Data Prefetch Mechanisms

STEVEN P. VANDERWIEL

IBM Server Group

AND

DAVID J. LILJA

University of Minnesota

The expanding gap between microprocessor and DRAM performance has necessitated the use of increasingly aggressive techniques designed to reduce or hide the latency of main memory access. Although large cache hierarchies have proven to be effective in reducing this latency for the most frequently used data, it is still not uncommon for many programs to spend more than half their run times stalled on memory requests. Data prefetching has been proposed as a technique for hiding the access latency of data referencing patterns that defeat caching strategies. Rather than waiting for a cache miss to initiate a memory fetch, data prefetching anticipates such misses and issues a fetch to the memory system in advance of the actual memory reference. To be effective, prefetching must be implemented in such a way that prefetches are timely, useful, and introduce little overhead. Secondary effects such as cache pollution and increased memory bandwidth requirements must also be taken into consideration. Despite these obstacles, prefetching has the potential to significantly improve overall program execution time by overlapping computation with memory accesses. Prefetching strategies are diverse, and no single strategy has yet been proposed that provides optimal performance. The following survey examines several alternative approaches, and discusses the design tradeoffs involved when implementing a data prefetch strategy.

Categories and Subject Descriptors: B.3.2 [**Memory Structures**]: Design Styles—*Cache memories*; B.3 [**Hardware**]: Memory Structures

General Terms: Design, Performance

Additional Key Words and Phrases: Memory latency, prefetching

This work was supported in part by National Science Foundation grants MIP-9610379, CDA-9502979, CDA-9414015, the Minnesota Supercomputing Institute, and the University of Minnesota-IBM Shared University Research Project. Steve VanderWiel was partially supported by an IBM Graduate Research Fellowship during the preparation of this work.

Authors' addresses: S. P. VanderWiel, System Architecture, Performance & Design, IBM Server Group, 3605 Highway 52, North Rochester, MN 55901; email: svw@us.ibm.com; D. J. Lilja, Dept. of Electrical & Computer Engineering, University of Minnesota, 200 Union St. SE, Minneapolis, MN 55455; email: lilja@ece.umn.edu.