Available online at www.sciencedirect.com

ScienceDirect

Applied Soft Computing 8 (2008) 383–391

Applied Soft Computing

www.elsevier.com/locate/asoc

A hardware/software partitioning algorithm based on artificial immune principles[☆]

Yiguo Zhang [a,b], Wenjian Luo [a,b,*], Zeming Zhang [a,b], Bin Li [b], Xufa Wang [a,b]

[a] Nature Inspired Computation and Applications Laboratory, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China

[b] Anhui Key Laboratory of Software in Computing and Communication, University of Science and Technology of China, Hefei 230027, China

Received 4 July 2006; received in revised form 2 February 2007; accepted 4 March 2007
Available online 7 March 2007

Abstract

Hardware/software codesign is the main approach to designing the embedded systems. One of the primary steps of the hardware/software codesign is the hardware/software partitioning. A good partitioning scheme is a tradeoff of some constraints, such as power, size, performance, and so on. Inspired by both negative selection model and evolutionary mechanism of the biological immune system, an evolutionary negative selection algorithm for hardware/software partitioning, namely ENSA-HSP, is proposed in this paper. This ENSA-HSP algorithm is proved to be convergent, and its ability to escape from the local optimum is also analyzed. The experimental results demonstrate that ENSA-HSP is more efficient than traditional evolutionary algorithm.

Keywords: Artificial immune system; Hardware/software partitioning; Evolutionary algorithm; Negative selection

1. Introduction

With the development of the design complexity of embedded systems, its design procedure has been revolutionized. The concurrent design of hardware and software has displaced traditional sequential design. Furthermore, hardware and software design begins before the system architecture is finalized [1]. The codesign approach allows testing the software and hardware concurrently at the early design period. Therefore, the problems on the design could be solved earlier and easier, and the design periods could be shortened. The codesign contains some primary issues, such as the task partitioning, schedule, synthesis, etc. [1,2]. Hardware/software partitioning is one of the key issues. The purpose of hardware/software partitioning is to determine that each task will be implemented by software or hardware [2]. Implementation with software module has more flexibility and need less cost, but more executing time, and vice versa in hardware case. The main target of hardware/software partitioning is to balance all the parameters to optimize some objects of the system under some constraints [2]. Heuristic algorithms including evolutionary algorithms are the main methods for hardware/software partitioning [2,9–11,14,17,18].

Artificial immune system (AIS) is a novel bio-inspired computation approach [3,4]. The basic function of immune system is to discriminate nonself from self, and then eliminate the nonself individuals. Inspired by biological immune system, many models and algorithms are designed in AIS community to solve different problems [3,4,21,22]. Evolutionary negative selection algorithms (ENSAs) are designed by combining the negative selection mechanism [5] with the evolutionary mechanism [6] of biological immune system. Currently, the main application fields of ENSA are anomaly detection [7], combinational logic circuit design [19,22] and simple function optimization [21,22].

An evolutionary negative selection algorithm for hardware/software partitioning of codesign is proposed in this paper, namely ENSA-HSP, as described in Ref. [22]. In every cycle of

[☆] This paper is based on Yiguo Zhang's Master thesis (in Chinese) in Department of Computer Science and Technology at University of Science and Technology of China in 2006.

[*] Corresponding author at: Nature Inspired Computation and Applications Laboratory, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China. Tel.: +86 551 3602824.

E-mail addresses: ygzhang@mail.ustc.edu.cn (Y. Zhang), wjluo@ustc.edu.cn (W. Luo), zmzhang@mail.ustc.edu.cn (Z. Zhang), binli@ustc.edu.cn (B. Li), xfwang@ustc.edu.cn (X. Wang).

1568-4946/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2007.03.003

the evolutionary iteration, the worst individuals are added into the self-set, and the self-set is updated by the first-in-first-out (FIFO) strategy. Based on such a self-set, all of the new immature individuals are filtered by negative selection. Compared with traditional evolutionary algorithm (EA), ENSA-HSP has better experimental results.

The rest of this paper is organized as follows. Section 2 describes the problem of hardware/software partitioning and introduces the related concepts briefly. Section 3 illustrates the main procedure of ENSA-HSP. Section 4 analyzes the ENSA-HSP theoretically. Section 5 gives some experimental results. The last section summarizes this paper with a brief conclusion.

2. Problem description

2.1. Function description

The flow chart of hardware/software partitioning is shown in Fig. 1 [18]. The system is usually described in high-level language. The input is formulated to a series of basic schedule blocks (BSB). And then all the blocks are formed into a control-data flow graph (CDFG), which is composed of nodes and arcs [10,11,14,18]. The nodes denote the BSBs, and the arcs denote the dataflows between BSBs. Generally, the CDFG is a directed acyclic graph (DAG). The partitioning algorithm determines which nodes in the DAG are implemented by hardware and which by software. After partitioning, all the BSBs are scheduled and the solution is evaluated. The partitioning algorithm will terminate when the termination conditions are satisfied, i.e., the solution achieves a good tradeoff of some constraints, such as power, size, performance, and so on. Otherwise, generate a new partition and evaluate again.

Every BSB (i.e. node in CDFG) can receive data from its previous nodes, and send data to its next nodes. The BSB is described by a seven-dimension vector $N_i = \langle as_i, ts_i, ah_i, th_i, r_i, w_i, pc_i \rangle$, where as_i and ts_i, respectively, denote the cost and the executing time implemented by software, ah_i and th_i denote these implemented by hardware. r_i and w_i are two arrays, which, respectively, denote the incoming and outcoming node sets, and, respectively, store the corresponding communication time sets. pc_i denotes that the node i executes pc_i times repeatedly [17].

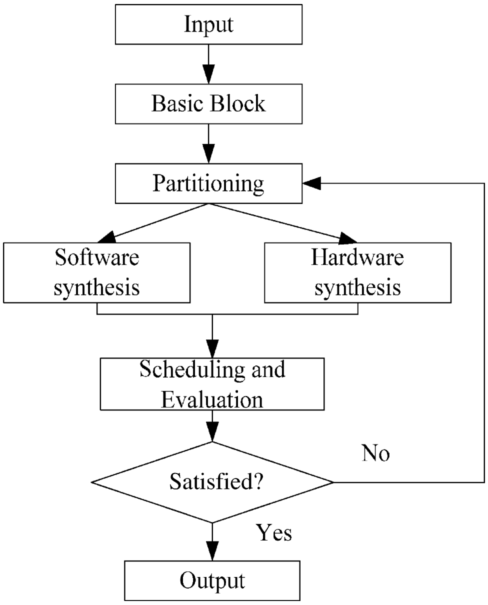

Fig. 1. Flow chart of hardware/software partitioning.

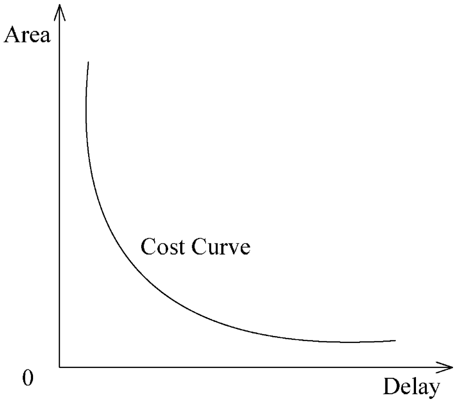

Fig. 2. Cost–delay relationship.

When the system function is defined, an increment of hardware nodes will reduce the delay, but increase the cost, as shown in Fig. 2 [8]. According to different constraints, there are two basic problems: optimize the time in terms of cost [9] and optimize the cost in terms of time [10]. In this paper, ENSA-HSP is used to optimize the cost under the time constraint.

2.2. Partitioning model

The partition model adopted in this paper is shown in Fig. 3 [18]. "HA1", "HA2", . . ., "HAm" represent the hardware nodes implemented by ASIC (application specific integrated circuit) or FPGA (field programmable gate array). The software nodes are executed on a programmable processor (denoted by CPU). All the nodes exchange data through a shared bus, and they share the common memory to store the interim data. All the hardware nodes without data dependency can execute concurrently.

The partitioning problem in this paper can be described as follows [10]:

$$\text{Minimize C}, \quad \text{Subject to T} <= \mathit{TimeReq},$$

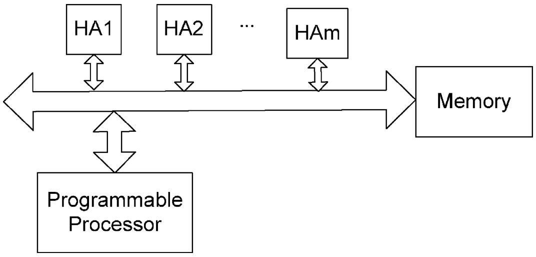

Fig. 3. The partition model used in this paper.