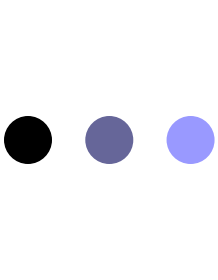
Increasing the Efficiency of Prefetching

weston harper

justin mintzer

adrian valenzuela



The Concept

- Current prefetching techniques can generate multiple simultaneous memory requests
- Having multiple outstanding memory requests provides opportunity to issue group requests

Hypothesis

A major component of communication delay time for prefetch units is bus latency. By adding the ability for memory to handle group requests from prefetch units, the average bus latency per memory access can be significantly reduced.

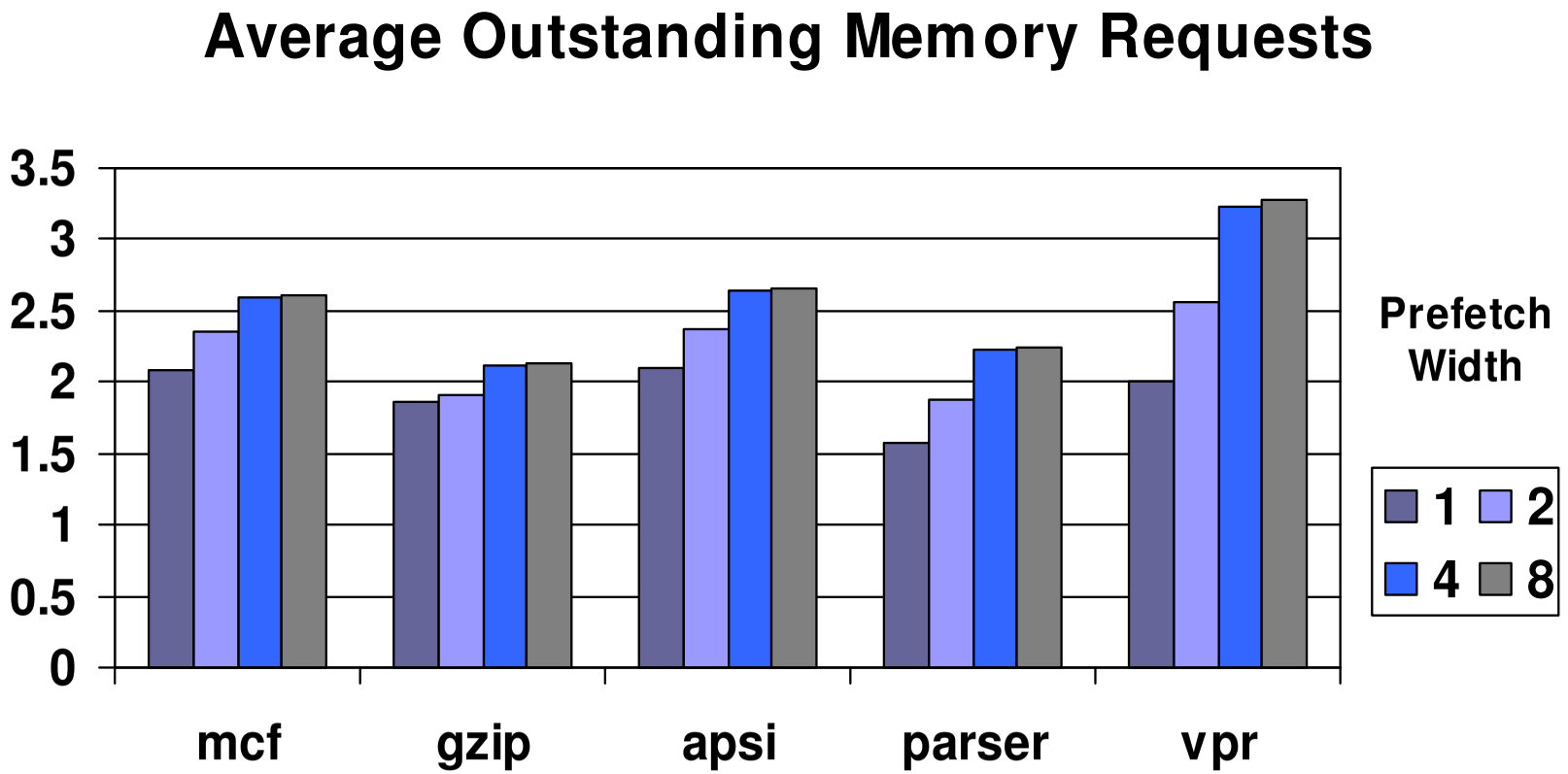
Architecture

- Use a next-n-line prefetcher with the L1 instruction and data cache
- Transmit the next-n requested lines in a single bus transmission via a sparse vector
- The L2 services the lines it can and sends a sparse vector on to the DRAM for the rest of the lines

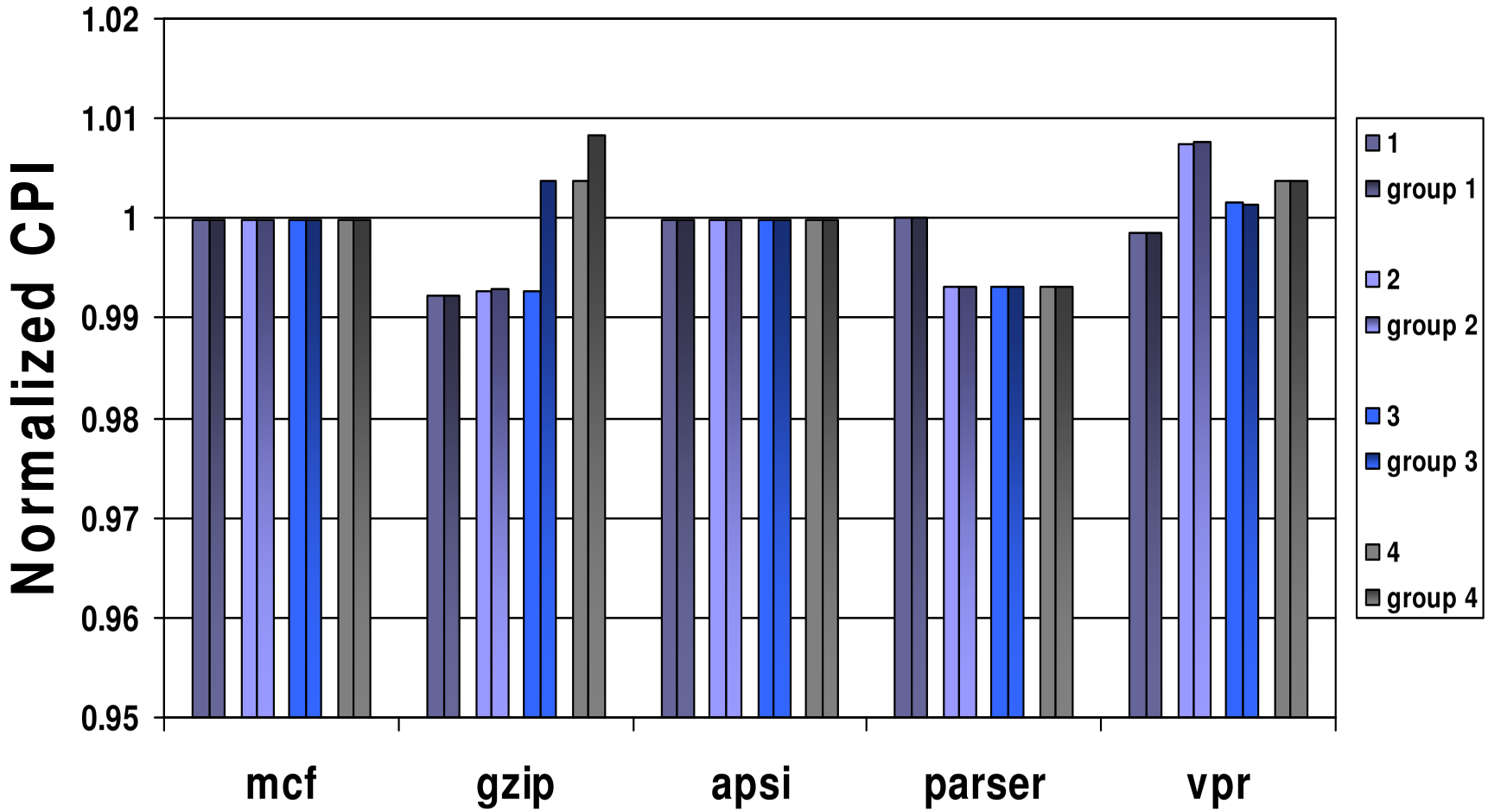
Experimental Setup

- 933 MHz Pentium 3 Linux System
- Running SimpleScalar 4.0
  - sim-alpha
  - SPEC2000
  - default memory system in place
  - next-n-line prefetch

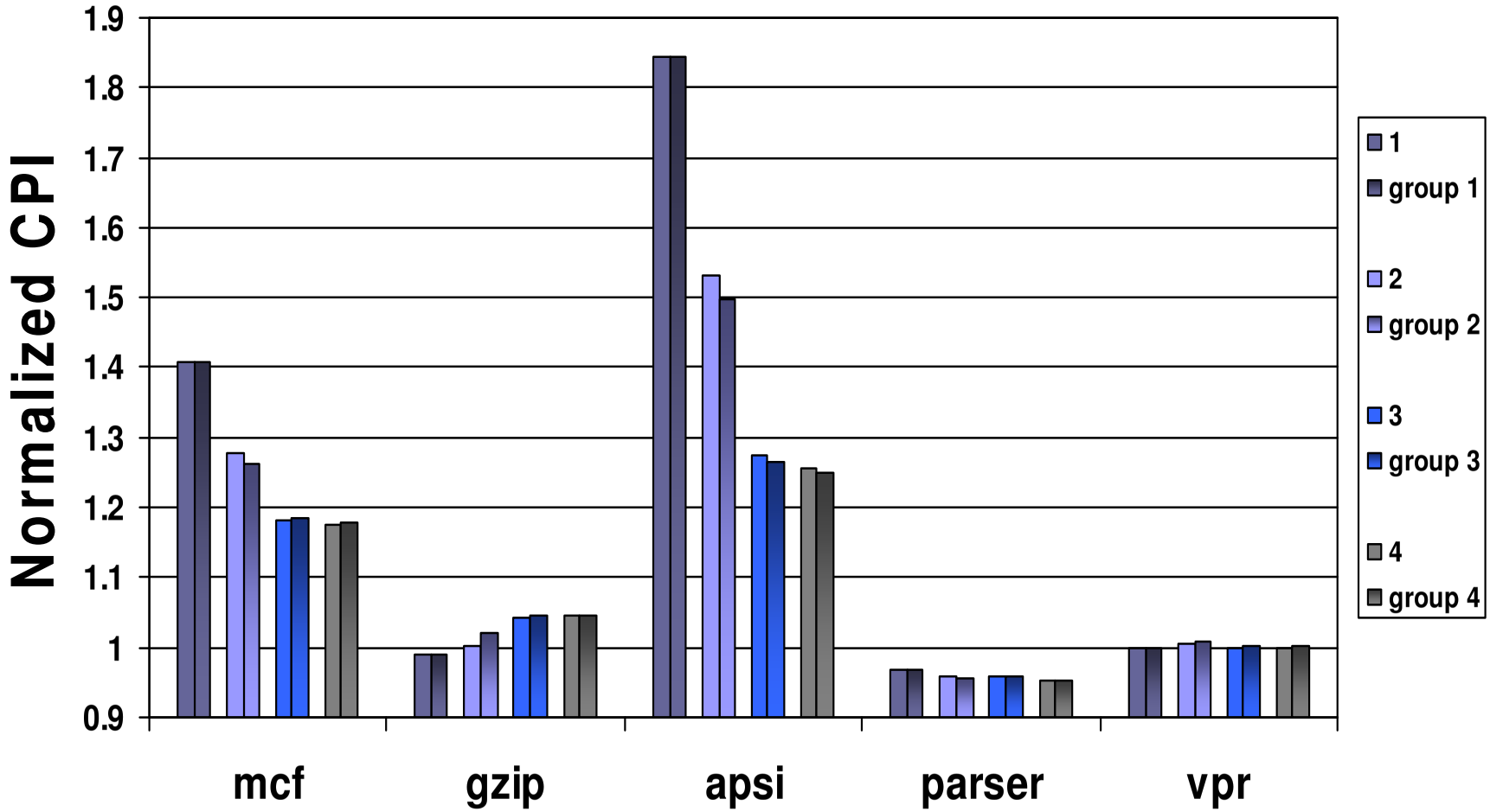
MPI Potential



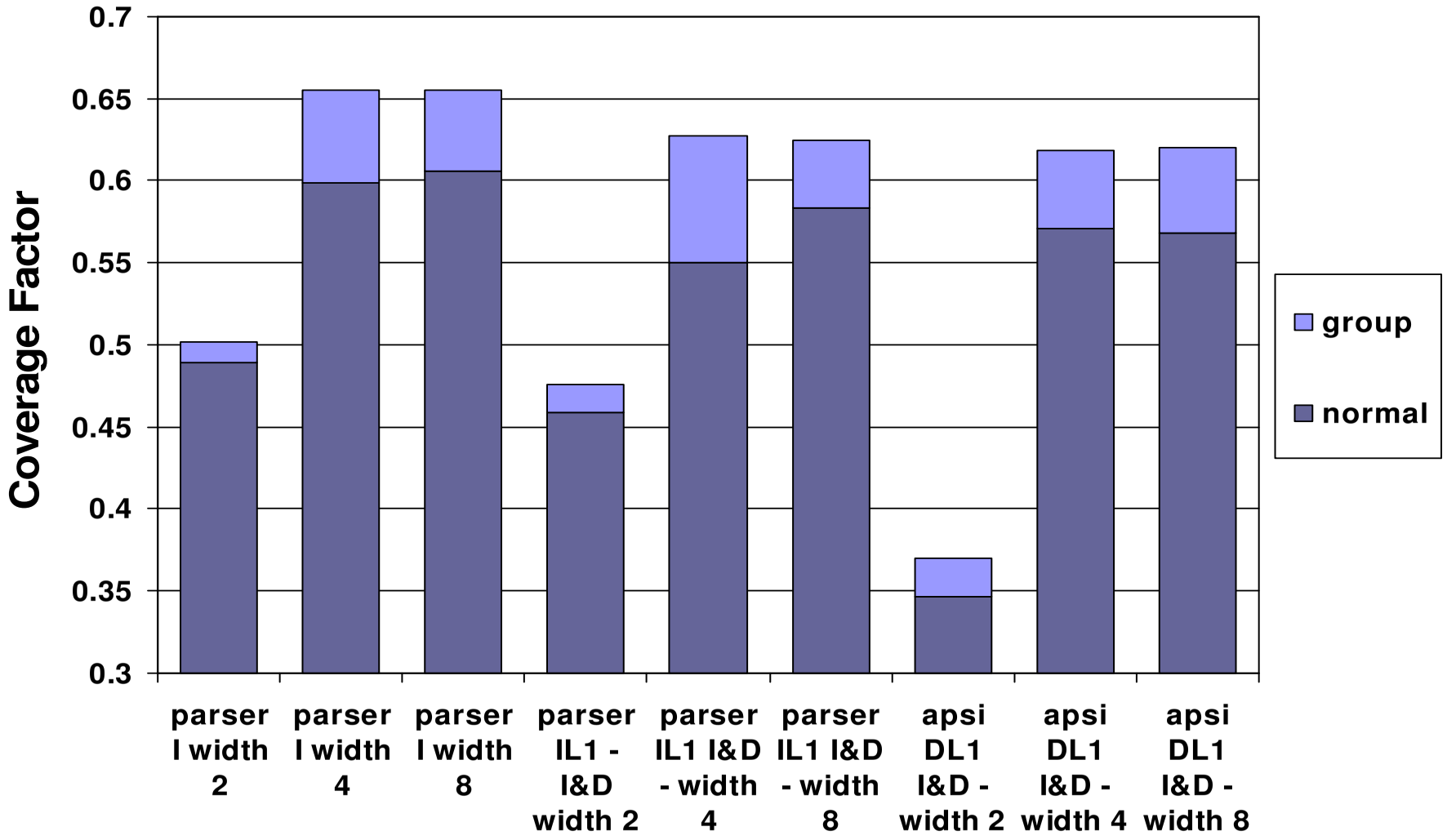
Instruction Prefetch CPI



Data & Instruction CPI



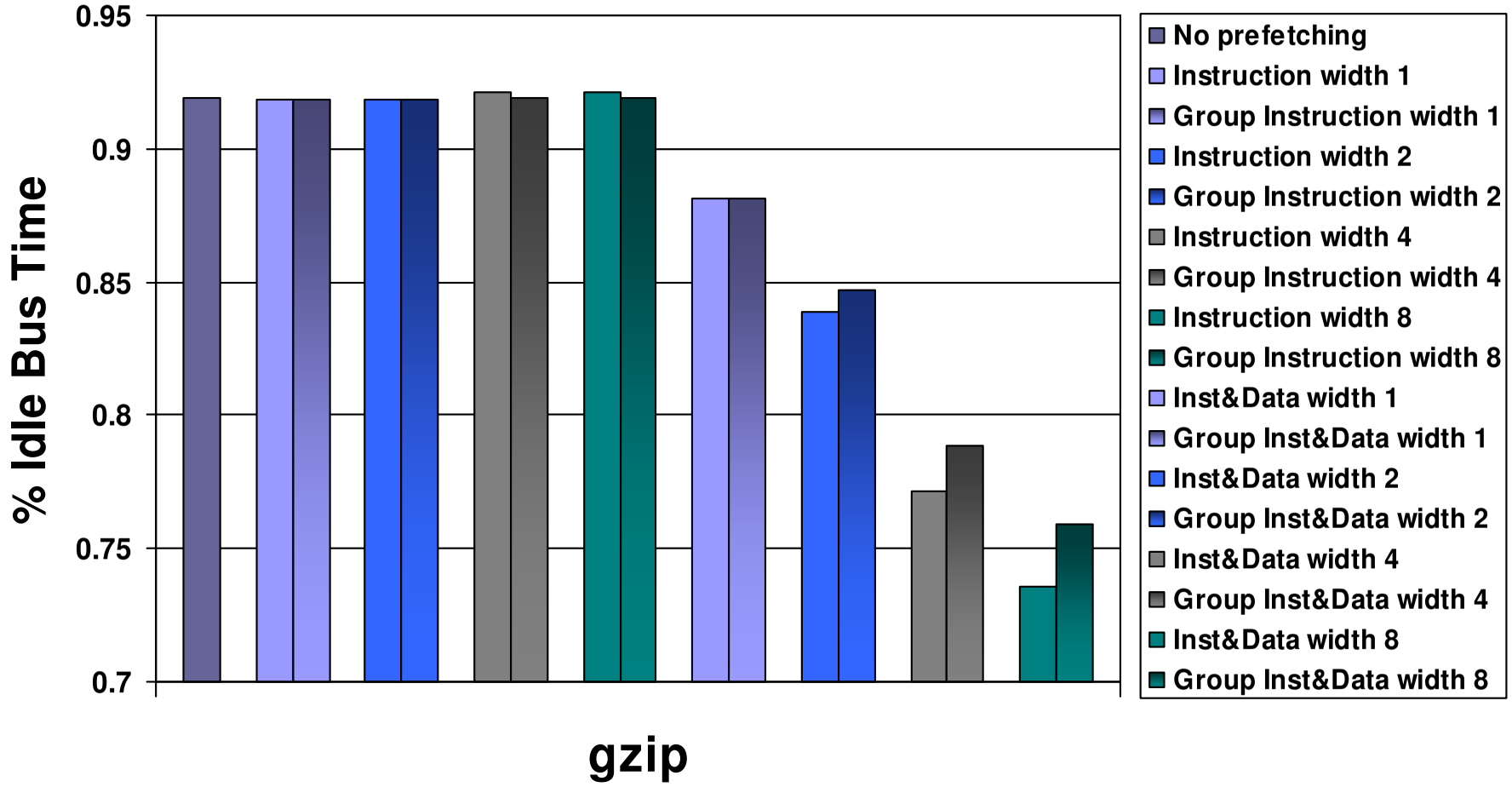
Prefetch Coverage Factor



Prefetch Characteristics

- Number of useless and unnecessary prefetches changes little with group prefetching
- Number of useful prefetches remain nearly the same except for data prefetching where group prefetching is much more accurate.

Bus Consumption



Hypothesis Evaluation

- Group prefetching only has a impact on the bus efficiency of data prefetching
- Did not help instruction prefetching

Further doubt

- Simulator assumes the additional logic to send and read a sparse vector does not increase the cache hit latency
- Added cost and design complexity of design is unknown.
- Design is not more affective with smaller L1 caches (tested at half size).

Possible improvements?

- Use with a more accurate prefetcher
- Use with an out of order memory controller
- Larger and more data intensive benchmarks might take better advantage of the prefetching