Yan XH, He FZ, Chen YL. A novel hardware/software partitioning method based on position disturbed particle swarm optimization with invasive weed optimization. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(2): 340–355 Mar. 2017. DOI 10.1007/s11390-017-1714-2

A Novel Hardware/Software Partitioning Method Based on Position Disturbed Particle Swarm Optimization with Invasive Weed Optimization

Xiao-Hu Yan, *Member, CCF*, Fa-Zhi He*, *Member, CCF*, and Yi-Lin Chen

State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China
School of Computer Science, Wuhan University, Wuhan 430072, China

E-mail: yanxiaohupaul@126.com; fzhe@whu.edu.cn; 285813238@qq.com

Received July 27, 2016; revised December 24, 2016.

Abstract With the development of the design complexity in embedded systems, hardware/software (HW/SW) partitioning becomes a challenging optimization problem in HW/SW co-design. A novel HW/SW partitioning method based on position disturbed particle swarm optimization with invasive weed optimization (PDPSO-IWO) is presented in this paper. It is found by biologists that the ground squirrels produce alarm calls which warn their peers to move away when there is potential predatory threat. Here, we present PDPSO algorithm, in each iteration of which the squirrel behavior of escaping from the global worst particle can be simulated to increase population diversity and avoid local optimum. We also present new initialization and reproduction strategies to improve IWO algorithm for searching a better position, with which the global best position can be updated. Then the search accuracy and the solution quality can be enhanced. PDPSO and improved IWO are synthesized into one single PDPSO-IWO algorithm, which can keep both searching diversification and searching intensification. Furthermore, a hybrid NodeRank (HNodeRank) algorithm is proposed to initialize the population of PDPSO-IWO, and the solution quality can be enhanced further. Since the HW/SW communication cost computing is the most time-consuming process for HW/SW partitioning algorithm, we adopt the GPU parallel technique to accelerate the computing. In this way, the runtime of PDPSO-IWO for large-scale HW/SW partitioning problem can be reduced efficiently. Finally, multiple experiments on benchmarks from state-of-the-art publications and large-scale HW/SW partitioning demonstrate that the proposed algorithm can achieve higher performance than other algorithms.

Keywords hardware/software partitioning, particle swarm optimization, invasive weed optimization, communication cost, parallel computing

1 Introduction

Hardware/software (HW/SW) partitioning which involves multidisciplinary and collaborative design is a key step in the design of modern embedded products[1]. Implementation with software module has more flexibility and needs less cost, but costs more executing time, and vice versa in hardware case[2]. The target of HW/SW partitioning is to balance all the tasks to optimize some objectives of the system under some constraints[3]. Traditionally, HW/SW partitioning is carried out manually. However, with the development of the design complexity in embedded systems, HW/SW partitioning becomes a challenging problem.

Many approaches about HW/SW partitioning have been proposed. Based on the partitioning algorithm, exact and heuristic solutions can be differentiated[4]. The proposed exact algorithms include dynamic programming[5-6], Branch-and-Bound (B&B)[7-9] and Integer Linear Programming (ILP)[10]. Exact algorithms cannot provide a feasible solution for the large-scale HW/SW partitioning problem because most formulations of the HW/SW partitioning problem are NP-hard. Thus, many researchers have applied heuris-

Regular Paper

The work was supported by the National Natural Science Foundation of China under Grant No. 61472289 and the National Key Research and Development Project of China under Grant No. 2016YFC0106305.

*Corresponding Author

tic algorithms to HW/SW partitioning. The traditional heuristic algorithms include hardware-oriented and software-oriented ones. The hardware-oriented approach starts with a complete hardware solution and swaps parts to software until constraints are violated[11-12], while the software-oriented approach means that the initial implementation of the system is supposed to be a software solution. General-purpose heuristics include genetic algorithm (GA)[13-16], simulated annealing (SA)[17-18], greedy algorithm[19], tabu search (TS)[20-21], ant colony optimization (ACO)[22-23] and particle swarm optimization (PSO)[24-25]. The other group of partitioning-related heuristics is the Kernighan-Lin heuristic[26], which is substantially improved by many others[27-29]. All these approaches can work perfectly within their own co-design environments, but due to the enormous differences among them, it is not possible to compare the results obtained[30].

In this paper, a novel HW/SW partitioning method based on position disturbed particle swarm optimization with invasive weed optimization (PDPSO-IWO) is presented. To fight against premature convergence and avoid local optimum, the particles in PDPSO-IWO move away from the worst particle in the population, near which there is a potential predatory threat. To improve the search accuracy and the solution quality, the improved IWO is integrated to search a better position, with which the global best position is updated. To enhance the solution quality, a hybrid NodeRank (HNodeRank) algorithm is proposed to initialize the population. To accelerate HW/SW partitioning method based on PDPSO-IWO, the HW/SW communication cost runs in GPU parallel computing environment.

The rest of the article is organized as follows. Section 2 describes related work. Position disturbed particle swarm optimization with invasive weed optimization is presented in Section 3. A novel HW/SW partitioning method based on PDPSO-IWO is presented in Section 4. Section 5 analyzes the performance of PDPSO-IWO experimentally. Finally, conclusions are drawn in Section 6.

2 Related Work

2.1 Problem Definition

In this paper, the partitioning problem definition is based on the following notations. Given an undirected graph $G = (V, E)$, where V is the vertex set and E is the edge set, the HW/SW partitioning problem can be formulated to the minimization problem P as follows:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n} h_i(1 - x_i) \\ &\text{subject to} && \sum_{i=1}^{n} s_i x_i + C(\boldsymbol{x}) \leqslant R, \\ &&& x_i \in \{0, 1\}, i = 1, 2, ..., n, \end{aligned} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ indicates a solution of HW/SW partitioning problem. $x_i = 1(0)$ denotes that the node is partitioned to software(hardware). s_i and h_i indicate the software and the hardware cost of node v_i, respectively. $C(\boldsymbol{x})$ indicates the communication cost of $\boldsymbol{x}$, and R is the constraint. In [19], $C(\boldsymbol{x})$ is replaced with uR, where $0 \leqslant u \leqslant 1$, and then the problem P can be converted to Q':

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{n} h_i x_i \\ &\text{subject to} && \sum_{i=1}^{n} s_i x_i \leqslant (1 - u)R, \\ &&& x_i \in \{0, 1\}, i = 1, 2, ..., n. \end{aligned}$$

The HW/SW partitioning problem Q' is reduced to a variation of knapsack problem in [19]. To obtain the global best solution, the PDPSO-IWO is used to solve the problem P in (1).

2.2 Heuristic Algorithms

As a key challenge in HW/SW co-design, HW/SW partitioning has been studied for many years. In early studies, the HW/SW partitioning models and algorithms have been tested on systems with some dozens of components. The target architecture is supposed to consist of a single software and a single hardware unit[31-32]. The HW/SW partitioning problem has specific optimization objectives, such as minimizing power, hardware area and communication overhead[33-35]. But in recent research, the HW/SW partitioning model and algorithm have been tested on systems with hundreds or even thousands of components. The HW/SW partitioning problem is formalized as a task graph, or a set of task graphs. In this work, the HW/SW partitioning problem is based on the same assumptions and system model which are used in [4, 19]. This paper does not aim at partitioning for a given architecture, nor does it propose a complete co-design environment. The HW/SW partitioning definition is general enough so that the proposed algorithm can be used in different practical cases[4].

Many exact algorithms for HW/SW partitioning were proposed in 1990s, such as B&B and ILP. Exact algorithms have mathematic solution and can find high-quality solutions rapidly for the small-scale HW/SW