doi:10.1093/comjnl/bxr044

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.

On the Topological Properties of Grid-Based Interconnection Networks: Surface Area and Volume of Radial Spheres. H. SARBAZI-AZAD, A. KHONSARI AND M. OULD-KHOUA

This paper aims at deriving topological properties of grid networks and at providing expressions for computing the number of nodes located within a certain distance from a certain node with respect to surface area and volume in various types of grids. After introducing basic definitions, notations and preliminaries in combinatorial foundation, the authors explain how to derive the surface area and volume of the radial sphere in both the mesh and the torus grids. Then, special cases that include the k-ary n-cube and the hypercube are considered for which the surface area and volume expressions are also derived. Thereafter, the application to performance modelling under uniform traffic load is considered as well as the application of modelling under hotspot traffic load.

Safety, Dependability and Performance Analysis of Extended AADL Models. MARCO BOZZANO, ALESSANDRO CIMATTI, JOOST-PIETER KATOEN, VIET YEN NGUYEN, THOMAS NOLL AND MARCO ROVERI

The authors argue that although several approaches to component-based design exist in the literature, error handling and modelling have not received the attention they deserve. Furthermore, the authors argue that there is a lack of good notation in such approaches. For these reasons, the paper proposes a practical component-based modelling approach with appropriate means for modelling probabilistic fault behaviour. The approach is based on the Architecture Analysis and Design Language which allows a good description of a number of operations and aspects as well as a dynamic reconfiguration of components and port connections. First, the System Level Integrated Modelling (SLIM) language is introduced with its nominal and error behaviours, and with its fault injection which describes the effect of the occurrence of an error on the nominal behaviour of the system. The semantics of SLIM is given and the fault injections are used to associate the error model to the nominal model leading to an extended model that represents the complete system behaviour. The new proposed framework is formally analysed for validation, model checking, safety and dependability, diagnosability and performance evaluation. A prototype tool is then presented and followed by a discussion of related work.

Fast Generation of t-ary Trees. JAMES F. KORSH

The paper starts by stating that of the algorithms for generating binary or t-ary trees represented by sequences of integers, only three generate the next tree directly from its predecessor. In an earlier work, these algorithms have been analysed for their average number of memory references and it was shown that the time complexity is proportional to these memory references. This paper presents new tree generation algorithms that generate the tree directly from its predecessor. After an overview of the background and motivation, a new binary tree generation algorithm is given and is then generalized to the t-ary version. Both algorithms generate a tree from its predecessor but neither is as efficient as existing algorithms. For this reason, the author proceeds to make these algorithms more efficient by keeping track of the parent of each node using an array and by using further measures to make these algorithms very fast. Three further algorithms are given with the purpose of achieving efficient tree generation. The efficiency of the various algorithms is analysed by determining the number of references required.

Local Memory Design Space Exploration for High-performance Computing. RAMON BERTRAN, MARC GONZALEZ, XAVIER MARTORELL, NACHO NAVARRO AND EDUARD AYGUADE

This paper argues that the optimal memory model for high-performance computing is not defined since it depends on design goals that include performance, energy consumption, scalability, area and programmability. Furthermore, all these memory models have a low-latency/high-bandwidth size-constrained stack of intermediate memories plus a

huge high-latency/low-bandwidth main memory. Intermediate memories are managed either implicitly by hardware or explicitly by software. This paper focuses on software-managed local memories (LMs) and proposes a hybrid memory model that combines an existing memory hierarchy with an LM plus a simple programmable DMA controller that performs asynchronous memory transfers. The new architecture is presented and implemented. Twenty-seven computational loops from four different benchmarks have been evaluated and performance results are presented. Two case studies are given and the overall evaluation is presented for performance, energy, energy delay product and area and access time considerations.

A Web-Based Personalized Mobility Service for Smartphone Applications. MURAT ALI BAYIR, MURAT DEMIRBAS AND AHMET COSAR

This paper aims to develop a personalized and lightweight mobility service for smartphone applications with two example applications (location prediction and air pollution exposure risk estimation). A review of related work is followed by the general system architecture TRACK ME of the web-based framework. Each component of the TRACK ME framework is given in detail. These components include the TRACK ME client software, which continuously logs cell tower connectivity data and periodically forwards location data via opportunistic http requests. Another component is the mobility profiler which is an off-line component that converts low-level location data units to high-level mobility profiles by periodically running the mobility path construction, eliminating noise over mobility paths, discovering mobility patterns and integrating data for mobility profiles. The most important component of the TRACK ME framework is the query engine, which provides application services with a rule-based query definition and an execution interface for the purpose of accessing mobility profiles. The TRACK ME framework and its components are illustrated for the two example applications (location prediction and air pollution exposure risk estimation). Experimental results emphasize the mobility profile discovery and the scalability of the system in terms of the complexity of queries forwarded to the query engine and the size of the location data processed by the whole framework. Detailed experimental results are discussed for the two example applications.

Data Cache Prefetching with Dynamic Adaptation. MINHAJ AHMAD KHAN

Cache prefetching allows data to be fetched from memory into the cache before processing, and enables better performance since cache miss penalties are not incurred if the processor finds the required data in the cache. However, the prefetch distance depends on the latency of the cache miss and the iteration of the process. Since latency cannot be statically determined, this paper proposes that latencies be calculated dynamically in order to obtain accurate prefetching of data. Dynamic cache prefetching fetches data into the cache at runtime and allows the adaptation of the prefetch distance in accordance with the runtime behaviour. This results in better performance. The author presents his approach to runtime data cache prefetching, which consists in generating templates at compile time, avoiding costly activities like code generation allocation of registers and scheduling of instructions. Static compile-time activities (like selection of variables, cost/benefit analysis and the generation of the template for run-time adaptation) are explained and then it is shown how the prefetch distance can be adapted at run-time. A prototype implementation has been carried out to perform dynamic cache prefetching with two different offsets resulting in two different versions which are compared. Finally, the paper discusses the experimental results and the related work.

An Optimal Rotation Distance Set. YEN-JU CHEN, JIA-JIE LIU AND YUE-LI WANG

A rotation transforms a binary tree T into another binary tree T' such that the symmetric order of T is preserved. The rotation distance $d(T, T')$ where T and T' are binary trees with the same number of leaves, is the minimum number of rotations needed to transform T into T'. It is not known whether rotation distances can be computated in polynomial time. A number of studies have been carried out on estimating rotations. One such study uses the left-weight sequences (LW-sequences) to characterize binary trees. This paper makes use of such characterization of binary trees into LW-sequences and defines a set S of binary trees such that the rotation distance between any two trees in S is the same and moreover, the computation of the rotation distance is linear in time. After an introduction to the preliminaries, left weight/right weight sequences, restricted LW-sequences (RLW-sequences) and RLW-trees are introduced along with the set containing all the RLW-trees of n internal nodes. This latter set has 2^{n-1} binary trees and is the restricted set for which the rotation distance is studied. An algorithm is given, which takes two RLW-sequences $w_l(T)$ and $w_l(T')$, finds the rotation distance between them and builds a sequence of transforming rotations between T and T'. The algorithm is then analysed in detail and it is shown that, given the RLW-sequences of two RLW-trees T and T' of nodes n, the algorithm produces exactly $d(T, T')$ rotations to transform T into T' in linear time.