ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Journal of Systems Architecture 48 (2003) 429–452

JOURNAL OF SYSTEMS ARCHITECTURE

www.elsevier.com/locate/sysarc

Highly accurate and efficient evaluation of randomising set index functions

Hans Vandierendonck *,[1], Koen De Bosschere

Department of Electronics and Information Systems, Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium

Received 8 July 2002; received in revised form 7 November 2002; accepted 12 March 2003

Abstract

Randomising set index functions can reduce the number of conflict misses in data caches by spreading the cache blocks uniformly over all sets. Typically, the randomisation functions compute the exclusive ors of several address bits. Not all randomising set index functions perform equally well, which calls for the evaluation of many set index functions. This paper discusses and improves a technique that tackles this problem by predicting the miss rate incurred by a randomisation function, based on profiling information. A new way of looking at randomisation functions is used, namely the null space of the randomisation function. The members of the null space describe pairs of cache blocks that are mapped to the same set. This paper presents an analytical model of the error made by the technique and uses this to propose several optimisations to the technique. The technique is then applied to generate a conflict-free randomisation function for the SPEC benchmarks.

Keywords: Microprocessor data cache; Conflict misses; Randomisation; Randomising set index function; Null space

1. Introduction

Randomising set index functions can strongly influence data cache miss rates and improve performance predictability [1,2]. A randomising set index function spreads the referenced cache blocks in a more uniform way over all sets of the cache, reducing the number of conflicts. Instead of selecting a slice of bits from the block address to index the cache, a randomising set index function computes a more complex function of the address, e.g., using exclusive ors (XOR). Randomisation is most often used in direct mapped caches because these caches typically show more conflict misses than set-associative caches.

The idea behind randomisation is that conflict misses occur in a systematic way (e.g., when an array is accessed with a stride that is a large power of 2). By applying a randomisation function to the address before accessing the cache, the addresses are spread over the sets of the cache in a

* Corresponding author. Tel.: +32-9-264-34-05; fax: +32-9-264-35-94.

E-mail addresses: hvdieren@elis.rug.ac.be (H. Vandierendonck), kdb@elis.rug.ac.be (K. De Bosschere).

[1] Hans Vandierendonck is supported by the Flemish Institute for the Promotion of Scientific-Technological Research in the Industry (IWT).

1383-7621/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S1383-7621(03)00040-7