

Chapter 3.6

A Scalable Approach to Real-Time System Timing Analysis

Alan Grigg

Loughborough University, UK

Lin Guan

Loughborough University, UK

ABSTRACT

This chapter describes a real-time system performance analysis approach known as reservation-based analysis (RBA). The scalability of RBA is derived from an abstract (target-independent) representation of system software components, their timing and resource requirements and run-time scheduling policies. The RBA timing analysis framework provides an evolvable modeling solution that can be instigated in early stages of system design, long before the software and hardware components have been developed, and continually refined through successive stages of detailed design, implementation and testing. At each stage of refinement, the abstract model provides a set of best-case and worst-case timing ‘guarantees’ that will be delivered subject to a set of scheduling ‘obligations’ being met by the target system implementation. An abstract scheduling model, known as the rate-based execution model then provides an implementation reference model with which compliance will ensure that the imposed set of timing obligations will be met by the target system.

INTRODUCTION

A key requirement in the development of a real-time system is the ability to demonstrate that the final target system has met its specified timing requirements. This can be carried out by

constructing a model of system timing behaviour that can be used to make predictions about worst-case performance in terms of maximum response times, communication delays, delay variations and resource utilisation. In order to develop a modelling solution that is scalable, however, the timing analysis model must be constructed with this goal specifically in mind.

DOI: 10.4018/978-1-4666-0879-5.ch3.6

Scalability of the timing analysis solution can ultimately be achieved by meeting two key modelling objectives:

- **Partitionable analysis:** The ability to model the timing behaviour of parts of the system independently from other parts and, correspondingly, limit the scope of re-analysis required in the event of localised changes;
- **Evolvable analysis:** The ability to assess the timing behaviour of the system incrementally throughout the development process (and through later modifications), based on the information available at each stage of development.

Partitionable Analysis

It was recognised by Audsley *et al* (1993) that the holistic nature of timing analysis models for distributed real-time systems (the inability to analyse the system in part due to circular interdependencies in the model) arises due to a combination of functional and physical integration effects. It therefore follows that the model could be partitioned by placing appropriate restrictions on the manner in which integration of the system software and hardware components is performed as follows:

- **Functional partitioning:** Constraints on data communication and ordering/ precedence relationships between related software components;
- **Physical partitioning:** Constraints on resource sharing policies (scheduling and communication protocols).

Functional partitioning can be implemented by breaking up end-to-end *transactions* (sequences of processing and communication activities required to meet some higher level system objective) into individual components based purely on knowl-

edge of the end-to-end timing requirements of the system (and the transaction topology). The net result is that the timing behaviour of groups of software components allocated to individual processors (and other resources) can potentially be analysed independently.

Functional partitioning, in its extreme, results in a ‘federated’ timing analysis model where resource boundaries in the physical architecture dictate the partitions in the timing model. The resulting model can only be applied/evaluated after the physical integration stage of system development has been performed, *i.e.* the allocation of software components to hardware processors (and other resources) has been defined and the details of the scheduling solution and communication protocols have been finalised.

Physical partitioning can be implemented by applying resource scheduling mechanisms that support temporal partitioning between the components being scheduled. The net result is that the timing behaviour of functionally-dependent groups of components, *ie.* end-to-end transactions, can be analysed independently. This offers some key advantages over functional partitioning. Firstly, for systems that embody a degree of safety-critical functionality, this means that there is inherent temporal isolation between safety-critical and non-safety-critical software components that share processing resources and/or communication media. Secondly, and more generally from the perspective of an overall system engineering process, there is potential for supporting independent development and verification of groups of functionally-related components for a target computing environment that is physically integrated. Moreover, the analysis can potentially be applied before the physical integration stage of development has been performed, *i.e.* much earlier in the development life of the system.

The *Reservation-Based Analysis* (RBA) method described in this chapter uses a physical partitioning approach throughout but also supports functional partitioning with the guideline that

30 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/scalable-approach-real-time-system/64507

Related Content

Monocular Vision System that Learns with Approximation Spaces

James F. Peters (2008). *Rough Computing: Theories, Technologies and Applications* (pp. 186-203).

www.irma-international.org/chapter/monocular-vision-system-learns-approximation/28474

Performance Analysis of Sequential and Parallel Neural Network Algorithm for Stock Price Forecasting

Rashedur M. Rahman, Ruppa K. Thulasiramand Parimala Thulasiraman (2013). *Applications and Developments in Grid, Cloud, and High Performance Computing* (pp. 97-121).

www.irma-international.org/chapter/performance-analysis-sequential-parallel-neural/69030

Automatic Dynamic Data Structures Recognition to Support the Migration of Applications to the Cloud

Beniamino Di Martinoand Antonio Esposito (2015). *International Journal of Grid and High Performance Computing* (pp. 1-22).

www.irma-international.org/article/automatic-dynamic-data-structures-recognition-to-support-the-migration-of-applications-to-the-cloud/141297

Migrating Android Applications to the Cloud

Shih-Hao Hung, Jeng-Peng Shiehand Chen-Pang Lee (2013). *Applications and Developments in Grid, Cloud, and High Performance Computing* (pp. 307-322).

www.irma-international.org/chapter/migrating-android-applications-cloud/69043

Privacy Preserving Distributed K-Means Clustering in Malicious Model Using Verifiable Secret Sharing Scheme

Sankita Patel, Mitali Sonarand Devesh C. Jinwala (2014). *International Journal of Distributed Systems and Technologies* (pp. 44-70).

www.irma-international.org/article/privacy-preserving-distributed-k-means-clustering-in-malicious-model-using-verifiable-secret-sharing-scheme/115253