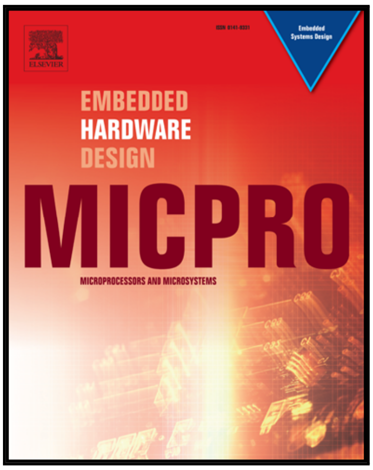

Scalable Embedded Computing through Reconfigurable Hardware: comparing DF-Threads, Cilk, OpenMPI and Jump.

Roberto Giorgi

PII: S0141-9331(18)30138-8
DOI: https://doi.org/10.1016/j.micpro.2018.08.005
Reference: MICPRO 2732

To appear in: *Microprocessors and Microsystems*

Received date: 13 March 2018
Revised date: 20 July 2018
Accepted date: 14 August 2018

Please cite this article as: Roberto Giorgi, Scalable Embedded Computing through Reconfigurable Hardware: comparing DF-Threads, Cilk, OpenMPI and Jump., *Microprocessors and Microsystems* (2018), doi: https://doi.org/10.1016/j.micpro.2018.08.005

Scalable Embedded Computing through Reconfigurable Hardware: comparing DF-Threads, Cilk, OpenMPI and Jump.

Roberto Giorgi

Dip. di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Italy. giorgi@dii.unisi.it

Abstract

Data-Flow Threads (DF-Threads) is a new execution model that permits to seamlessly distribute the workload across several cores (in a multi-core) and several nodes (in a multi-node/multi-board configuration).

In this paper, the advance in deploying this execution model is shown while developing it by using a combination of a simulator model (i.e., the COTSon framework) and a reconfigurable hardware platform (i.e., the AXIOM-board). The AXIOM platform consists of a custom board based on the Xilinx Zynq Ultrascale+ ZU9EG, which incorporates the largest FPGA available on that System-on-Chip at the moment, four 64-bit ARM cores and two 32-bit ARM cores, up to 32GiB of main memory and several 16Gbit/s transceivers.

While a complete DF-Threads system is still under development, but is already capable of running a full Linux OS and simple applications, so some initial results are presented here. In particular, well-known programming models that are used to exploit the Thread-Level Parallelism such as Cilk, OpenMPI and Jump are compared with DF-thread execution. Cilk is good for multi-cores, but it is not suitable for multi-nodes systems. In the latter cases, the distribution of the workload could be managed partly by the programmer when using programming models such as message-passing (OpenMPI has been chosen for reference) or distributed shared-memory (Jump in our case).

The obtained results show that a DF-Thread execution on a cluster of eight 4-core boards can provide a speed-up of more than 14x compared to the same configuration when using OpenMPI and more than 80x when compared with a OpenMPI single core, single node execution.

Keywords: Cyber-Physical Systems; Reconfigurable Systems; FPGA Programming; Distributed Shared Memory; Programming Model; Performance Evaluation; Memory Model.

1. Introduction

Embedded Computing embraces almost every application that is present in our lives and it includes a wide variety of hardware and software platforms, from very simple ones to very sophisticated ones, depending on the application [1].

While examining the ever increasing demand for computational power, in a small space and with limited energy,especially for highly-demanding Cyber-Physical Systems [1], an opportunity to start bringing high-performance computing concepts in the embedded domain aroused. In fact, the availability of a larger estate of reconfigurable logic (i.e., FPGA) such as in system such as the Intel Arria-10, or the Xilinx Zynq Ultrascale+ now permits the integration of several needed functionalities such as acceleration of specific functions and integration of specific IPs.

In the context of AXIOM project [2, 3, 4, 5, 6, 7, 8, 9], it was realized that there is currently an excessive fragmentation of both platforms and tools for embedded systems. In particular, when more sophisticated functionalities are needed, the whole system has to be redesigned and often a different toolchain has to be used. Therefore, our aim was to allow the users to easily program the platform with possibly a standard and open-source toolchain upon a full Linux-based software distribution.

Some very successful examples of simplicity as a key methodology for designing Cyber-Physical-System oriented applications include the well-known Arduino, that permits a rapid design [10, 11] both in hardware and in software. The simplic-