4-2017

AutoV: An Automotive Testbed for Real-Time Virtualization

Meng Xu
University of Pennsylvania, mengxu@cis.upenn.edu

Insup Lee
University of Pennsylvania, lee@cis.upenn.edu

Recommended Citation

Meng Xu and Insup Lee, "AutoV: An Automotive Testbed for Real-Time Virtualization", *Second TCRTS Workshop on Certifiable Multicore Avionics and Automotive Systems (CMAAS 2017)* . April 2017.

Second TCRTS Workshop on Certifiable Multicore Avionics and Automotive Systems (CMAAS 2017), Pittsburgh, PA, Apr. 2017.

AutoV: An Automotive Testbed for Real-Time Virtualization

Abstract
Timing isolation is critical for automotive systems. Real-time virtualization, such as RT-Xen, is a promising technique to integrate legacy automotive systems onto a powerful multi-core platform for achieving better performance and lower cost without breaking the timing isolation. However, the real-time virtualization has never been evaluated with real automotive applications in a non-simulation environment. In order to facilitate the evaluation of real-time virtualization for automotive systems, we propose the AutoV, an affordable and accessible automotive testbed for real-time virtualization. We present a case study to demonstrate the applications of the AutoV.

Disciplines
Computer Engineering | Computer Sciences

Comments
Second TCRTS Workshop on Certifiable Multicore Avionics and Automotive Systems (CMAAS 2017), Pittsburgh, PA, Apr. 2017.

AutoV: An Automotive Testbed for Real-Time Virtualization

Meng Xu Insup Lee
PRECISE Center
University of Pennsylvania

I. Introduction

Automotive systems are becoming increasingly complex. Virtualization is a promising technique to achieve low size, weight, power and cost (SWaP-C) for automotive systems: functionalities on multiple ECUs can be consolidated into multiple virtual machines (VMs) on a commodity multicore processor. However, the temporal coupling among concurrently running applications in VMs on a multicore processor makes the certification of virtualized automotive systems particularly challenging. The functional safety of automotive applications may be violated by the interference channels on multicore processors, such as CPU, cache, shared memory, and I/O devices. In order to leverage the benefits of virtualization in automotive domain, we must identify, analyze, and mitigate these sources of interference for virtualization systems.

Real-time virtualization, such as the RT-Xen project [2], is a promising technique to achieve the certifiable virtualization platform for automotive systems. The goal of the RT-Xen project is to mitigate and analyze all sources of interference on multicore processors. We, together with our collaborators, have eliminated the CPU interference by developing a set of real-time schedulers in the RT-Xen 2.0 [2]. We have also analyzed the private cache interference by proposing the cache-aware compositional analysis [4]. Recently, we have mitigated the shared cache interference by designing the vCAT, a dynamic cache management framework using Intel Cache Allocation Technology virtualization [3]. We are currently working on mitigating the interference introduced by the shared memory. As a future work, we will investigate how to mitigate the interference introduced by the shared I/O devices. With all interference channels of multicore processors being mitigated and analyzed, we believe that the RT-Xen can be used towards the certified virtualization platform for automotive systems.

The RT-Xen has never been evaluated with real automotive applications in a non-simulation environment. Evaluating existing or new real-time virtualization ideas on automotive systems is challenging due to two reasons: (i) the industrial automotive applications are usually private and not accessible to most researchers; and (ii) building a virtualized automotive systems on a real car is too costly. In order to facilitate the evaluation of real-time virtualization for automotive systems, we propose the AutoV, an affordable and accessible automotive testbed for real-time virtualization.

We show the applications of the AutoV with a case study: (1) we demonstrate that virtualization can be used to integrate different automotive functionalities in VMs. We also discuss the experience of enabling virtualization for the automotive system; (2) we show that timing predictability is important for the safety of automotive systems in virtualization environment; (3) we demonstrate the capabilities of RT-Xen for providing the temporal isolation to the automotive system; (4) we also identify the remaining interference channels the RT-Xen should mitigate in the future work.

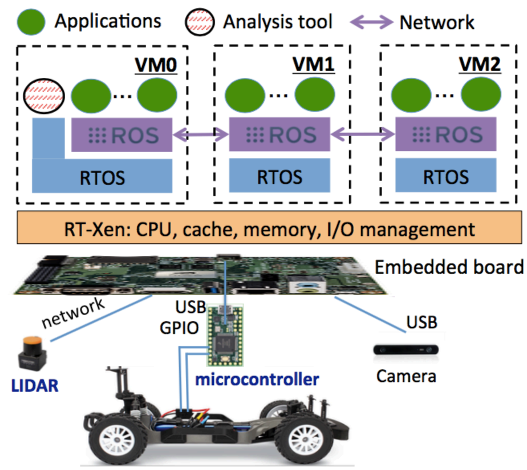

Fig. 1: Overview of AutoV architecture.

II. AutoV framework

The AutoV integrates (1) an autonomous car to provide the automotive applications for evaluation. The car can autonomously drive in indoor hallways from point A to point B by following a pre-provided map; (2) a RT-Xen hypervisor to consolidate automotive functionalities into VMs; (3) a compositional analysis tool, e.g., CARTS tool [1], to analyze the resource requirement of VMs in a compositional way. Fig. 1 illustrates the architecture of the AutoV.

III. Case study

In the case study, the car follows the same map under four different scenarios: (i) baseline scenario when the autonomous applications run alone in non-virtualization environment; (ii) virtualization-baseline scenario when the autonomous applications run alone in VMs on Xen; (iii) virtualization-unsafe scenario when the interference applications and autonomous applications run in different VMs on Xen and interfere with each other; (iv) real-time virtualization scenario which is similar to the virtualization-unsafe scenario but replaces Xen with RT-Xen. We compare the real-time performance of the autonomous applications in these four different scenarios.

References

[1] L. T. X. Phan, J. Lee, A. Easwaran, V. Ramaswamy, S. Chen, I. Lee, and O. Sokolsky. Carts: A tool for compositional analysis of real-time systems. *SIGBED Rev.*, 8(1):62–63, Mar. 2011.

[2] S. Xi, M. Xu, C. Lu, L. T. X. Phan, C. Gill, O. Sokolsky, and I. Lee. Real-time multi-core virtual machine scheduling in xen. In *EMSOFT*, 2014.

[3] M. Xu, L. T. X. Phan, H.-Y. Choi, and I. Lee. vcat: Dynamic cache management using cat virtualization. In *RTAS*, 2017.

[4] M. Xu, L. T. X. Phan, I. Lee, O. Sokolsky, S. Xi, C. Lu, and C. Gill. Cache-aware compositional analysis of real-time multicore virtualization platforms. In *RTSS*, 2013.