doi:10.1093/comjnl/bxn056

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 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.

Reliability and Performance of Mirrored Disk Organizations. A. THOMASIAN AND JUN XU

There are seven classifications for redundant arrays of independent disks (RAID), known as RAID0 . . . RAID6. Of these classifications, RAID1 is a most popular paradigm for achieving fault tolerance and higher disk access bandwidth for read requests. As the RAID paradigm utilizes redundancy to attain high reliability, this paper investigates the performance and performability of several RAID1 organizations aimed at avoiding the shortcomings of the basic mirroring (BM) technique. First, the paper discusses the related work with a focus on mirrored disk-scheduling algorithms (see also [Thomasian]) and the analysis, performance and simulation of various mirrored disk-scheduling policies. In Section 3, the authors present the four RAID1 organizations whose performance and performability will be studied: BM (which is the most common type of mirroring), group rotate declustering (GRD, which balances the disk loads when a single disk failure occurs), interleaved declustering [ID, where each disk is divided into a primary and a secondary area with equal capacities and where the read load of the surviving disks (after one single disk fails) is balanced and increased] and chained declustering (CD, which is a variation on ID which attains a higher reliability level than ID). In Section 4, the authors review reliability analysis, summarize the reliability expressions for BM, GRD, ID and CD and propose an alternative approach. In Section 5, the four RAID1 organizations are compared from the point of view of their mean time to data loss (MTTDL) and their performability measure. Under the reliability comparison, the authors find that BM is the most reliable technique, CD is the next, followed by ID followed by GRD. To develop formulae to compare the relative performance of the four RAID1 organizations in fault-free modes and with disk failure, the paper presents the modelling assumptions and gives the fault-free normal mode of operation followed by the degraded mode. The BM organization cannot balance the disk loads in degraded mode whereas GRD can balance the primary and secondary disk loads. Similarly, balancing is possible in ID and CD organizations. Finally, the authors compare the performance of RAID1 organizations with RAID0, RAID5 and RAID6 disk arrays and conclude.

A Review of SIMD Multimedia Extensions and their Usage in Scientific and Engineering Applications. HASSABALLAH, OMRAN AND MAHDY

Single instruction multiple data (SIMD) are a set of instructions that can speed up an application performance by allowing a basic operation to be performed on multiple data elements in parallel. Such extensions to multimedia achieve higher performance by processing more data with fewer instructions. The paper discusses the SIMD model and the technique in detail, concentrating on the most common supporting technologies and microprocessors. The paper reviews in detail the research that has taken place to improve multimedia application performance using SIMD extensions. In particular, a comprehensive overview is carried out of the research that makes use of Intel's SIMD multimedia extensions on a single processor to enhance the performance of a number of applications like data security, databases and scientific applications (e.g. the comparison of DNA and protein sequences). The paper is well informed on both the theoretical and implementation aspects of the literature and makes useful suggestions for future research.

Affordance and Symmetry in User Interfaces. PAUL CAIRNS AND HAROLD THIMBLEBY

This paper aims to provide a more operational definition for affordance [a notion that was coined by Gibson to mean the interactive properties of an object recognized from (usually visual) stimuli].

The authors put forward the idea that the key to their approach is that the aspects of affordance can be explained in terms of symmetries. While doing so, the authors formalize symmetry from the point of view of human–computer interaction (HCI). First, the authors discuss the appeals and/or uses of affordance and symmetry from different view points. Then, they set out to carefully formalize symmetry with the aim of ensuring correct applications in situations involving

humans, explicitly. Throughout this formalization, the authors compare with the definition of symmetry given by Weyl and emphasize the importance of expliciting the perception of symmetry. Furthermore, the authors exploit the idea that symmetries propose actions that a user can do. This formalization of symmetry (and its perception) is used to represent how affordances work. This results in a sufficiently rich machinery that is able to explain how symmetry can be used in user interfaces and to deduce a number of observations (e.g. symmetry–affordance suggests commutativity). To illustrate their theory especially for affordance, the authors demonstrate how the theory works for common examples of affordance. One particular example is the breaking of symmetry–affordance in situations that include the different roles of the % key on different calculators [Thimbleby]. These case studies lead to a number of insights as discussed in the paper.

The Collective Index: A Technique for Efficient Processing of Progressive Queries. ZHU, MEDJAHED, SHARMA AND HUANG

Data-intensive applications require sophisticated database techniques for processing advanced types of queries and for querying vast amounts of data. A query is progressive if it is formulated progressively in more than one step, each of which is called a step-query. A progressive query cannot be known beforehand and poses challenges to any attempt of processing it efficiently. This paper proposes the so-called collective index technique that aims to efficiently process progressive queries. A special index structure is employed to maintain member indexes for processing special progressive queries (basically transforming member indexes on an input relation of a step-query into member indexes on the result relation to be utilised by the following step-query). First, the paper presents the progressive query model where dependencies between step-queries are formalized and the two types of progressive queries are given (the single-input linear and the multiple input linear). Five strategies for processing progressive queries are presented. The collective index technique is explained in detail for single-input linear progressive queries where a step-query algorithm through a collective index is given and its advantages and performance are discussed. Then, this collective index technique is revisited to deal with multiple input linear progressive queries. The step-query algorithm for single-input linear progressive queries is extended to deal with the multiple input case. Extensive experiments are reported, which illustrate the performance of the algorithms of both the single and multiple input progressive queries.

The Availability of Complemental *k*-Coteries. KUO AND WU

The distributed *k*-mutual exclusion problem is the problem to guarantee that in an environment where there are *k*-identical shared resources each of which can be accessed by only one computing node at a time; the nodes accessing a shared resource must execute in a mutually exclusive way. One algorithm to solve the *k*-mutual exclusion problem is based on the quorum concept where a quorum is a set of some nodes. A *k*-coterie is a collection of quorums that possess at most *k* mutually disjoint quorums. This intersection property of *k*-coteries guarantees the *k*-mutual exclusion. Two metrics measure the fault-tolerant capability of a *k*-coterie: its availability and its complementalness. This paper studies the availability of the complemental *k*-coterie and shows that every complemental *k*-coterie has the same characteristic on its availability and hence, verifying complementalness and evaluating availability could be combined to reduce the redundancy work on measuring fault-tolerance. The paper starts by introducing *k*-coteries and complemental *k*-coteries. Then, it formalizes the availability of complemental (ND) *k*-coteries establishing the relation between a complemental *k*-coterie and its availability, and giving necessary and sufficient conditions for a *k*-coterie to be complemental. The paper uses this formalization to verify complemental *k*-coteries through availability. The paper also extends its method to the so-called wr-coteries.

Vertex-ordering Algorithms for Automatic Differentiation of Computer Codes. TADJOUDDINE

The finite-differencing method of calculating functions derivatives incurs truncation errors that could lead to loss of robustness. Automatic differentiation (AD) computes the derivatives without truncation errors. In AD, the original code is broken into a code list where additional statements compute the derivatives using an elimination algorithm that helps calculate Jacoboians and permits to exploit the sparcity of computation. Two standard AD algorithms calculate the derivatives: the forward mode and the reverse mode. These two algorithms are closely related to the elimination algorithm mentioned above. Furthermore, it is possible to mix these forward and reverse mode algorithms to obtain the so-called interface contraction (or vertex cut). The paper concentrates on finding elimination sequences that reduce floating-point operations and memory traffic and on giving an algorithm that mixes backward and reverse AD elimination. The paper shows that, by doing so, interface contraction is minimum and Jacobian codes can be obtained much faster. First, the paper gives the necessary background on the automatic computation of Jacobians explaining how elimination sequences determine the number of multiplications required by the Jacobian accumulation which is calculated by eliminating intermediate vertices. Owing to the presence of numerous elimination sequences, the paper proposes the use of heuristics from sparse matrix technology in the process of choosing a sequence and to efficiently accumulate the Jacobian. Recall that the paper's main purpose is to increase the efficiency of

AD by elimination and for this, the paper attempts to reduce the memory traffic by making efficient use of the processor's registers and cache. Hence, the paper introduces a statement reordering algorithm where the data dependency graph is preprocessed to compute a ranking function that gives relative priority to each vertex. Then, an interface interaction algorithm is given to partition the graph so that both forward and reverse AD algorithms can be suitably used. The paper illustrates the benefits of exploiting interface contraction through a motivating example. Ample discussions of experiments, implementation and tests illustrate the benefits of the method proposed.

An Introduction to Reversible Latches. RICE

This paper starts by the observation that the use of reversible logic in building chips may provide a solution for the so-called reversible computing where incremental improvements can be made so that devices can use less power and batteries can last longer. The paper addresses the issues raised regarding the use of sequential logic in a reversible context and refutes the claim that sequential reversible networks are not possible, arguing that the problem lies in the implementation and the target technology. Reversible logic is presented in detail and the concept of a set–reset (SR) latch is presented. Two reversible latch designs are introduced and analysed in detail: the Fredkin-based SR latch and the Taffoli-based SR latch.

The behaviour of these latches is studied and compared as is their speed. The paper concludes that both designs are very similar in behaviour. Both the Fredkin and the Taffoli latches are investigated further using other considerations.

Visual Cryptography Schemes with Reversing. YANG, WANG AND CHEN

Usual visual cryptography schemes (VCSs) reconstruct the encoded secret without computation, by superimposing shadow images. However, VCSs cannot reconstruct the ideal contrast image. Earlier efforts at using the so-called reversing-based VCSs allow an almost ideal contrast image that can only be constructed using perfect black VCS (PBVCS) where the black pixel is perfectly reconstructed. This paper proposes a real perfect contrast VCS with reversing where an ideal (rather than almost ideal) image is achieved by using the non-perfect black VCS (NPBVCS). After a careful introduction of both VCS and VCS with reversing and of the importance of reversing-based VCSs, the paper introduces the real perfect contrast VCS (RPCVCS) with reversing based on PBVCS followed by the RPCVCS based on NPBVCS. The security of both proposed RPCVCS is studied and it is shown that they are both secure. Experimental results are given where a number of issues are studied. These issues include, amongst others, the problem of stacking shadows in different runs, decoding complexity and compatibility.