A Multicore Real-Time Mixed-Criticality Framework for Avionics

Department of Computer Science University of North Carolina at Chapel Hill August 2014

The Challenge

Embedded avionics software systems have stringent certification requirements that typically entail the validation of temporal correctness in addition to logical correctness. Informally,*logical correctness* means that tasks (i.e., programs) produce correct outputs, while *temporal correctness* means that such outputs are produced at the correct time (e.g., by specified deadlines). To validate temporal correctness, some knowledge of task execution times is required. Ideally, *provable* upper bounds on execution times would be used. Unfortunately, obtaining such a bound usually requires pessimistic assumptions regarding task behaviors and hardware functionality. This can cause predicted execution times to greatly exceed those that actually occur. The resulting over-provisioning can be detrimental from a size, weight, and power (SWaP) viewpoint.

A few years ago, Steve Vestal (while working in the avionics industry at Honeywell) observed that the extent of over-provisioning can be eased by tailoring execution-time assumptions to the criticality of the software component being analyzed. Specifically, he noted that *real* systems are usually comprised of tasks of differing criticalities; further, different execution-time estimation methods are often used for different criticality levels: for highly critical components, pessimistic tool-produced upper bounds may be required, but for less critical components, empirically estimated times might be reasonable. Vestal proposed reflecting such assumptions in validation: when validating a system at criticality level L, execution times should be assumed (for tasks at *any* level) commensurate with estimation methodologies appropriate for level L. The formal task model resulting from Vestal's work has come to be known as the **mixed-criticality task model**.

Vestal's observations led to a flurry of research within the real-time systems community on mixed-criticality task systems. However, this research has been almost entirely theoretical in nature and has little practical impact (which is disconcerting, given that the proposed mixed-criticality task model was suggested by someone actually working in industry). One serious limitation of prior work is that it has been mostly limited to uniprocessor platforms. Given the advent of multicore technologies, continued reliance on uniprocessor platforms in avionics systems will impede future functional advances. Prior work has also emphasized theoretical issues such as approximation ratios (which enable comparison to "ideal" allocations) over and above practical validation issues.

The Approach

In this project, we are seeking to return work on mixed-criticality real-time systems to its roots by conducting research that emphasizes algorithms and techniques that can be practically applied. We also seek to expand the focus of mixed-criticality resource allocation by directly addressing issues of relevance to multicore platforms. Our research agenda includes the development of a multicore-based mixed-criticality resource allocation framework that includes analysis for checking timing constraints, and an experimental evaluation of this framework that focuses on workloads pertaining to future unmanned air vehicles (UAVs). In comparison to current UAVs, these future UAVs will have far greater autonomous capabilities and will be significantly better equipped to adapt to changing environmental conditions. They will also have intensive computational workloads (hence the need for multicore), have system components of varying criticalities, and be subject to stringent certification requirements.

The proposed mixed-criticality framework is being developed by extending prior work by the investigators and colleagues at Northrop Grumman Corp. (NGC) that resulted in the development of a basic multicore-based mixed-criticality scheduling framework called **MC^2** (**M**ixed-**C**riticality on **M**ulti**C**ore). In this project, this basic framework is being extended in several significant ways. For example, to enable dynamic workload changes to be supported in MC^2, new techniques are being developed for changing a task's time-related parameters at runtime and for analyzing the effects of such changes. Research on such techniques and other issues will proceed by following a research agenda that includes work on real-time scheduling and synchronization mechanisms of fundamental relevance to the proposed framework, work on analysis methods for certifying timing constraints, and evaluations of the resulting framework based on prototype implementations. In all of this work, interactions with colleagues at NGC will continue so that the obtained framework has real industry relevance.

Significance

Over the past decade, the U.S. Air Force and other service branches of the U.S. Armed Forces have recognized the need to efficiently utilize multicore computers aboard deployed systems. Unfortunately, the pessimism noted above regarding the validation of timing constraints has been a key stumbling block in this regard. Vestal’s proposed mixed-criticality analysis methods are a possible way forward. However, practical multicore-ready mixed-criticality frameworks must be devised for this way forward to become a reality. Such a framework will be developed and implemented in this project.

Project Members

James Anderson, professor
Sanjoy Baruah, professor

Research Sponsor

U.S. Air Force Office of Scientific Research

For More Information

Dr. James Anderson
Department of Computer Science
University of North Carolina at Chapel Hill
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175
Phone: (919) 590-6057
Fax: (919) 590-6105
Email: anderson@cs.unc.edu

http://www.cs.unc.edu/~anderson/projects/mcavionics.html