Array Organization in Parallel Memories

Mayez Al-Mouhamed

The bandwidth mismatch between processor and main memory is one major throughput limiting problem. Although streamed computations have predictable access patterns their references have little temporal locality and are generally too long to cache. A memory and compiler co-optimization aimed at reducing low-level memory accesses using software and hardware locality optimizations is presented. We propose a scalable and predictable parallel memory based on a compiler synthesis of storage schemes for multi-dimensional arrays that are accessed by an arbitrary but known set of data access patterns. Using algebra of non-singular Boolean matrices, we present analysis of conflict-free access to (1) parallel memories, and (2) alignment networks. Finding a multi-pattern storage scheme is one NP-complete problem. An effective compiler heuristic is proposed for finding a storage matrix that minimizes overall memory access time. This applies to arbitrary linear patterns and arbitrary alignment networks. It is shown that the proposed storage scheme finds an optimal storage scheme for parallel (1) FFT, and (2) bitonic sorting. The proposed storage scheme outperforms statically optimized storages in the case of power-of-2 multi-stride access. The case of non power-of-2 strides is also addressed. The performance and scalability of the proposed parallel memory and its predictable access time are presented using numerical and multimedia algorithms. It is shown that a memory utilization above 83% is achieved by our storage scheme for 64 memories, which largely outperforms previous proposals. Our approach provides a tool for matching the storage pattern with the data access patterns needed for embedded systems running streamed computations with predictable data access patterns.

KEY WORDS: Access patterns; embedded systems; compiler optimization; parallel memory; streamed computations.

1 Computer Engineering Department, College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, P.O. Box 787, Diharan 31261, Saudi Arabia. E-mail: mayez@ccse.kfupm.edu.sa
1. INTRODUCTION

Effective utilization of bandwidth in hierarchical memory systems aims at exploiting compile-time knowledge of a program to reduce unnecessary data transfer between processor and main memory. A compiler optimization that attempts maximizing temporal and spatial localities and minimizing mapping conflicts produced encouraging results.\(^{(1)}\) To reduce memory conflicts in multiprocessors, compiler directed compaction-based data partitioning\(^{(2)}\) improved performance from 13% to 40% for a class of synchronous dataflow computations. The compiler\(^{(3)}\) knowledge of the access patterns of parallel applications is used to create compiler directed page coloring to direct run-time virtual memory page mapping. Here the compiler explicitly attempts predicting the access patterns of parallelized applications. A 50% improvement over a standard page mapping policy was achieved. Compile-time analysis of memory access can also improve scalar access in parallel memories where scheduling of a very low number of data transfers proved that a very high percentage of memory access conflicts can be avoided.

While media processing is becoming the dominating force in desktop computing a substantial fraction of processor resource is used to hide the latency of an unpredictable hierarchical memory system.\(^{(4)}\) The integration of high-speed logic and memory on a single chip provided large sequential and random memory bandwidth while making the delivered performance highly predictable. There is pressing need for innovative memory architectures and organization to reduce bandwidth imbalance between processor and main memory. The TERA MTA high-performance computer\(^{(5)}\) uses multi-threaded execution pipelines to tolerate a 100-clock latency between the logic and the memory. The bandwidth mismatch between the processor and the main memory is expected to be more acute in the future. It is predicted that processor-memory latency will be \(10^4–10^5\) clocks in the Petaflops HTMT Computer\(^{(6)}\) even with the use of exotic memory and optical network technologies.

For many synchronous dataflow multimedia computing a scalable and predictable parallel memory can be achieved if the data structure is allocated to separate memories whenever the compiler finds that data may be accessed in parallel, i.e., fine-grain data parallelism.

In high performance parallel memories, network contention and serialization of memory accesses are responsible for significant performance degradation.\(^{(7–10)}\) Memory interleaving causes non-uniform memory access, especially in the case of sequential addresses that differ by a constant amount, or stride, that is not relatively-prime to the number of memories. To reduce memory and network conflicts, q prime-number of memories\(^{(11)}\)