Lightweight Simulation of Programmable Memory Hierarchies*

Gergely Dévai

Eötvös Loránd University, Budapest
deva@elte.hu

Abstract

In most performance critical applications the bottleneck is data access. This problem is mitigated by applying memory hierarchies: Several layers of gradually smaller and faster memory layers are added to the large and slow main memory. There are application domains where the traditional, hard-wired cache control mechanisms are not satisfactory and the programmer is given full control where to place and when to move data. A known optimization technique on these kind of architectures is to do block-transfer of data instead of element-by-element access. Theoretical background of architectures providing such instructions has been established decades ago [1].

This contribution presents a lightweight library, written in and for C, to support experiments with algorithm performance on simulated programmable memory hierarchies. The library provides functions and macros to define memory layers, available block-transfer operations and data layout. The algorithm then can be run on a simple desktop computer and the library combines its real runtime with simulated memory access penalties. A similar approach, using C++ and mainly targetting cache memory simulation is described in [2].

Keywords: Memory hierarchy, block transfer, simulation

MSC: 68U20

References

[1] Alok Aggarwal, Ashok K Chandra, and Marc Snir. Hierarchical memory with block transfer. In *28th Annual Symposium on Foundations of Computer Science*, pages 204–216. IEEE, 1987.

[2] Theodoros Lioris, Grigoris Dimitroulakos, and Kostas Masselos. Xmsim: Extensible memory simulator for early memory hierarchy evaluation. In *VLSI 2010 Annual Symposium*, pages 199–216. Springer, 2011.

*Supported by EITKIC 12-1-2012-0001.