Journal of Systems Architecture 70 (2016) 3–14

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

High performance real-time scheduling of multiple mixed-criticality functions in heterogeneous distributed embedded systems

Guoqi Xie [a,b,*], Gang Zeng [c], Liangjiao Liu [a,b], Renfa Li [a,b], Keqin Li [a,d]

[a] College of Computer Science and Electronic Engineering, Hunan University, China
[b] Key Laboratory for Embedded and Network Computing of Hunan Province, China
[c] Graduate School of Engineering, Nagoya University, Japan
[d] Department of Computer Science, State University of New York, New Paltz, New York, USA

ARTICLE INFO

Article history:
Received 8 September 2015
Revised 7 December 2015
Accepted 14 April 2016
Available online 28 April 2016

Keywords:
Criticality level
Deadline missed ratio (DMR)
Deadline-span
High performance
Real-time

ABSTRACT

The architectures of high-end embedded system have evolved into heterogeneous distributed integrated architectures. The scheduling of multiple distributed mixed-criticality functions in heterogeneous distributed embedded systems is a considerable challenge because of the different requirements of systems and functions. Overall scheduling length (i.e., makespan) is the main concern in system performance, whereas deadlines represent the major timing constraints of functions. Most algorithms use the fairness policies to reduce the makespan in heterogeneous distributed systems. However, these fairness policies cannot meet the deadlines of most functions. Each function has different criticality levels (e.g., severity), and missing the deadlines of certain high-criticality functions may cause fatal injuries to people under this situation. This study first constructs related models for heterogeneous distributed embedded systems. Thereafter, the criticality certification, scheduling framework, and fairness of multiple heterogeneous earliest finish time (F_MHEFT) algorithm for heterogeneous distributed embedded systems are presented. Finally, this study proposes a novel algorithm called the deadline-span of multiple heterogeneous earliest finish time (D_MHEFT), which is a scheduling algorithm for multiple mixed-criticality functions. The F_MHEFT algorithm aims at improving the performance of systems, while the D_MHEFT algorithm tries to meet the deadlines of more high-criticality functions by sacrificing a certain performance. The experimental results demonstrate that the D_MHEFT algorithm can significantly reduce the deadline miss ratio (DMR) and keep satisfactory performance over existing methods.

1. Introduction

1.1. Background

High-end embedded system architectures have evolved into heterogeneous distributed architectures because of the size, weight, and power consumption (SWaP) for cost and high performance benefits. For example, automotive electronic architectures consist of many heterogeneous electronic control units (ECUs) that are distributed on multiple network buses, which are interconnected by a central gateway. Today, a luxury car comprises at least 70 heterogeneous ECUs with approximately 2500 signals [1]. The number of ECUs is expected to increase further in future automotive electronic systems.

The aforementioned distributed architecture leads to an increase in distributed functions (also called functionalities or applications in a few studies) with precedence-constrained tasks in automotive electronic systems [2]. Examples of active safety functions are x-by-wires and adaptive cruise control [3]. The integration of multiple functions in the same architecture is called "integrated architecture," in which multiple functions can be supported by one ECU and one function can be distributed over multiple ECUs [3]. Integrated architectures are indeed an essential evolution to cope with the SWaP problems and seize the opportunity for cost reduction. This transition requires the development of new models and methods [3].

Integrated architecture drives the integration of several levels of safety-criticality and non-safety-criticality functions into the same platform; criticality levels and mixed-criticality systems have also been introduced [4]. Criticality level is represented by the automotive safety integrity level (ASIL) in the automotive functional safety standard ISO 26262 [5]. ASIL refers to a classification of inherent safety goals required by the standard to ensure the accomplish-

---

* Corresponding author.
E-mail addresses: xgqman@gmail.com, xgqman@hnu.edu.cn (G. Xie), sogo@ertl.jp (G. Zeng), llj1984109@qq.com (L. Liu), lirenfa@hnu.edu.cn (R. Li), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.sysarc.2016.04.008

ment of goals in the system; ASIL D and ASIL A represent the highest and lowest criticality levels, respectively [5]. Mixed-criticality systems are new systems that attempt to combine multiple functions with different criticality levels on the same platform.

1.2. Motivations

To make full use of the numerous ECUs in automobiles, efficient scheduling policies are required to achieve substantially high performance improvement. However, scheduling multiple distributed mixed-criticality functions in heterogeneous distributed embedded systems involves the following challenges.

First, many scheduling methods for mixed-criticality systems have been developed in the past years, but such methods are mainly based on periodic and sporadic task models. Many distributed functions have apparent precedence constraints among tasks in high-end heterogeneous distributed embedded systems (e.g., automotive electronic systems). Evidence shows that models for mapping distributed functions are highly criticality to the analysis of automotive electronic systems. A few models, such as time chains [6] and task chains [7], have been employed in automobiles; however, these models are only suitable for simple distributed functions. With the increasing complexity and parallelization of automobile functions, a model that accurately reflects the distributed characteristics of automotive functions is desirable. In heterogeneous distributed systems, a distributed function with precedence-constrained tasks at a high level is described as a directed acyclic graph (DAG), in which the nodes represent the tasks and the edges represent the communication messages between the tasks [1,8]. The DAG-based model has also been applied to automotive electronic systems [9,10].

Second, systems and functions in heterogeneous distributed embedded systems involve considerable conflicts. Overall scheduling length (makespan) is the main concern in system performance, whereas deadlines are the major timing constraints of functions. The deadlines of all functions cannot be met in heterogeneous distributed embedded systems, particularly in resource-constrained distributed embedded environments. A high-criticality function (i.e., a function with high criticality level) has a considerably important and strict timing constraint for a given deadline. Missing the deadlines of high-criticality functions results in fatal injuries to people. Most algorithms use fairness policies to reduce the overall makespan of systems in heterogeneous distributed systems; however, these policies could lead to the failure to meet the deadlines of high-criticality functions. Therefore, both performance and timing constraints should be considered to achieve a good makespan and low deadline miss ratio (DMR) [11].

1.3. Our contributions

Our contributions are summarized as follows. First, we construct a series of models for heterogeneous distributed embedded systems from the “distributed computing” and “functional safety” perspectives. Second, we propose a functional level scheduling algorithm with a round-robin fairness policy from the “system performance” perspective. Third, we further propose a functional level scheduling algorithm with a deadline-span-driven policy to achieve satisfactory system performance and low DMR.

The rest of this paper is organized as follows. Section 2 reviews the related literature. Section 3 constructs a series of models for heterogeneous distributed embedded systems. Section 4 proposes the certification method, scheduling framework, and round-robin fairness scheduling. Section 5 proposes a scheduling algorithm with a deadline-span-driven policy. Section 6 verifies the performance ratios of all the proposed methods of this study. Section 7 concludes this study.

2. Related works

High performance is an important concern of heterogeneous distributed systems, whereas timing constraints represent an important requirement of high-criticality functions. This section first reviews the related research for high performance scheduling and then discusses real-time scheduling.

2.1. High performance scheduling

The scheduling of a single distributed function (also called single DAG-based function scheduling) is the basis of the scheduling of multiple distributed functions (also called multiple DAG-based function scheduling). Thus, we briefly introduce the single DAG-based function list scheduling. List scheduling includes two phases: the first phase orders tasks according to the descending order of priorities (task prioritizing), whereas the second phase allocates each task to a proper processor (task allocation). Scheduling tasks for a single DAG-based function with the fastest execution is a well-known NP-hard optimization problem [8]. In [8], Topcuoglu et al. proposed the popular algorithm called the heterogeneous earliest finish time (HEFT) for the single DAG-based function scheduling in heterogeneous distributed systems to reduce makespan to a minimum. The HEFT algorithm uses upward rank values for task ordering and the earliest finish time (EFT) based on the insertion-based policy for task allocation. The aforementioned study further inspired substantial investigations and the development of other algorithms, including constrained EFT (CEFT) [12], predict EFT (PEFT) [13], and heterogeneous selection value (HSV) [1].

The multiple DAG-based functions scheduling of heterogeneous systems also involves two steps, namely, task prioritizing and task allocation. In [14], Honig et al. first proposed a composition approach to merge multiple distributed functions into one new function and then used a single DAG-based function scheduling algorithm (e.g., HEFT) to schedule the new DAG-based function. However, apparent unfairness to functions with short makespans emerges because the upward rank values of these functions are significantly lower than those of functions with long makespans. This approach limits the execution opportunities of functions with short makespans, and such limitation results in an unfairness to them and in a considerably long overall makespan in systems. In [15], Zhao et al. first identified the fairness issue in the scheduling of multiple DAG-based functions. The authors proposed a fairness scheduling algorithm called Fairness with a slowdown-driven policy that ensures the fairness of different functions. Other related studies, such as those on online workflow management (OWM) [16] for overall makespan minimization and fairness dynamic workflow scheduling (FDWS) [17] for minimization of individual functions were conducted.

2.2. Real-time scheduling

The mixed-criticality scheduling problem was first identified and formalized by Vestal [18]. whose work has been extended and has inspired further substantial investigations [19–22]. However, the models of these works are only periodic [19,20] and sporadic tasks models [21,22]. Hence, these works only considered mixed-criticality from the “task level” perspective and cannot reflect the distributed characteristics of functions in automobiles. For the functional safety of automobiles, scheduling should be considered at the “functional level” and not at the “task level.”

Some related researches are concerned about function scheduling with deadline constraints [23–25]. However, these solutions are merely for single DAG-based scheduling, and not suitable for multiple DAG-based scheduling issues.