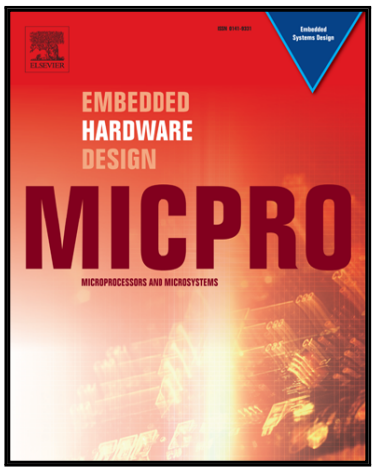

Memory Resources Aware Run-Time Automated Scheduling Policy for Multi-core Systems

Tassadaq Hussain

PII: S0141-9331(16)30403-3
DOI: 10.1016/j.micpro.2017.12.005
Reference: MICPRO 2644

To appear in: *Microprocessors and Microsystems*

Received date: 4 December 2016
Revised date: 1 December 2017
Accepted date: 7 December 2017

Please cite this article as: Tassadaq Hussain, Memory Resources Aware Run-Time Automated Scheduling Policy for Multi-core Systems, *Microprocessors and Microsystems* (2017), doi: 10.1016/j.micpro.2017.12.005

Memory Resources Aware Run-Time Automated Scheduling Policy for Multi-core Systems

Tassadaq Hussain
[1] Microsoft Research Center and Barcelona Supercomputing Center, Spain
UCERD: Unal Color of Education Research and Development, Islamabad
Riphah International University, Islamabad
tassadaq@ucerd.com

Abstract

Traditionally, the computer architect improves the system performance by integrating multiple types of processing cores and memory systems. However, there is relatively limited work done on investigating data transfers on the memory systems and scheduling the memory data transfers at the hardware level. Furthermore, the variable and unpredictable nature of the applications data transfers create unfair memory resource utilization that reduce the overall performance of a system. In this paper, a Memory Resource Aware Pattern-based Controller (MRAPC) is proposed and designed. MRAPC organizes the data transfers in pattern descriptors, prioritizes them with respect to the number and size of the transfer requests and manages the local and main memories. In order to measures the performance and effectiveness, the MRAPC is integrated into high performance ARM processing, FPGA based prototyping and Tasksim based Simulation enrichments. When compared to the baseline ARM and FPGA based multi-core systems, the FPGA and ARM based MRAPC systems achieve up to 2.15x and 1.91x performance respectively. While comparing the results of simulator environment, the MRAPC transfers data-structures up to 5.09x faster.

1. Introduction

To overcome the *memory wall* [1] effects, traditional DRAMs based architectures are adding wider and wider paths into memory and greater interleaving of memory banks. In recent years, a number of advanced DRAM devices have developed, which give the high performance by organizing and pipelining the data at