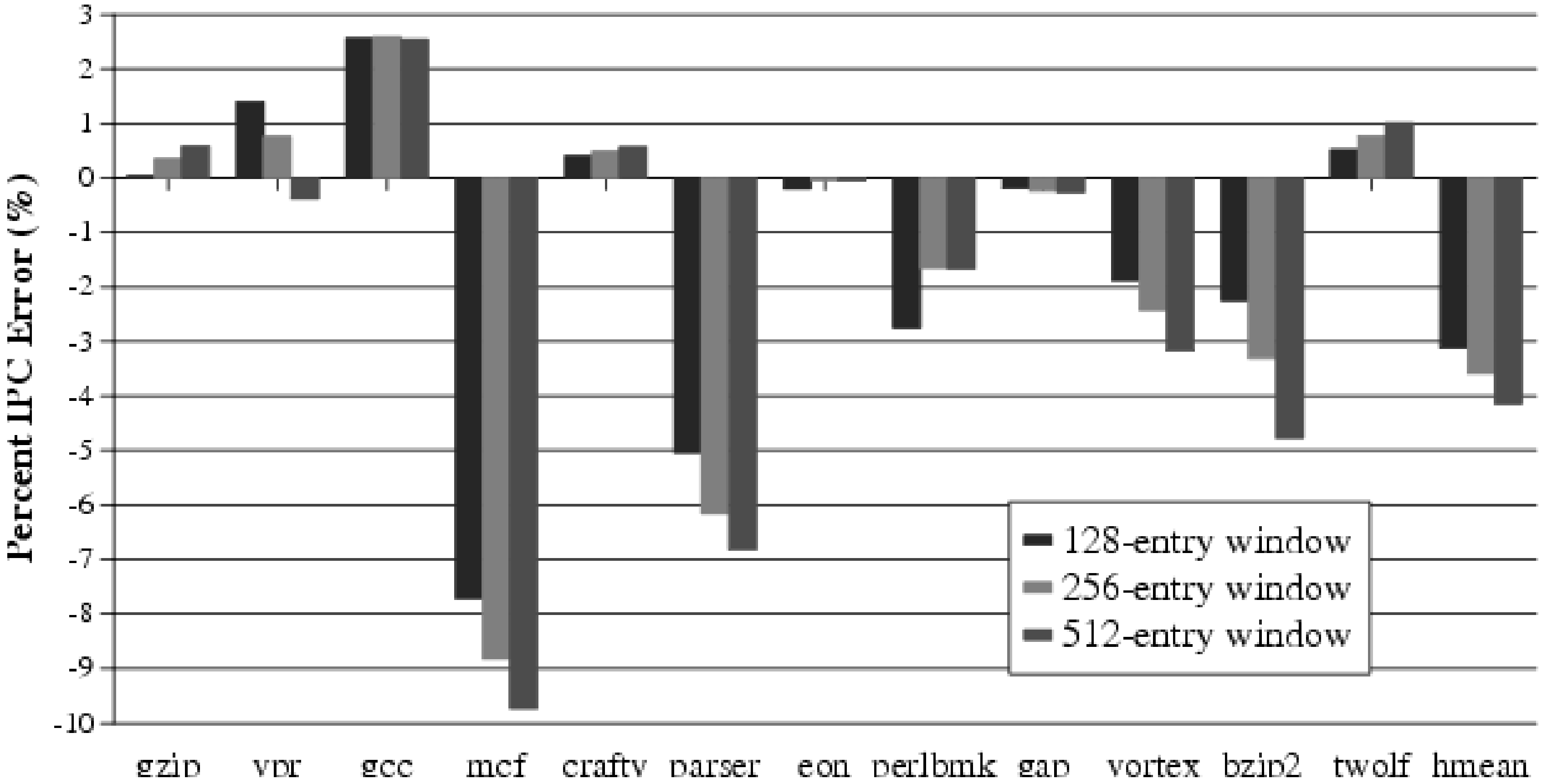
- Execution-driven simulator, accurate wrong-path model
- Cycle-accurate, aggressive memory model
- Baseline processor
  - Models the Alpha ISA
  - 8-wide fetch, decode, issue, retire
  - 128-entry instruction window
  - Hybrid conditional branch predictor
    - 64K-entry PAs, 64K-entry gshare, 64K-entry selector
  - Aggressive indirect branch predictor (64K-entry target cache)
  - 20-cycle branch misprediction latency
  - 64 KB, 4-way, 2-cycle L1 data and instruction caches
    - Maximum 128 outstanding L1 misses
  - 1 MB, 8-way, 10-cycle, unified L2 cache
  - Minimum 500-cycle L2 miss latency
- 12 SPEC INT 2000 benchmarks evaluated

Error in IPC if Wrong-path References are not Modeled: The Effect of Memory Latency



- Negative error means wrong-path references are beneficial for performance.
- Maximum error is 7%, average error is 3.5% for a 500-cycle memory latency
- In general, error increases as memory latency increases

Error in IPC if Wrong-path References are not Modeled: The Effect of Instruction Window Size



- Negative error means wrong-path references are beneficial for performance
- Maximum error is 10%, average error is 4% for a 512-entry window
- In general, error increases as instruction window size increases

Insights

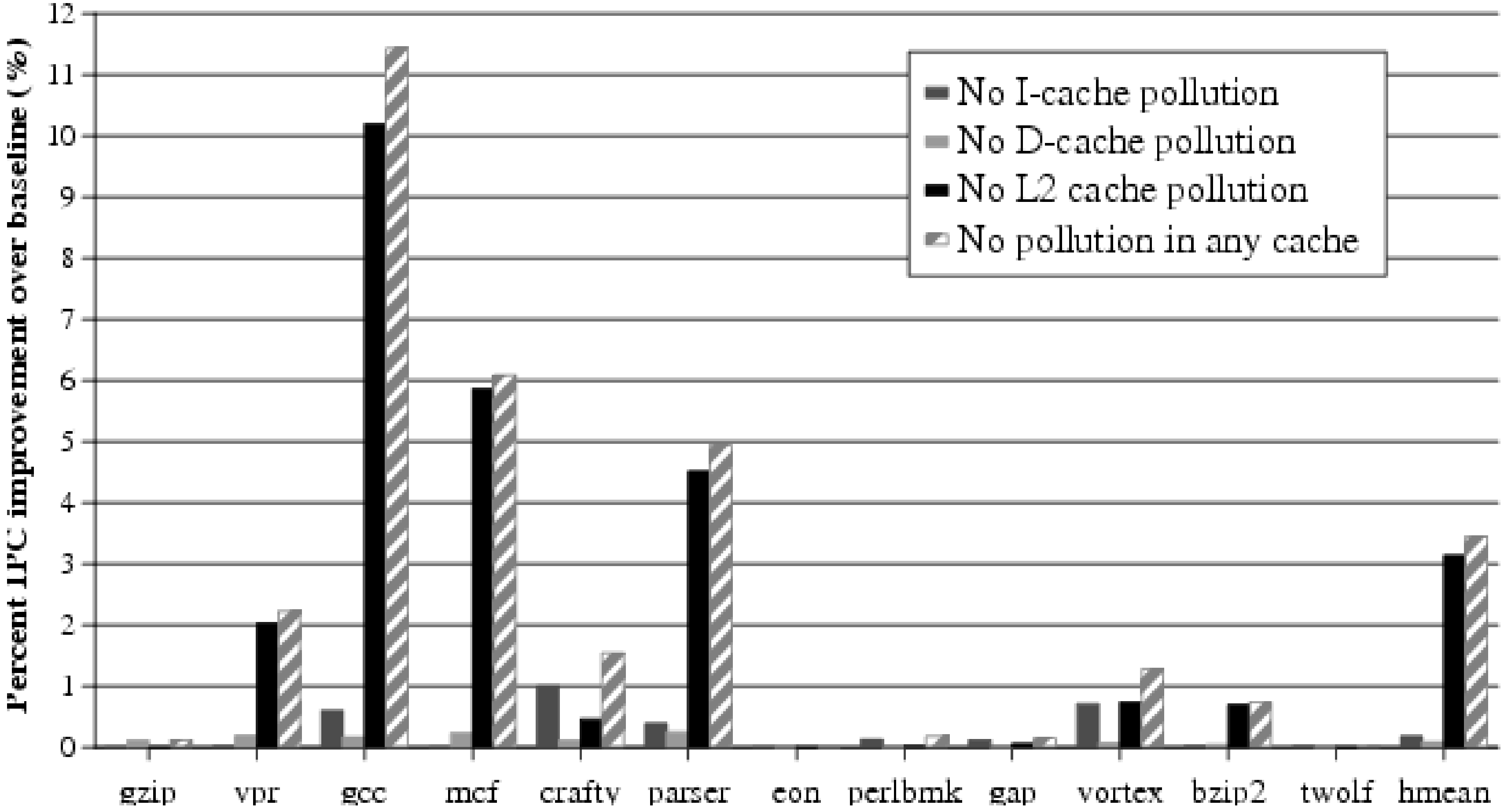
- Wrong-path references are important to model (to avoid errors of up to 10%)
- Wrong-path references usually have a positive impact on IPC performance.
- Wrong-path references have a negative impact on performance for a few benchmarks (notably gcc and vpr)
  - We would like to understand why.

Negative Effects of Wrong-path References

1. Bandwidth and resource contention
   - Our simulations show this effect is not significant.

2. Cache pollution
   - To evaluate this effect, we examine four idealized models:
     - No I-cache pollution: Wrong-path references cause no pollution in the instruction cache
     - No D-cache pollution: Wrong-path references cause no pollution in the data cache
     - No L2 cache pollution: Wrong-path references cause no pollution in the L2 cache
     - No pollution in any cache: Wrong-path references cause no pollution in any of the caches
   - We compare the performance of these models to the baseline model which correctly models wrong-path references

IPC Improvement if Pollution Caused by Wrong-path References is Eliminated



- L1 pollution does not significantly affect performance.
- L2 pollution is the most significant negative effect of wrong-path references.
  - L2 pollution is the cause of performance loss in gcc and vpr.

Insights

- Why is bandwidth/resource contention not a problem?
  - Enough bandwidth/resources in the memory system to accommodate the wrong-path references.

- Why is L1 pollution not significant?
  - L1 misses due to pollution are short-latency misses (10 cycles)
  - These latencies are tolerated by the instruction window.

- Why is L2 pollution the dominant negative effect?
  - L2 misses incur a very high penalty (500 cycles)
  - L2 miss latency cannot be tolerated by the instruction window.
  - Hence, an L2 miss very likely results in a full-window stall on the correct path.

Positive Effects of Wrong-path References: Prefetching

- A cache miss caused by a wrong-path reference is useful if the data brought by the miss is later accessed by the correct-path before being evicted.

- Most wrong-path cache misses are useful
  - 76% of all wrong-path data cache misses are useful
  - 69% of all wrong-path instruction cache misses are useful
  - 75% of all wrong-path L2 misses are useful

- But why?
  - We would like to understand the code structures that result in useful wrong-path data prefetches.

Code Structures that Cause Useful Wrong-path Data References

1. Prefetching for later loop iterations
   - Wrong-path execution of a loop iteration can prefetch data for the correct-path execution of the same iteration
   - Most useful wrong-path prefetches in mcf and bzip2 are generated this way

2. One loop prefetching for another
   - Wrong-path execution of a loop can prefetch data for the correct-path execution of another loop, if they are both working on the same data structure

3. Prefetching in control-flow hammocks
   - Wrong-path execution of the taken path can prefetch data for the correct-path execution of the not-taken path

Prefetching for Later Loop Iterations (an example from mcf)

```
1 :   arc_t *arc; // array of arc_t structures
2 :   // initialize arc (arc = ...)
3 :
4 :   for ( ; arc < stop_arcs; arc += size) {
5 :         if (arc->ident > 0) { // frequently mispredicted branch
6 :               // function calls and
7 :               // operations on the structure pointed to by arc
8 :               // ...
9 :         }
10:   }
```

- Processor mispredicts the branch on line 5 and does not execute the body of the if statement on the wrong path. Instead, next iteration is executed on the wrong path.

- This next iteration initiates a load request for `arc->ident`, which misses the data cache.

- When the mispredicted branch is resolved, processor recovers and first executes the body of the if statement on the correct path.

- On the correct path, the processor executes the next iteration and initiates a load request for `arc->ident`, which is already prefetched.

Conclusions

- Modeling wrong-path memory references is important.
  - Not modeling them causes errors of up to 10% in IPC estimates.
  - Effect of wrong-path references on IPC increases with increasing memory latency and increasing window size.
- In general, wrong-path memory references are beneficial for performance due to prefetching effects.
- The dominant negative effect of wrong-path references is the L2 pollution they cause in the L2 cache.
- We identified three code structures that are responsible for the prefetching benefits of wrong-path memory references.

Backup Slides

Related Work

- Butler [1993] was the first one to realize that wrong-path references can be beneficial for performance.
- Moudgill et. al. [1998] investigated the performance impact of wrong-path references with a 40-cycle memory latency. They found that IPC error is negligible if wrong-path references are not modeled.
- Pierce and Mudge [1994] studied the effect of wrong-path references on cache hit rates. They found wrong-path references can increase correct-path cache hit rates.
- Bahar and Albera [1998] proposed the use of a small fully-associative buffer at the L1 cache level to reduce the pollution caused by wrong-path references. Unfortunately, they assumed a priori that wrong-path references degrade performance.

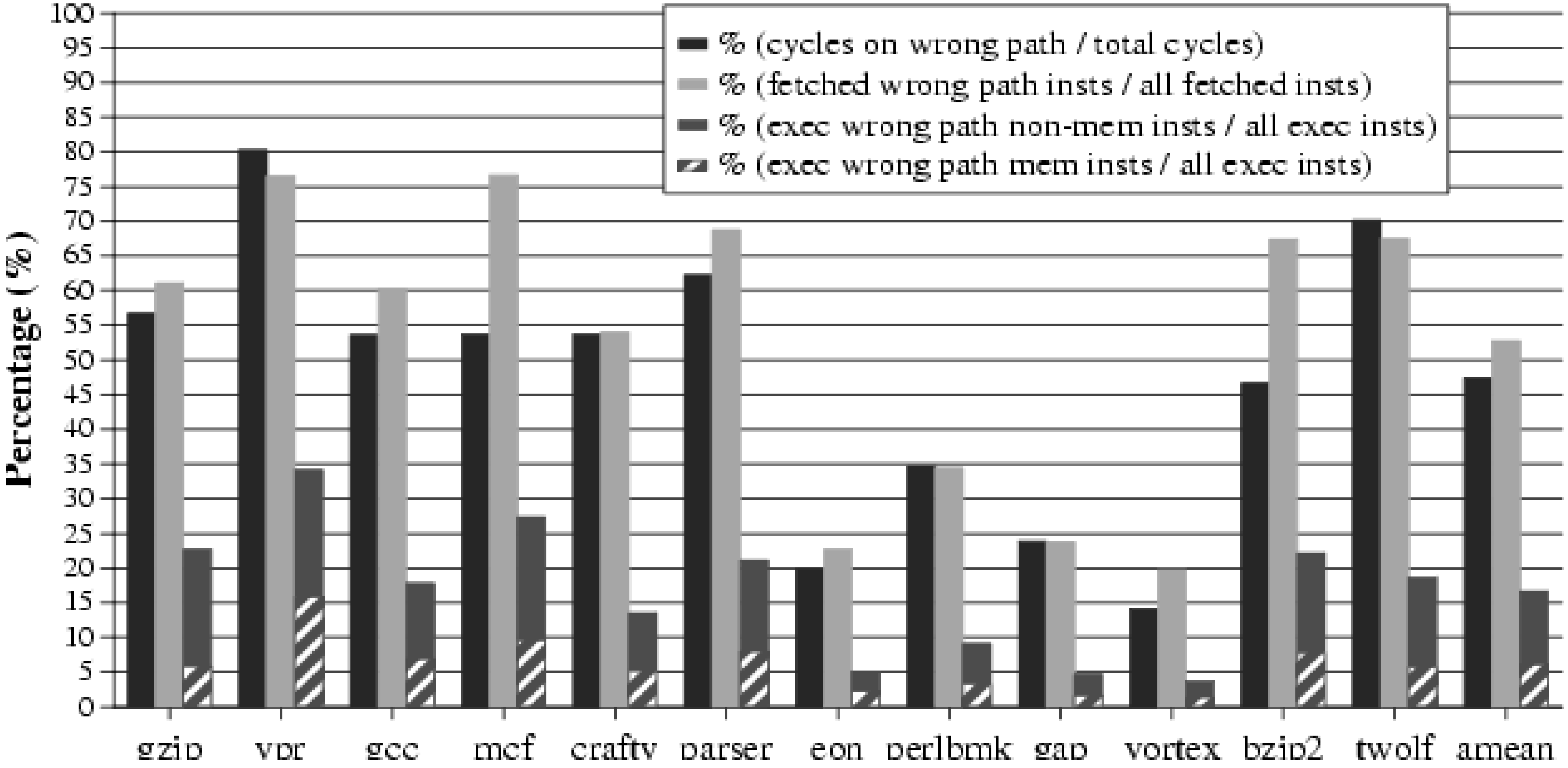
- We build on previous work by focusing on *understanding* the relative significance of different negative and positive effects of wrong-path references. We focus on IPC performance instead of cache hit rates.
- We also aim to understand *why* wrong-path references are useful by identifying the code structures that cause the positive effects of wrong-path references.

Future Research Directions

- Techniques to reduce L2 cache pollution due to wrong-path references
  - Caching wrong-path data in a separate buffer?
  - Predicting the usefulness of wrong-path references?

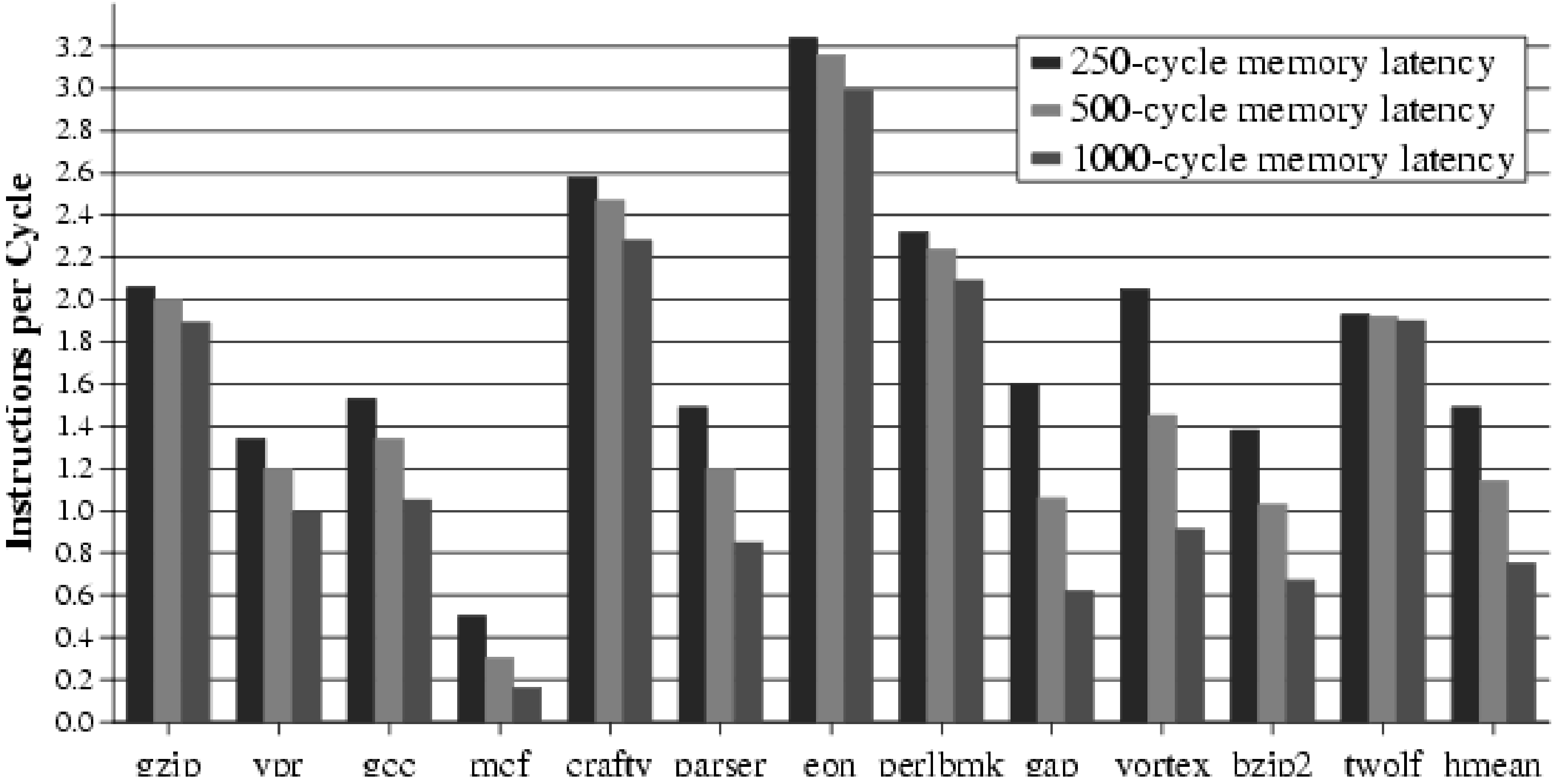
- Techniques to make wrong-path execution more beneficial for performance
  - Can the compiler structure the code such that wrong-path execution always (or usually) provides prefetching benefits?

How Much Time Does the Processor Spend on Wrong -path Instructions?

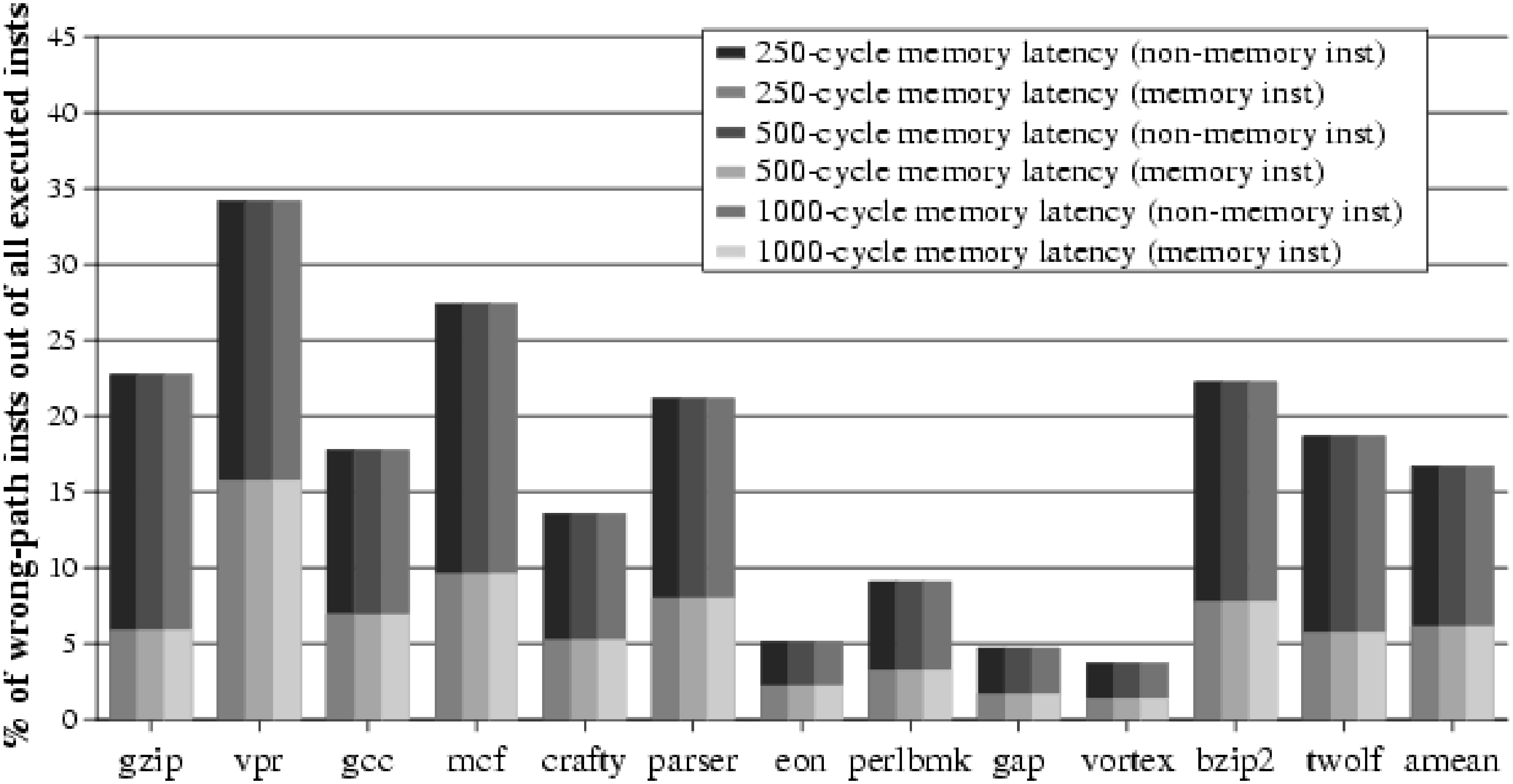


- 47% of all cycles are spent on the wrong path for SPEC INT 2000 benchmarks
- 53% of all fetched instructions and 17% of all executed instructions are on the wrong path

IPC of the Baseline Processor

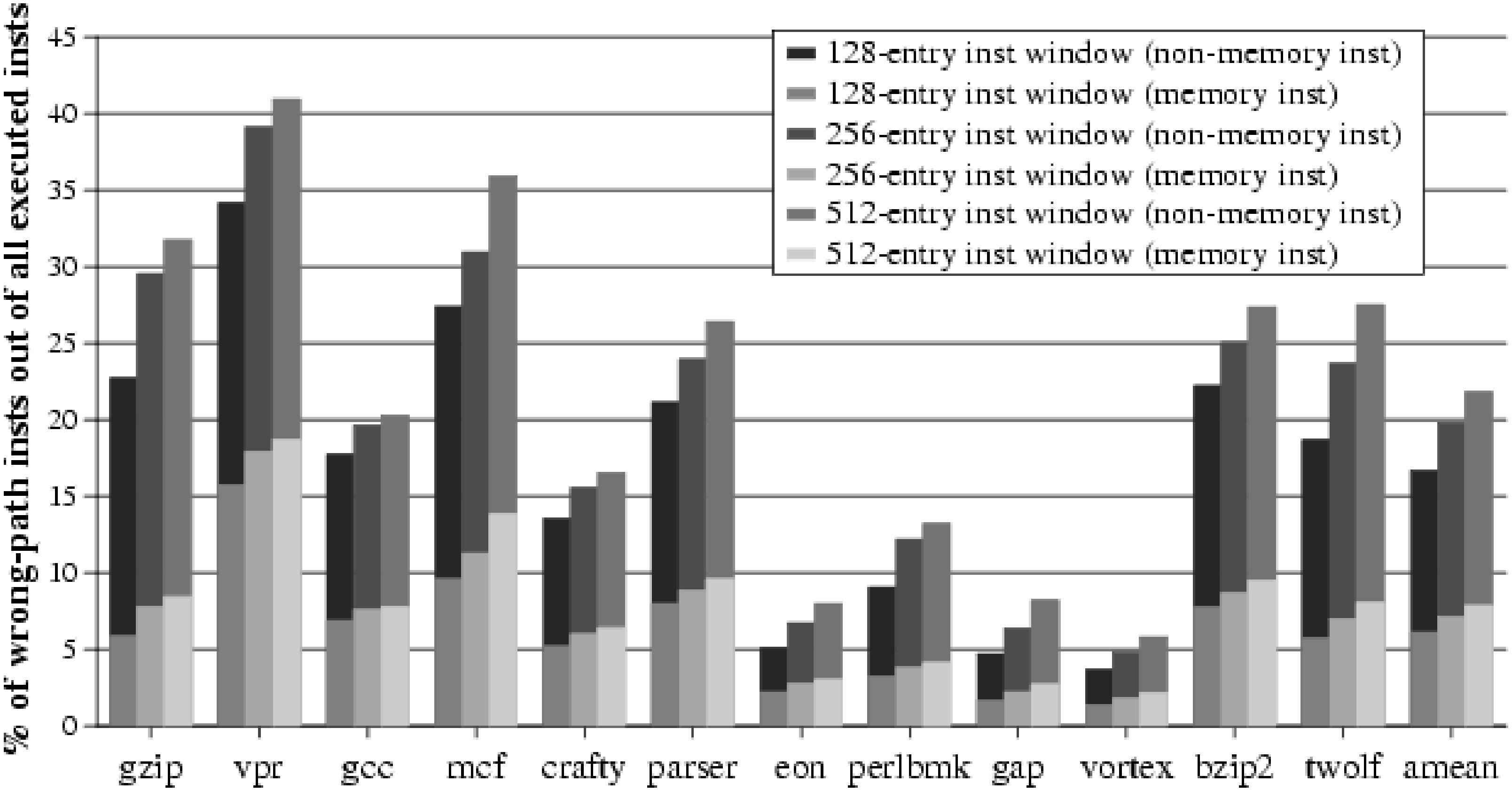


Percentage of Executed Wrong-path Instructions: The Effect of Memory Latency



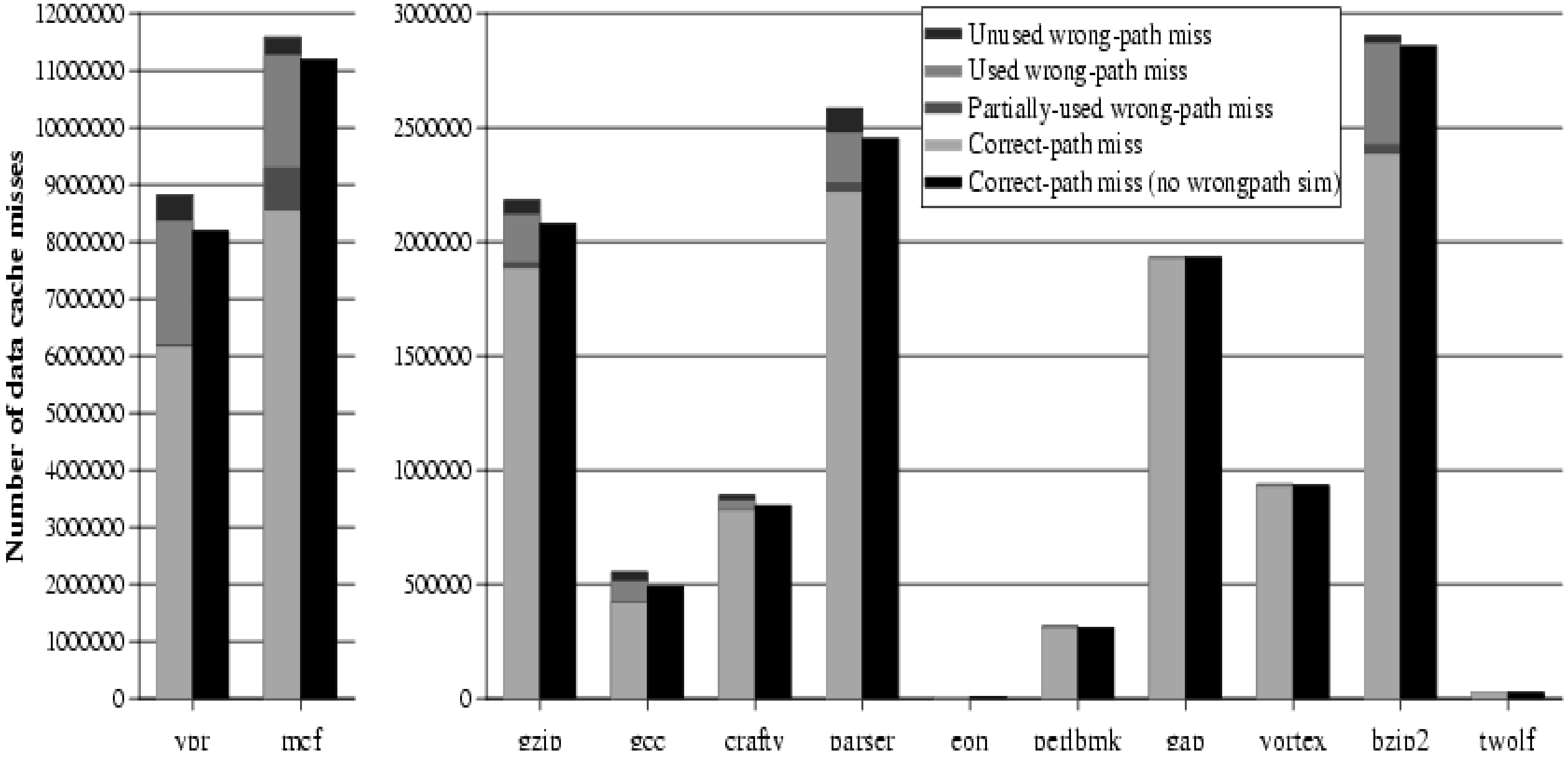
- Increasing the memory latency does not increase the number of executed wrong-path instructions due to the limited (128-entry) instruction window size

Percentage of Executed Wrong-path Instructions: The Effect of Instruction Window Size



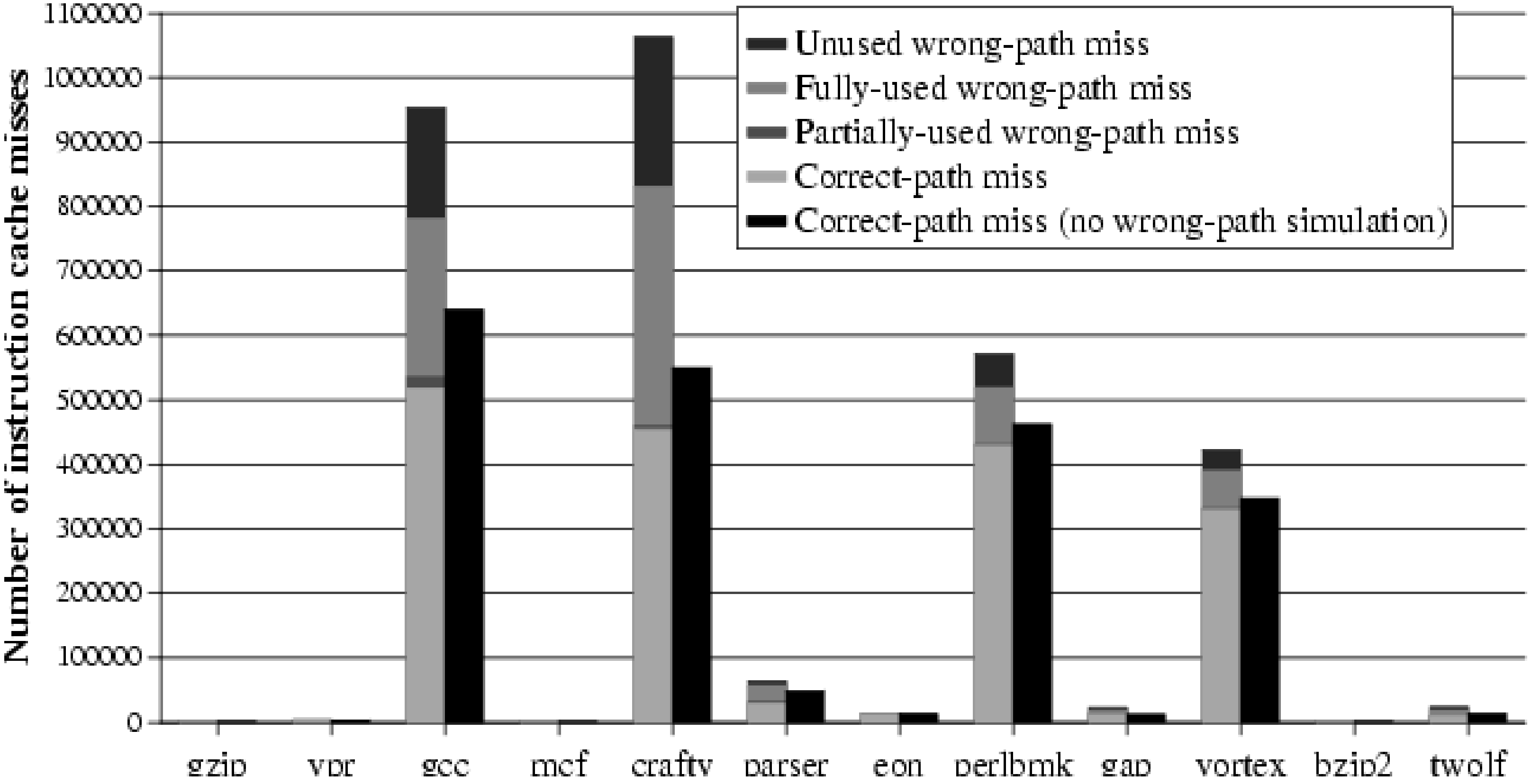
- A larger instruction window is able to execute more instructions (hence, more memory references) on the wrong path

Classification of Data Cache Misses



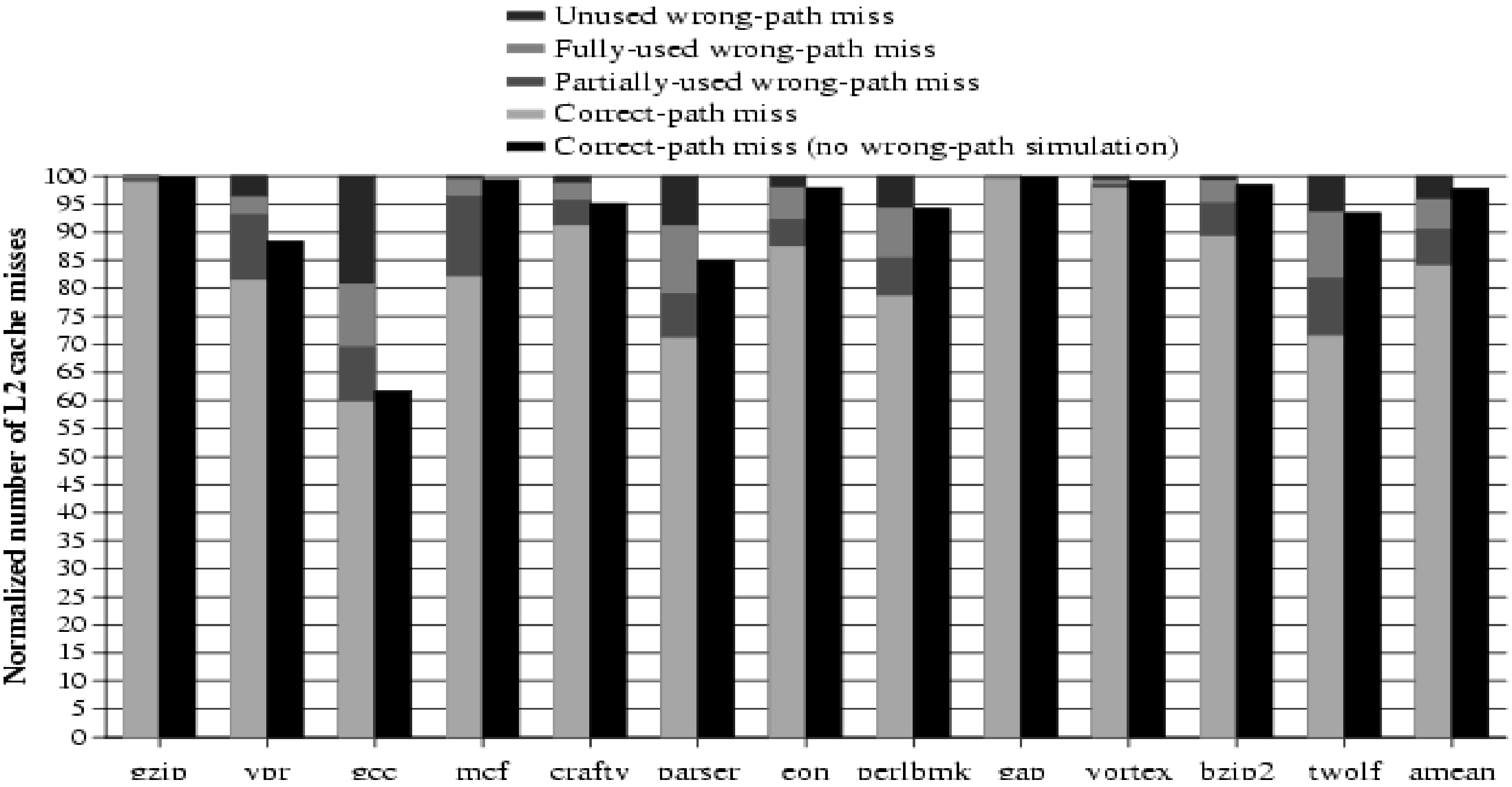
- Right bar: No wrong-path memory references
- Left bar: Wrong-path memory references are correctly modeled
- On average, 76% of the wrong-path data cache misses are fully or partially-used by the correct path.

Classification of Instruction Cache Misses



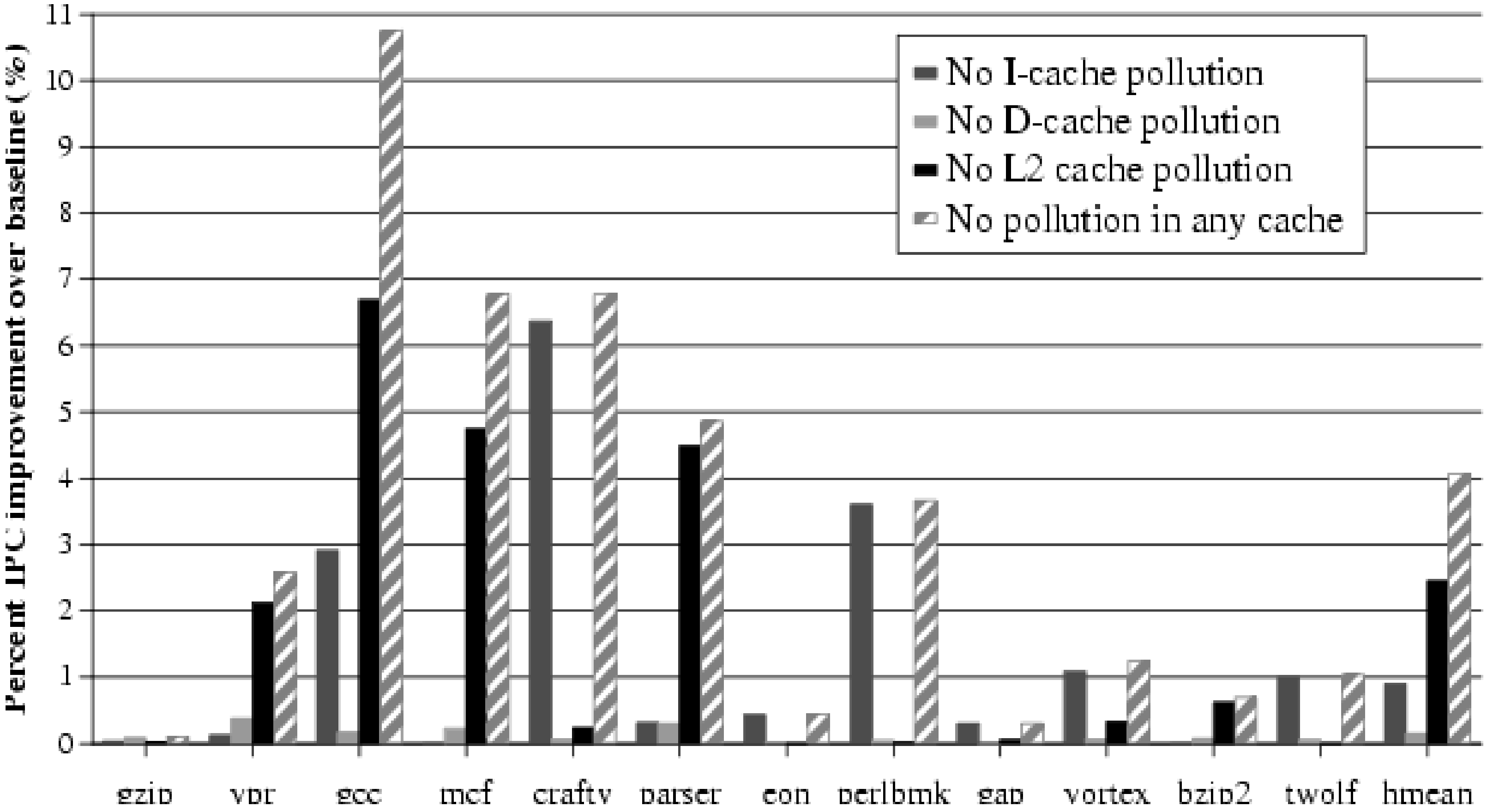
- Right bar: No wrong-path memory references
- Left bar: Wrong-path memory references are correctly modeled
- On average, 69% of the wrong-path instruction cache misses are fully or partially-used by the correct path.

Classification of L2 Cache Misses



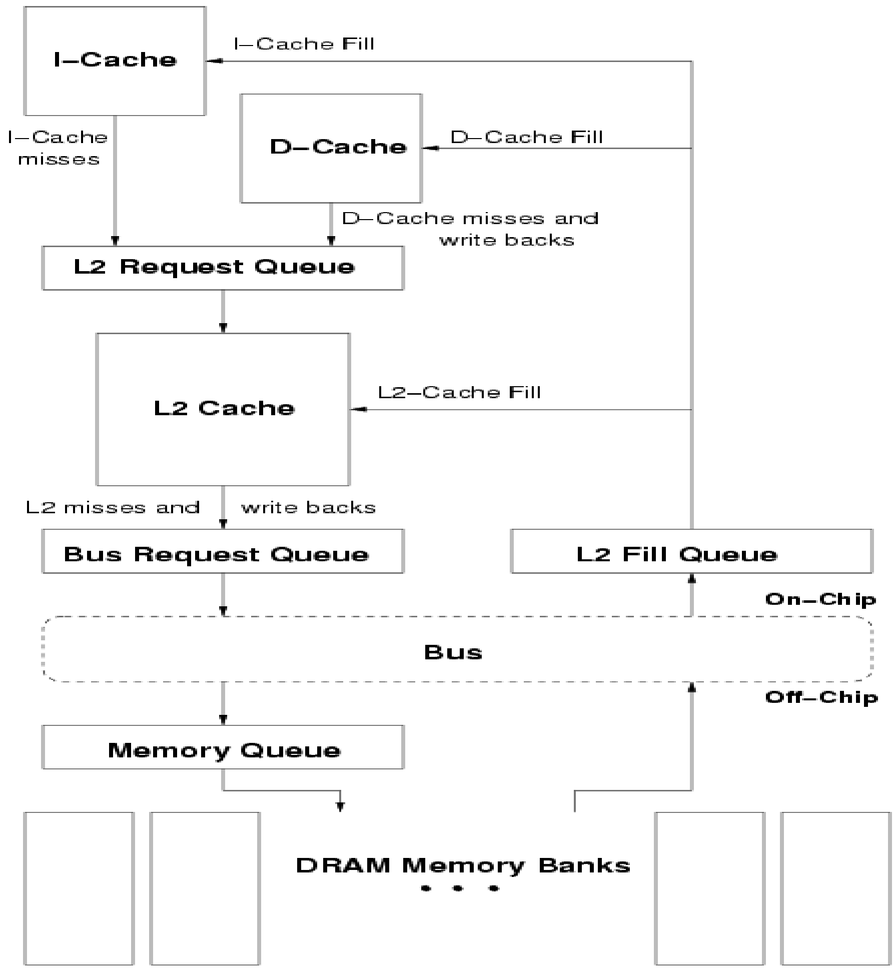
- The number of misses suffered on the correct path (correct-path miss + partially-used wrong-path miss) increase for gcc and vpr if wrong-path memory references are correctly modeled. This is the cause for IPC degradation for gcc and vpr.
- On average, 75% of the wrong-path L2 cache misses are fully or partially-used by the correct path.

IPC Improvement if Pollution Caused by Wrong-path References is Eliminated (16 KB L1 Caches)



- L1 pollution does not significantly affect performance even with 16 KB L1 caches.
- L2 pollution is the most significant negative effect of wrong-path references.

Memory System


I–Cache
I–Cache Fill
D–Cache
D–Cache Fill
I–Cache misses
D–Cache misses and write backs
L2 Request Queue
L2 Cache
L2–Cache Fill
L2 misses and
write backs
Bus Request Queue
L2 Fill Queue
On–Chip
Bus
Off–Chip
Memory Queue
DRAM Memory Banks
• • •