Advance Access publication on doi:10.1093/comjnl/bxr075

Capsule Reviews

FAIROUZ KAMAREDDINE

The Capsule Reviews are intended to provide a short succinct review of each paper in the issue in order to bring it to a wider readership. The Capsule Reviews were compiled by Fairouz Kamareddine. Professor Kamareddine is an Associate Editor of *The Computer Journal* and is based in the Department of Mathematical and Computer Sciences at Heriot-Watt University, Edinburgh, UK.

A New Energy-aware Dynamic Task Set Partitioning Algorithm for Soft and Hard Embedded Real-time Systems. JOSE LUIS MARCH, JULIO SAHUQUILLO, HOUCINE HASSAN, SALVADOR PETIT AND JOSE DUATO

Dynamic voltage scaling (DVS) is a technique that allows the system to work at different frequency/voltage levels. In a previous work, the authors proposed a partitioning heuristic aimed at increasing the overlapping time between memory access and computation time and hence reducing energy consumption in coarse-grain multithreaded multicore processors working on a global DVS regulator. Since that work focused on hard real-time (HRT) and most current embedded applications run both HRT and soft real-time (SRT) tasks, this paper extends the earlier work by adding support for SRT tasks too. After the introduction to the modelled system, the partitioning and HRT heuristics, and the power-aware HRT scheduler, the authors provide support for SRT tasks and evaluate their proposed method for the partitioner extended with the power-aware scheduler for both HRT and SRT tasks. Related works on energy management are then discussed and results/conclusions are drawn for the extended heuristic scheduler.

Multiprocessors and Asynchronous Event Handling in the Real-Time Specification for Java. MINSEONG KIM AND ANDY WELLINGS

The goal of the real-time specification for Java (RTSJ) is to give a platform, a Java execution environment and application program interface (API), that allows programmers correctly reason about the temporal behaviour of executing software. Crucial to this goal is RTSJ's generalization of Java's mechanism for asynchronous event handling (AEH) by distinguishing events and event handlers and hence the light weight mechanism that allows event handlers to behave like threads. The authors explain how current RTSJ does not provide any guidelines on how events and their handlers can be implemented efficiently and how their timing requirements can be guaranteed. Then, the authors propose a new AEH mapping model for non-blocking handlers in multiprocessor systems, which achieve the lightweightness requirement by restricting the number of concurrent real-time threads. First, the authors explain AEH in RTSJ, how lightweightness can be achieved and the importance of limiting the number of server threads without jeopardizing the schedulability of the overall system. Previous approaches that reduce the number of server threads (in uniprocessor systems) required for blocking/non-blocking handlers are reviewed and a classification of AEH mapping models is given. Then, the critical sequence of non-blocking handlers in uniprocessor systems is extended for multiprocessor systems. On the basis of this extension, the authors derive the greatest lower and least upper bounds of server threads to execute an arbitrary number of released handlers for given numbers of priority levels, and the worst and best case behaviour is observed. Then, a formal model of the proposed AEH model is given with the purpose of increasing the level of its reliability and robustness using the automata formalisms of the UPPAAL toolbox for modelling validation and verification of real-time systems. The formal analysis of the constructed model is carried out to guarantee its liveness and safety and then, the model is evaluated with respect to the lightweight requirement of AEH in the RTSJ.

An Abstract Semantically Rich Compiler Collocative and Interpretative Model for OpenMP Programs. MOHAMMED F. MOKBEL, ROBERT D. KENT AND MICHAEL WONG

OpenMP is an application programming interface (API) for shared memory parallel programming architecture. Considering that incremental parallelism and sequential equivalence play an important role in OpenMP, the scalability of an OpenMP application is bounded by the efficient design of the original sequential source code. The authors state that there is neither a systematic way nor a defined model to provide a high-level dimensionality capable of assessing the behaviour and performance of OpenMP programs and compilers and set out to propose a comparative model that provides a methodical and well-defined approach to solve these problems from the compiler perspective. The model is based at a higher level of abstraction to reason about the most probable performance

problems. First, the model is defined using set notations that describe the relationships between program states and each compiler, and low-level details are conceptualized at a higher level of abstraction. Formulations are grouped into three categories: horizontal (those that work per state), vertical (those that work per compiler) and horizontal × vertical (those that provide a joint analysis about the overall performance of the compilers). A compilers' pattern detection algorithm is given and based on the states' definitions and formulations, the behaviour of an OpenMP program is characterized with respect to 2CA optimum domains through an OpenMP optimum performance characterization algorithm. In order to experimentally examine the proposed compiler comparative model, three (anonymous) commercial compilers are chosen and three C++ examples annotated with the appropriate OpenMP programs are used as test programs.

Embedded Socio-Oriented Model for End-to-End Reliable Stream Schedules by Using Collaborative Outsourcing in MP2P Systems. CONSTANDINOS MAVROMOUSTAKIS AND HELEN KARATZA

Resource-sharing platforms used in mobile peer-to-peer (MP2P) systems that can be applied into various sectors are important tools for effectively monitoring the physical world. However, vehicle-based devices require innovative solutions in efficient sharing of resources, exhibit pre-determined and distance-limitation-constrained mobility patterns, and have no strict limits on processing power and storage capabilities. Multimedia streaming in such environments faces the problems of reliability and wasted resources. This paper proposes an embedded framework for preserving and enabling the reliability in any requested path and for any requested resources. A new technique called the k-hop mobility-based cache replication strategy is proposed where a replication scheme takes into account the pre-determined vehicular mobility in a k-hop path in order to effectively disseminate and within a specified duration, any requested packets. After a review of the related work, the embedded socio-oriented model for end-to-end reliable stream schedules in introduced. This mechanism enables scalability of resources and acts as a middleware distinct mechanism that traverses the data, network, transport, and application layers and considers the scarceness degree of the requested resources. In order to avoid any redundant transmissions and retransmissions, the authors propose a cluster-based k-hop cooperation scheme and mobility model in MP2P devices. Community-oriented relay regions are introduced that identify the set of points in the plane for which communication through the relay node is more reliable than direct communications. Thereafter, the impact of chunk's blocks and the mobility pattern for low-latency outsourcing in collaborative caching are discussed. Experimental results are given to evaluate the proposed scheme.

An Empirical Study of Deploying Storage Class Memory into the I/O Path of Portable Systems. IN HWAN DOH, JONGMOO CHOI, DONGHEE LEE AND SAM NOH

Flash memory provides non-volatile storage in portable devices; however, their physical limitations lead to performance limitations. For this reason, this paper proposes storage class memory (SCM) to overcome these limitations and explores the impact of SCM on the performance of flash file systems for portable systems. Experimental results demonstrate that innovative deployment of SCM into the I/O path will satiate user requirements for portable systems in terms of I/O performance and battery life and will improve flash memory lifetime. After a discussion of the background and related work, the design and implementation of the MiNV file system (MiNVFS) are introduced. MiNVFS is a flash file system that exploits SCM as a metadata store and its performance benefits are discussed in detail especially with respect to storing flash file system metadata in SCM. It is shown that exploiting SCM in MiNVFS significantly reduces operations executed in flash memory by excluding both write operations and invalidated pages incurred by frequent metadata updates. The feasibility of deploying MiNVFS in real portable computing devices is assessed in theory and in practice. To do so, the SCM space requirement model is presented and is used to conduct the analysis. Thereafter, the hardware and software setup and the experimental environment for the performance evaluation of MiNVFS are presented and the correctness of the SCM space requirement model is validated.

The L(h,k)-Labelling Problem: An Updated Survey and Annotated Bibliography. TIZIANA CALAMONERI

The author states that while in the classical vertex colouring problem a condition is only imposed on colours of adjacent nodes, many generalizations require colours to respect stronger conditions. The paper concentrates on a specific graph-colouring generalization that has intensively been studied: the $L(h,k)$-labelling problem, which gives any fixed non-negative integer values h and k, consists in an assignment of non-negative integers to the nodes of a graph such that adjacent nodes receive values which differ by at least h, and nodes connected by a 2-length path receive values which differ by at least k. The goal of the problem is to find out an $L(h,k)$-labelling with a minimum span (i.e. a minimum difference between the largest and the smallest assigned frequency). This paper reviews the results of previously published literature from a graph algorithmic approach.

After listing some general complexity hard results divided by different values of h and k, followed by some general bounds on the $\lambda_{h,k}$-number and illustrating the gap between the upper and lower bounds, the author moves to describe further bounds, exact results and approximation algorithms found by restricting the classes of graphs under consideration.