



ELSEVIER

Journal of Systems Architecture 46 (2000) 49–77

---

---

JOURNAL OF  
SYSTEMS  
ARCHITECTURE

---

---

www.elsevier.com/locate/sysarc

# Scheduling optional computations for adaptive real-time systems

Charlie McElhone \*, Alan Burns

*Real-Time Systems Research Group, Department of Computer Science, University of York, xxxx, UK*

Received 23 July 1997; received in revised form 19 March 1998; accepted 11 September 1998

---

## Abstract

At present, the critical computations of real-time systems are guaranteed before run-time by performing a worst-case analysis of the system's timing and resource requirements. The result is that real-time systems are engineered to have *spare capacity*, under normal operation. A challenge of current research is to make use of this spare capacity, in order to satisfy requirements for *adaptivity* in the system. Adaptivity can be implemented by *optional computations* with *firm* deadlines, which can be guaranteed at run-time by the use of *flexible scheduling*. This report assumes that the algorithms which attempt to guarantee optional computations at run-time, actually run on the same processor as the optional and critical computations themselves. The report starts with a brief survey of the complex requirements for adaptivity within real-time systems. Such requirements can include task hierarchies composed of interdependent subtasks each with its own utility. Evidence is cited which indicates that the run-time support for a computational model which supports all such requirements, would incur overheads so large, that little spare capacity would remain for the optional computations themselves. Following this, the report presents a *constrained computational model*, which, it is claimed, could be cost-effectively supported at run-time. The model is nevertheless general enough to satisfy many of the requirements for adaptivity. The constrained model uses Best Effort Admissions Policy to arbitrate between three categories of optional computation, each with its own utility level. The viability of the constrained model is demonstrated by simulation studies which compare the performance of the model to that of First-Come-First-Served Admissions Policy. © 2000 Elsevier Science B.V. All rights reserved.

*Keywords:* Real-time; Scheduling; Computational model; Guarantee algorithms; Flexible scheduling; Optional computations

---

---

\* Corresponding author. Address: Teesside University, School of Computing and Maths, Middlesbrough TS1 3BA, UK. Tel.: 44 1742 342678; fax: 44 1642 230527; e-mail: c.g.mcelhone@tees.ac.uk.

## 1. Introduction

According to Stankovic [14] the next generation of real-time systems will be more complex and capable of adaptivity, as well as meeting their time constraints for mission and safety-critical functions. Currently, the critical functions of real-time systems are guaranteed before run-time by performing a worst-case analysis of the system's timing and resource requirements. The result is that real-time systems are engineered to have spare capacity, under normal operation. The work of this report focuses on using the spare processing capacity to provide adaptivity within the system.

Liu [8] has divided real-time computations into those which are *mandatory* and those which are *optional*. Mandatory computations, for example critical tasks, must be guaranteed before run-time. In contrast, optional computations can provide adaptivity and need not be guaranteed before run-time. However, *optional computations* may have *firm* deadlines which means that, although missing such deadlines is not catastrophic, it could result in a significant loss of utility to the system. Liu describes [9] how optional computations can be used in the implementation of techniques which support adaptivity, such as Imprecise Computations, Sieve Functions, Multiple Versions, etc.

Optional computations can be scheduled, and even guaranteed, at run-time, by methods of *flexible scheduling*. The Spring project [15] supports *flexible scheduling* by allowing optional computations to be guaranteed, on-line, on the node on which they arise. If this is not possible, then attempts can be made to guarantee the optional computations on other nodes of the system. An algorithm used to guarantee the schedulability of optional computations is called a *guarantee algorithm*. A guarantee algorithm ascertains whether a

given optional computation can be scheduled within its deadline, according to whatever scheduling policy (e.g. Deadline Monotonic Scheduling [1]) is in force.

Each optional computation may be associated with a value called its *utility*. This value gives a measure of the service which the computation provides for the system when it has completed. *Admission policies* may be used to arbitrate between optional computation which are competing to be guaranteed, and have different utilities. An example of an admission policy is *Best Effort* [10] in which an optional computation of high utility may abort optional computations of lower utility, in order that the high utility computation may be guaranteed.

Flexible scheduling can require run-time support which incurs prohibitive overheads when executed on the same processor as the mandatory and optional computations themselves (see Section 3). Previous research efforts [13,16] have used specialised or additional hardware in order to reduce the effect of these overheads. This report presents a different approach by arguing that the run-time support for flexible scheduling can be made to perform cost-effectively on a conventional processor which also runs mandatory and optional computations. This is achieved by the adoption of a *constrained computational model*, which reduces the overheads required for run-time support.

Section 2 reviews some of the complex application requirements for adaptivity (such as task hierarchies with interdependent subtasks) which optional computations could support. Section 3 provides evidence that the provision of complex run-time support for such requirements would incur prohibitive overheads. Subsequent sections discuss and evaluate the constrained computational model which reduces the overheads of run-time support.

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات