Richard Cole, Ernst W. Mayr,
Friedhelm Meyer auf der Heide (editors):

Parallel and Distributed Algorithms

Dagstuhl-Seminar-Report; 72
13.09.-17.09.93 (9337)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

INTERNATIONALES BEGEGNUNGS- UND FORSCHUNGSZENTRUM FÜR INFORMATIK

Schloß Dagstuhl

Seminar Report 9337

Parallel and Distributed Algorithms

September 13 – 17, 1993

OVERVIEW

The second Dagstuhl Seminar on *Parallel and Distributed Algorithms* was organized by Richard Cole (Courant Institute, New York), Ernst W. Mayr (TU München), and Friedhelm Meyer auf der Heide (Universität Paderborn). It brought together 25 participants from 6 countries, 7 of them came from overseas.

The 25 talks presented covered a wide range of topics including parallel data structures, models of parallel computing, parallel complexity theory, and efficient algorithms for PRAMs, interconnection networks, and Boolean circuits, and randomization techniques for parallel algorithms.

The abstracts of all talks can be found in this report.

An interesting guided excursion to the ancient Roman city, and therefore oldest German city, of Trier took place on Thursday.

The outstanding environment and organization of Schloß Dagstuhl greatly contributed to the success of the seminar.

Reported by Rolf Wanka

Participants

Greg Barnes, MPI Saarbrücken
Sandeep Bhatt, Bellcore Morristown
Gianfranco Bilardi, Universitá di Padova
Richard Cole, Courant Institute, New York
Artur Czumaj, Heinz-Nixdorf-Institut, Paderborn
Torben Hagerup, MPI Saarbrücken
Juraj Hromkovič, Universität Paderborn
Marek Karpiński, Universität Bonn
Lefteris Kirousis, University of Patras
Manfred Kunde, TU München
Mirosław Kutyłowski, Universytet Wrocławski
Klaus-Jörn Lange, TU München
Reinhard Lüling, Universität Paderborn
Bruce Maggs, NEC Research Institute, Princeton
Ernst W. Mayr, TU München
Friedhelm Meyer auf der Heide, Universität Paderborn
Naomi Nishimura, University of Waterloo
Prabhakar Ragde, University of Waterloo
Wojciech Rytter, University of Warsaw
Alan Siegel, Courant Institute, New York
Paul Spirakis, University of Patras
Prasoon Tiwari, University of Wisconsin-Madison
Rolf Wanka, Universität Paderborn
Ingo Wegener, Universität Dortmund
Ralph Werchner, Universität Frankfurt

Contents

Abstracts

A Method for Implementing Lock-Free Shared Memory Data Structures

by GREG BARNES

We are interested in implementing data structures on shared memory multiprocessors. A natural model for these machines is an asynchronous parallel machine, in which the processors are subject to arbitrary delays. On such machines, it is desirable for algorithms to be *lock-free*, that is, they must allow concurrent access to data without using mutual exclusion. Efficient lock-free implementations are known for some specific data structures, but these algorithms do not generalize well to other structures. For most data structures, the only previously known lock-free algorithm is due to Herlihy. Herlihy presents a simple methodology to create a lock-free implementation of a general data structure, but his approach can be very expensive.

We present a technique that provides the semantics of exclusive access to data without using mutual exclusion. Using this technique, we devise the *caching method*, a general method of implementing lock-free data structures that is provably better than Herlihy's methodology for many well-known data structures. The cost of one operation using the caching method is proportional to $T \log T$, where T is the sequential cost of the operation. Under Herlihy's methodology, the cost is proportional to $T + C$, where C is the time needed to make a logical copy of the data structure. For many data structures, such as arrays and *well connected* pointer-based structures (e.g., a doubly linked list), the best known value for C is proportional to the size of the structure, making the copying time much larger than the sequential cost of an operation. The new method can also allow *concurrent updates* to the data structure; Herlihy's methodology cannot. A correct lock-free implementation can be derived from a correct sequential implementation in a straightforward manner using this method. The method is also flexible; a programmer can change many of the details of the default implementation to optimize for a particular pattern of data structure use.

Models for Distributed Memory Communication

by SANDEEP BHATT (joint work with Pangfeng Liu and William Aiello)

We present a simple model to investigate the effect of contention in message-passing network systems. Within one time step, each processor in the "atomic message-passing model" can receive one message, perform local computation and send one message. When several messages from distinct sources are destined to the same processor then one is transmitted and the rest are blocked. Blocked messages cannot be retrieved by their sending processors; each processor must wait for its blocked message to clear before sending more messages into the network. Depending on the traffic pattern, messages can remain blocked for arbitrary number of steps.

The atomic model is conservative with existing message-passing architectures. Nonetheless, we prove linear speedup for backtrack and branch-and-bound searches using simple randomized algorithms. We will also present a number of problems that remain to be solved.

An Area-universal VLSI Circuit

by GIANFRANCO BILARDI (joint work with Paul Bay)

An *area-universal* VLSI circuit is proposed which has area $O(A)$ and can simulate any synchronous digital VLSI circuit of area A with only $O(\log A)$ slowdown. The universal circuit is a hybrid of a mesh and a concentrator fat-tree and accomplishes its simulations by a combination of routing techniques. It could be viewed as a Field Programmable Gate Array (FPGA). The layout of the simulated circuit is "compiled" into a "program" that is loaded into control registers of the universal circuit. The compilation takes $O(A)$ sequential time.

Two-dimensional Pattern Matching

by RICHARD COLE (joint work with Crochemore, Galil, Gasieniec, Hariharan, Muthukrishnan, Park, Rytter)

The talk described a parallel algorithm for the problem for the CRCW PRAM. The problem is to find all instances of an $m \times m$ pattern in a larger $n \times n$ text; each pattern and text entry is a character over some arbitrary alphabet. Preprocessing of the pattern, but not of the text, is allowed. The preprocessing has a complexity of $O(\log \log m)$ time and performs $O(m^2)$ operations and the main (matching) algorithm takes $O(1)$ time and performs $O(n^2)$ operations. All these results are optimal.

The Matrix Chain Product Problem

by ARTUR CZUMAJ

The talk will deal with the problem of finding an optimal order of the multiplication chain of matrices and the problem of finding an optimal triangulation of a convex polygon. For both these problems the best sequential algorithms run in $\Theta(n \log n)$ time. All parallel algorithms previously known use the dynamic programming paradigm and run in a polylogarithmic time using, in the best case, $O(n^6/\log^6 n)$ processors. We give a new algorithm which uses a different approach and reduces the problem to computing certain recurrence in a tree. We show that this recurrence can be optimally solved and describe an algorithm that runs in $O(\log^3 n)$ time using $n^2/\log^3 n$ processors on a CREW PRAM.

We also show that the problem of finding an optimal triangulation in a monotone (bitonic) polygon can be solved in $O(\log^2 n)$ time using n processors on a CREW PRAM.

Optimal Parallel String Operations

by TORBEN HAGERUP

We study fundamental ordering operations on strings of characters, equipped with the usual lexicographical ordering. For each problem studied, we give an algorithm that is optimal with respect to at least one criterion for which no optimal algorithm was previously known. Specifically, our main results are:

- Two sorted sequences of strings, containing altogether n characters, can be merged in $O(\log n)$ time using $O(n)$ operations on an EREW PRAM. This is optimal as regards both the running time and the number of operations.
- A sequence of strings, containing altogether n characters represented by integers of size

polynomial in n, can be sorted in $O(\log n/\log\log n)$ time using $O(n\log\log n)$ operations on a CRCW PRAM. The running time is optimal for any polynomial number of processors.

- The minimum string in a sequence of strings containing altogether n characters can be found in $O(\log\log n)$ time using $O(n)$ operations on a CRCW PRAM. The number of operations is optimal, and the running time is optimal for any algorithm using at most n processors.

Dissemination of Information in Edge(Vertex)-Disjoint Paths Modes

by JURAJ HROMKOVIČ (joint work with R. Klasing, E. Stöhr, W. Unger, H. Wagener)

The gossip complexity of fundamental interconnection networks in disjoint-paths modes of communication is investigated. Several effective (in some cases optimal) gossip algorithms are designed. All algorithms follow a very natural concept of so-called three-phase design. Some lower bounds (in some cases very close to the achieved upper bounds) on the gossip complexity of networks are presented. A list of open problems is presented and a connection between the gossip complexity of constant-degree graphs and the size of permutation networks for some subclasses of permutations is shown.

Explicit Simulation of Small Depth Threshold Circuits

by MAREK KARPINSKI (joint work with M. Goldmann)

We prove that the single threshold gate can be simulated by an explicit polynomial size depth 2 majority circuit. In general, we show that a depth d threshold circuit can be simulated uniformly by a majority circuit of depth $d+1$. Our construction answers two open problems of Goldmann, Hastad and Razborov (1992): we give the first explicit construction for which they use a randomized existence argument, and we show that such a simulation is possible even if the depth grows with the number of variables n (the simulation of Goldmann, Hastad and Razborov gives polynomial size circuits only when d is constant). Our result entails the first explicit constructions for the optimal depth, polynomial size majority circuits for the number of basic functions including, among others, powering (depth 3), integer multiplication (depth 3), and integer division (depth 3).

Efficient Algorithms for Checking the Atomicity of a Run of Read and Write Operations

by LEFTERIS KIROUSIS (joint work with A. Veneris)

Let $X_1,\ldots,X_c$ be variables shared by a number of processors $P_1,\ldots,P_q$ which operate in a totally asynchronous and wait-free manner. An operation by a processor is either a write to one of the variables or a read of the values of *all* variables. Operations are *not* assumed to be instantaneous and may arbitrarily overlap in time. A succession of possibly overlapping operations $a_1,\ldots,a_n$ (i.e., a run) is said to be atomic, if these operations can be serialized in a way compatible with any existing precedences among them and that any read operation returns for each variable the value of the most recent — with respect to the serialization — write operation on this variable. This talk examines the complexity of the combinatorial problem of testing a run for atomicity. First it is pointed out that when there is only one shared variable or when only one processor is allowed to write to each variable, known theorem lead

to polynomial-time algorithms for checking the atomicity of a run (the variable of the time-complexity function is the number of operations in the run). It is then proved that checking atomicity has polynomial-time complexity in the general case of more than one variables and with all processors allowed to read and write. For the proof, the atomicity problem is reduced to the problem of consecutive 1s in matrices. The reduction entails showing a combinatorial result which might be interesting on its own. Moreover, our algorithm can be used to efficiently construct an atomic serialization of the run, if there is one.

The All-to-all Mapping and its Impact on Sorting and Routing

by MANFRED KUNDE

We present a new method for deterministic sorting and routing on nets of processors. The method is based on a standard global routing routine, the all-to-all mapping, and local sorting procedures. On r-dimensional $n \times \cdots \times n$ grids of processors we apply this method to $h-h$ problems, $h \geq 4r$, where each processor initially and finally contains at most h elements. We show that the general $h-h$ sorting as well as $h-h$ routing problem can be solved within $hn/2 + o(hr^2n)$ steps. That is, the bisection bound is asymptotically tight for deterministic $h-h$ sorting and $h-h$ routing. On an r-dimensional torus, a grid with wrap-arounds, the number of transfer steps is $hn/4+o(hrn)$, again matching the corresponding bisection bound. This shows that in spite of the fact that routing problems contain more information at the beginning than the sorting problems there is no substantial difference between them on grids and tori.

By the help of the new method we can show for the first time that sorting and routing on meshes with diagonals can be performed faster than the best possible algorithms on meshes without diagonal connections.

Periodic Constant Depth Sorting Networks

by MIROSŁAW KUTYŁOWSKI (joint work with Marcin Kik and Grzegorz Stachowiak)

Comparator networks of constant depth can be used for sorting in the following way. The computation consists of a number of rounds, say t, each round being a single run through the comparator network. The output of a round j ($j < t$) is used as the input for the round $j+1$. The output of the round t is the output of the computation. In such a way, we may apply a small comparator network for sorting long input sequences. The main problem, however, is achieving a good computation time.

Odd-Even Transposition Sort gives a periodic sorting network of depth 2, that sorts n numbers in $n/2$ iterations. The network of depth 8 proposed by Schwiegelshohn sorts n numbers in $O(\sqrt{n}\log n)$ iterations. Krammer modified the algorithm and obtained a network of depth 6 sorting in $O(\sqrt{n\log n})$ iterations.

For a fixed but arbitrary $k \in I\!N$, we present a periodic sorting network of depth $O(k)$ that sorts n input numbers in $O(n^{1/k})$ steps.

Problems of Parallel Complexity Theory

by KLAUS-JÖRN LANGE

The current parallel complexity theory offers the class NC as a tool to characterize those problems inside of P, which possess efficient parallel algorithms. There are certain aspects which let appear this approach unsuitable. These objections may be arranged into two groups.

The Growth Problem: The original notions of sequential complexity theory to observe completeness results are polynomial time and logspace many-one reductions. These have a polynomial length of output and hence lead to complexity classes which are closed under polynomial growth. This carried over to newer types of reducibility notions like projections or circuit based reductions.

This led to very robust classes and hence to a very rich structure. But these reductions are not suitable to consider or measure speed-ups and efficiencies of parallel programms. To look at these, a linear bound on the output length seems to be necessary. On the other hand, linear growth doesn't seem to be robust enough to allow for a reasonable structure. Some approaches to circumvent this difficulty are presented.

Machine Problem: Most reducibility notions are based on sequential machines. This is the reason that they are not capable in dealing with communication problems, in particular with problems like latencies, topologies or hierarchical memories. There are many attempts to model these problems under the aspect of preserving constants and yielding good predictions. But it seems very hard to get appropriate models which are sufficiently robust. Two ways to deal with this difficulty are presented.

Dynamic Distributed Load Balancing

by REINHARD LÜLING (joint work with Burkhard Monien)

The overall efficiency of parallel algorithms is most decisively effected by the strategy applied for the mapping of workload. Strategies for balancing dynamically generated workload on a processor network which are also useful for practical applications have intensively been investigated by simulations and by direct applications.

Within the talk we present the theoretical analysis of a dynamically distributed load balancing strategy. The algorithm is adaptive by nature and is therefore useful for a broad range of applications. A similar algorithmic principle has already been implemented for a number of applications in the areas of combinatorial optimization, parallel programming languages and graphical animation. The algorithm performed convincingly for all these applications.

In our analysis we prove that the expected number of packets on each processor varies only by a constant factor compared with that on any other processor, independent of the generation and consumption of workload on each processor. We give exact bounds for these values and prove an exact upper bound, independent of the number of processors. Thus, the algorithm achieves a well-balanced workload distribution on the network for any underlying application. We also prove that the variation of the expected number of packets on a processor is very small and only dependent on the parameters of the algorithm. Furthermore, we present some analysis of the costs of our algorithm. We will also show that all tradeoffs between balancing quality, variation and costs can be determined by the parameters of the algorithm.

Reconfiguring Parallel Computers with Faulty Components

by BRUCE MAGGS (joint work with Richard Cole, Tom Leighton, and Ramesh Sitaraman)

This talk shows that a parallel computer whose underlying communication network is a tree, mesh, or butterfly can sustain many faults and still emulate a fault-free computer of the same size and type with little slowdown. The type of faults that we consider are failures of the nodes (processors) and edges (communication links) of the network. We assume that faults are static and detectable, and that information regarding the locations of the faults can be used to reconfigure the network. We show, for example, that even if an adversary is allowed to place a set of $N^{1-\epsilon}$ worst-case faults in an $N \times N$ mesh, for any fixed $\epsilon > 0$, the network can still emulate a fault-free $N \times N$ mesh with only constant slowdown. Before this research, there was no example of any connected bounded-degree network that could sustain more than a constant number of worst-case faults without suffering more than constant slowdown.

Scheduling Interval Orders in Parallel

by ERNST W. MAYR

Interval orders are partial orders defined by having interval representations. It is well known that a transitively oriented digraph G is an interval order iff its (undirected) complement $\bar{G}$ is chordal. We investigate parallel algorithms for the following scheduling problem: Given a system consisting of a set $\mathcal{T}$ of n tasks (each requiring unit execution time) and an interval order $\prec$ over $\mathcal{T}$, and given m identical parallel processors, construct an optimal (i.e., minimal length) schedule for $(\mathcal{T}, \prec)$.

Our algorithm is based on a subroutine for computing so-called scheduling distances, i.e., the minimal number of time steps needed to schedule all those tasks succeeding some given task t and preceding some other task t'. For a given interval order with n tasks, these scheduling distances can be computed using n^4 processors and $O(\log^2 n)$ time on a CREW-PRAM. We then give an incremental version of the scheduling distance algorithm, which can be used to compute the empty slots in an optimal schedule. From these, it is straightforward to derive the optimal schedule, requiring no more resources than for the initial scheduling distance computation.

The algorithm can also be extended to handle task systems which, in addition to interval order precedence constraints, have individual deadlines and/or release times for the tasks. Our algorithm is the first NC-algorithm for this problem.

Shared Memory Simulations: Theory and Experiments

by FRIEDHELM MEYER AUF DER HEIDE (joint work with Martin Dietzfelbinger, Christian Scheideler, and Uwe Kastens/Alf Wachsmann/Friedrich Wichmann)

A distributed memory machine (DMM) consists of n processors and memory modules. Each processor can access any cell in any module in unit time, but concurrent accesses to cells in the same module are not possible. A CRCW-DMM tolerates attempts to concurrently access the same module, and processes one of the requests. A c-Collision DMM also tolerates concurrent accesses, but only processes a request, if at most c requests arrive at the module concurrently.

We present PRAM simulations based on hashing the shared memory cells to the modules and show:

- a scheme using three hash functions that yields a simulation with expected delay $O(\log \log n)$ on 2-Collision DMMs,
- a refinement using nine hash functions that runs on 1-Collision DMMs,
- simple time-processor optimal simulations that run on CRCW-DMM's (using one hash function) and on 2-Collision DMM's (using three hash functions).

Finally we report on experiments to implement hashing strategies for virtualizing global variables on Transputer networks, and describe our OCCAM extension OCCAM light.

Using Trees to Obtain Parallel Algorithms for Graphs

by NAOMI NISHIMURA (joint work with Arvind Gupta)

Many difficult or even intractable graph problems can be easily solved in parallel when the inputs are trees. One can make use of a tree-like representation of a graph (the tree-decomposition of a graph) to apply techniques used to solve problems on trees to solve problems on more general classes of graphs. For the problem of subgraph isomorphism on partial k-trees, we present an NC algorithm when the source graph has bounded degree and an RNC algorithm when the source graph is k-connected; the problem is known to be NP-complete when both inputs are unrestricted partial k-trees. Generalizations of subgraph isomorphism, namely topological embedding and minor containment, are considered for similar classes of partial k-trees.

Parallel Algorithms for Memory-mapped Persistent Stores

by PRABHAKAR RAGDE (joint work with P. Buhr, A. Goel, and N. Nishimura)

Traditional methods of managing large persistent data files have several drawbacks: they require mechanisms external to the programming language (file systems); they require filtering or conversion of pointer-based structures; they are type-unsafe; they are usually non-portable. These drawbacks can be avoided by extending virtual memory to cover all of external storage, thereby eliminating explicit I/O. Our work (in progress) aims to demonstrate the effectiveness of this approach in shared-memory concurrent environments. We have developed a performance model which will be validated by experiments on a prototype system coded in C++ with concurrent extensions. Our initial experiments use parallel versions of algorithms for database join (such as sort-merge and hybrid hash). In the future, we will also look at fundamental structures and operations used in CAD/CAM, text management, and GIS.

Parallel Construction of Optimal Alphabetic Trees

by WOJCIECH RYTTER (joint work with L. Larmore and T. Przytycka)

The problem is to construct a full binary tree of minimal cost for a given sequence of positive weights of n items. The items are to be stored in the leaves in the same (left to right) order as given by the input sequence. The best previously known algorithm working in polylogarithmic time employed around n^6 processors. We construct a completely different algorithm and reduce the number of processors to n^2. This is done by parallelizing a version of the Garsia-Wachs version of the Hu-Tucker algorithm. The input sequence of items is decomposed into special segments called 1-valleys. Each of them is processed independently in parallel. This

can be done efficiently by reducing the tree construction problem for such special sequence to the concave least weight subsequence problem.

Remarks on Limited Independence

by ALAN SIEGEL

Part I is a probabilistic inequality: Let $X_1, X_2, \ldots, X_k$ be random variables with zero means, and let $H(X_i, a)$, for $a \geq 0$, denote the Chernoff-Hoeffding estimate for $Prob\{X_i \geq a\}$, i.e. the overestimate $\min_\lambda e^{-\lambda a} \mathrm{E}[e^{\lambda X_i}]$. Suppose $X = X_1 + x_2 + \ldots + X_k$ and the X_i might even be worst case dependent. Suppose $H(X_i, a_i) \leq C$, for $i = 1, 2, \ldots, k$, and $\sum_i a_i = a$. Then $H(X, a) \leq C$. More succinctly,

$$H(X, a) \leq \inf_{a_1 + a_2 + \ldots + a_k} max_i H(x_i, a_i).$$

The X_i need not have similar distributions.

Part II concerns the construction of Universal Families of highly random hash functions that can be evaluated by the *exclusive or* of a finite number of random seeds in a moderately large array. Array access follows edges in an expander-like bipartite graph. Nearly matching upper and lower bounds are presented, but the upper bounds are not presently achievable, because there are no graphs known to exhibit the good properties needed, despite a simple probabilistic existence argument. A simple fix gives constructions, but at the price of exponentiating the number of requisite exclusive or's.

Fast Parallel Approximations to P-complete Problems and Beyond

by PAUL SPIRAKIS (joint work with J. Diaz , M. Serna , J. Toran, and L. Kirousis)

We present here recent and new results on fast parallel approximations to problems that are hard to parallelize. We discuss five paradigms for efficient parallel approximations.

(1) The dense graph properties paradigm where a particular algorithm that eliminates vertices of low degrees is shown to produce in NC constant ratio approximations for any graph problem satisfying a certain property of a certain class of extremal graph properties.

(2) By suitably modifying the L-reductions of the class Max-SNP to be log-space, we show that any problem in SNP admits constant ratio NC approximations .

(3) We discuss the primal-dual method of Luby and Nisan which shows that positive LP can be approximated in NC.

(4) We present the scaling down method for problems whose hardness is due to size of numbers. We show that general Max Flow admits a fully RNC approximation scheme and an NC approximation scheme. We show the same for Maximum Matching.

(5) We overview the method of decomposing in NC a planar graph into k-outerplanar components (for fixed k) and discuss the constant ratio approximations obtained for MIS and other planar graph problems by this decomposition.

Scheduling Malleable and Nonmalleable Parallel Tasks

by PRASOON TIWARI (joint work with Walter Ludwig)

A *malleable* parallel task is one that can be executed on any number of processors, with its execution time being a function of the number of processors allotted to it. A *nonmalleable* parallel task is one that requires a specific number of processors. In addition, for various parallel architectures, it may be that not only the number of processors but also their configuration is a factor. In this talk, we show that for *any* architecture, any algorithm for scheduling nonmalleable tasks on that architecture can be extended to an algorithm for scheduling malleable tasks on the same architecture. In many cases, the approximation factor of the new algorithm is identical to that of the original algorithm. Meanwhile, only an $O(mn)$ term is added to the running time, where m is the number of processors and n is the number of tasks. Thus we get the best known approximation factor of a polynomial-time algorithm for scheduling malleable parallel tasks, and the fastest running time in which that factor can be achieved, for all architectures. Furthermore, we provide an algorithm for scheduling malleable tasks for the case when the addresses of the processors assigned to each task must be contiguous. The algorithm has an approximation factor of 2, under a natural assumption on the task execution times. In contrast, the best known approximation factor for nonmalleable tasks in this case is 2.5. To the best of our knowledge, this is the only case where the best known approximation factor for malleable tasks is strictly less than that for nonmalleable tasks.

The Shearsort Algorithm for Three-dimensional Meshes

by ROLF WANKA (joint work with Mirosław Kutyłowski)

We consider the following generalization of the sorting procedure ShearSort to multi-dimensional meshes of processors: Initially, each node contains one number. We proceed in rounds each round consisting of sorting the rows of the mesh according to the snake-like order where the dimensions are passed through in descending order. The execution terminates when the numbers stored in the mesh are sorted according to the snake-like order. Up to now, there were no small bounds on the number of rounds known. We examine the 3-dimensional case and show the following bounds: For $l \times l \times l$-meshes, l arbitrary, $2\log l + \Theta(1)$ rounds are necessary and sufficient to sort in the worst case. For an $m_3 \times m_2 \times m_1$-mesh, m_i arbitrary, $\log m_3 + \log m_2 + O(\frac{\log m_3}{\log m_2})$ rounds are sufficient.

NC-Algorithms for Operations on Binary Decision Diagrams

by INGO WEGENER (joint work with Detlef Sieling)

Ordered binary decision diagrams are the state of the art data structure for Boolean functions. They are applied for symbolic verification, model checking, test pattern generation, logical synthesis and many other problems. NC-algorithms are presented for the typical operations on ordered binary decision diagrams like synthesis, reduction, satisfiability, equality test or replacement. The number of processors used is for some operations quite large due to the well-known transitive closure bottleneck.

Optimal Tree Contraction on the Hypercube and Related Networks

by RALPH WERCHNER (joint work with Ernst Mayr)

An optimal tree contraction algorithm for the boolean hypercube and the constant degree hypercubic networks, such as the shuffle exchange or the butterfly network, is presented. For trees of size n, stored on a p processor hypercube in in-order, the running time of the algorithm is $O(\lceil \frac{n}{p} \rceil \log p)$. The resulting speed-up of $\Theta(p/\log p)$ is optimal due to logarithmic communication overhead, as shown by a corresponding lower bound.

The algorithm consists of three phases: First, maximal subtrees stored in subcubes of size $p^{3/4}$ are identified and contracted in a recursive call. In the resulting tree, the remaining "chains" are contracted by a segmented parallel prefix operation, prepared by a suitable reorganization of the chains employing a novel routing technique. Finally, a standard PRAM tree contraction algorithm is simulated with only constant slow-down, which can be achieved after rearranging the nodes in the cube according to the communication structure of the PRAM algorithm.

Tree contraction can be used to evaluate algebraic expressions consisting of operators $+, -, \cdot, /$ and rational operands, as well as for the membership problem for certain subclasses of languages in DCFL.

E-Mail Addresses

Greg Barnes	barnes@mpi-sb.mpg.de
Sandeep Bhatt	bhatt@bellcore.com
Gianfranco Bilardi	bilardi@dei.unipd.it
Richard Cole	cole@cs.nyu.edu
Artur Czumaj	artur@hni.uni-paderborn.de
Torben Hagerup	torben@mpi-sb.mpg.de
Juraj Hromkovič	juraj@uni-paderborn.de
Marek Karpiński	marek@cs.uni-bonn.de
Lefteris Kirousis	kirousis@grpatvx1.bitnet
Manfred Kunde	kunde@informatik.tu-muenchen.de
Mirosław Kutyłowski	mirekk@uni-paderborn.de
Klaus-Jörn Lange	lange@informatik.tu-muenchen.de
Reinhard Lüling	rl@uni-paderborn.de
Bruce Maggs	bruce@research.nj.nec.com
Ernst W. Mayr	mayr@informatik.tu-muenchen.de
Friedhelm Meyer auf der Heide	fmadh@uni-paderborn.de
Naomi Nishimura	nishi@maytag.uwaterloo.ca
Prabhakar Ragde	plragde@maytag.uwaterloo.ca
Wojciech Rytter	rytter@mimuw.edu.pl
Alan Siegel	siegel@cs.nyu.edu
Paul Spirakis	spirakis@grpatvx1.bitnet
Prasoon Tiwari	tiwari@cs.wisc.edu
Rolf Wanka	wanka@uni-paderborn.de
Ingo Wegener	wegener@cantor.informatik.uni-dortmund.de
Ralph Werchner	werchner@thi.informatik.uni-frankfurt.de

Addresses

Greg Barnes
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
Germany
☎ +49-681-302-5442

Sandeep Bhatt
Bell Communications Research
445 South Street
Morristown NJ 07960
USA
☎ +1-201-829-4319

Gianfranco Bilardi
Universitá di Padova
Dipartimento di Elettronica ed Informatica
Via Gradenigo 6/A
I-35131 Padova
Italy
☎ +39-49-8287653

Richard Cole
New York University
Courant Institute of
Mathematical Sciences
251 Mercer Street
New York NY 10012-1085
USA
☎ +1-212-998-3119

Artur Czumaj
Universität Paderborn
Heinz-Nixdorf-Institut
33095 Paderborn
Germany

Torben Hagerup
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
Germany
☎ +49-681-302-5358

Juraj Hromkovič
University of Paderborn
FB 17 Mathematik-Informatik
33095 Paderborn
Germany
☎ +49-5251-60-3344

Marek Karpinski
Institut für Informatik; Abt. V
Universität Bonn
Römerstraße 164
53117 Bonn
Germany
☎ +49-228-550-327 (Secr.: 224)

Lefteris Kirousis
University of Patras
Department of Computer
Engineering und Informatics
Rio
26500 Patras
Greece
☎ +30-61-99 75 85

Manfred Kunde
TU München
Institut für Informatik
Arcisstraße 21
80290 München
Germany
☎ +49-89-2105-2390

Mirosław Kutyłowski
Universytet Wrocławski
Instytut Informatyki
ul. Przesmyckiego 20
PL-51-151 Wrocław
Poland
☎ +48-71-57-59-64 (private)

Reinhard Lüling
Universität GH Paderborn
FB 17 - Mathematik/Informatik
D-33095 Paderborn
Germany
☎ +49-5251-60-3331

Klaus-Jörn Lange
TU München
Institut für Informatik
Arcisstraße 21
80290 München
Germany
☎ +49-89-2105-2403 (Secr.: 2404)

Bruce Maggs
NEC Research Institute
4 Independence Way
Princeton NJ 08540
USA
☎ +1-609-951-2736

Ernst W. Mayr
TU München
Institut für Informatik
80290 München
Germany
☎ +49-89-2105-2680 /-2681

Friedhelm Meyer auf der Heide
Universität GH Paderborn
FB 17 - Mathematik/Informatik
und Heinz Nixdorf Institut
33095 Paderborn
Germany
☎ +49-5251-60 3310

Naomi Nishimura
University of Waterloo
Department of Computer Science
Waterloo Ontario N2L 3G1
Canada
☎ +1-519-888-4835

Prabhakar Ragde
University of Waterloo
Department of Computer Science
Waterloo Ontario N2L 3G1
Canada
☎ +1-519-888-4660

Wojciech Rytter
University of Warsaw
Dept. of Mathematics & Informatics
Ul. Banacha 2
02-097 Warsaw 59
Poland
☎ +48-2-6583165

Alan Siegel
New York University
Courant Institute of
Mathematical Sciences
251 Mercer Street
New York NY 10012
USA
☎ +1-212-998-3122

Paul Spirakis
University of Patras
Dept. of Computer Engineering & CTI
PO Box 1122
26110 Patras
Greece
☎ +30-61-991650

Prasoon Tiwari
University of Wisconsin-Madison
Department of Computer Sciences
1210 W. Dayton Street
Madison WI 53706
USA

Rolf Wanka
Universität GH Paderborn
FB 17 - Mathematik/Informatik und
Heinz Nixdorf Institut
33095 Paderborn
Germany
☎ +49-5251-60-3309

Ingo Wegener
Universität Dortmund
Fachbereich Informatik LS II
August-Schmidt-Str. 12
44221 Dortmund
Germany
☎ +49-231-755-2776

Ralph Werchner
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
60054 Frankfurt
Germany
☎ +49-69-798-8324

Zuletzt erschienene und geplante Titel:

C.A. Ellis, M. Jarke (editors):
Distributed Cooperation in Integrated Information Systems; Dagstuhl-Seminar-Report; 38; 5.4.-9.4.92 (9215)

J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 39; 22.06.-26.06.92 (9226)

E. Börger, Y. Gurevich, H. Kleine-Büning, M.M. Richter (editors):
Computer Science Logic, Dagstuhl-Seminar-Report; 40; 13.07.-17.07.92 (9229)

J. von zur Gathen, M. Karpinski, D. Kozen (editors):
Algebraic Complexity and Parallelism, Dagstuhl-Seminar-Report; 41; 20.07.-24.07.92 (9230)

F. Baader, J. Siekmann, W. Snyder (editors):
6th International Workshop on Unification, Dagstuhl-Seminar-Report; 42; 29.07.-31.07.92 (9231)

J.W. Davenport, F. Krückeberg, R.E. Moore, S. Rump (editors):
Symbolic, algebraic and validated numerical Computation, Dagstuhl-Seminar-Report; 43; 03.08.-07.08.92 (9232)

R. Cohen, R. Kass, C. Paris, W. Wahlster (editors):
Third International Workshop on User Modeling (UM'92), Dagstuhl-Seminar-Report; 44; 10.-13.8.92 (9233)

R. Reischuk, D. Uhlig (editors):
Complexity and Realization of Boolean Functions, Dagstuhl-Seminar-Report; 45; 24.08.-28.08.92 (9235)

Th. Lengauer, D. Schomburg, M.S. Waterman (editors):
Molecular Bioinformatics, Dagstuhl-Seminar-Report; 46; 07.09.-11.09.92 (9237)

V.R. Basili, H.D. Rombach, R.W. Selby (editors):
Experimental Software Engineering Issues, Dagstuhl-Seminar-Report; 47; 14.-18.09.92 (9238)

Y. Dittrich, H. Hastedt, P. Schefe (editors):
Computer Science and Philosophy, Dagstuhl-Seminar-Report; 48; 21.09.-25.09.92 (9239)

R.P. Daley, U. Furbach, K.P. Jantke (editors):
Analogical and Inductive Inference 1992 , Dagstuhl-Seminar-Report; 49; 05.10.-09.10.92 (9241)

E. Novak, St. Smale, J.F. Traub (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 50; 12.10.-16.10.92 (9242)

J. Encarnação, J. Foley (editors):
Multimedia - System Architectures and Applications, Dagstuhl-Seminar-Report; 51; 02.11.-06.11.92 (9245)

F.J. Rammig, J. Staunstrup, G. Zimmermann (editors):
Self-Timed Design, Dagstuhl-Seminar-Report; 52; 30.11.-04.12.92 (9249)

B. Courcelle, H. Ehrig, G. Rozenberg, H.J. Schneider (editors):
Graph-Transformations in Computer Science, Dagstuhl-Seminar-Report; 53; 04.01.-08.01.93 (9301)

A. Arnold, L. Priese, R. Vollmar (editors):
Automata Theory: Distributed Models, Dagstuhl-Seminar-Report; 54; 11.01.-15.01.93 (9302)

W. Cellary, K. Vidyasankar , G. Vossen (editors):
Versioning in Database Management Systems, Dagstuhl-Seminar-Report; 55; 01.02.-05.02.93 (9305)

B. Becker, R. Bryant, Ch. Meinel (editors):
Computer Aided Design and Test , Dagstuhl-Seminar-Report; 56; 15.02.-19.02.93 (9307)

M. Pinkal, R. Scha, L. Schubert (editors):
Semantic Formalisms in Natural Language Processing, Dagstuhl-Seminar-Report; 57; 23.02.-26.02.93 (9308)

W. Bibel, K. Furukawa, M. Stickel (editors):
Deduction , Dagstuhl-Seminar-Report; 58; 08.03.-12.03.93 (9310)

H. Alt, B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 59; 22.03.-26.03.93 (9312)

H. Kamp, J. Pustejovsky (editors):
Universals in the Lexicon: At the Intersection of Lexical Semantic Theories, Dagstuhl-Seminar-Report; 60; 29.03.-02.04.93 (9313)

W. Strasser, F. Wahl (editors):
Graphics & Robotics, Dagstuhl-Seminar-Report; 61; 19.04.-22.04.93 (9316)

C. Beeri, A. Heuer, G. Saake, S. Urban (editors):
Formal Aspects of Object Base Dynamics , Dagstuhl-Seminar-Report; 62; 26.04.-30.04.93 (9317)

R. V. Book, E. Pednault, D. Wotschke (editors):
Descriptional Complexity, Dagstuhl-Seminar-Report; 63; 03.05.-07.05.93 (9318)

H.-D. Ehrig, F. von Henke, J. Meseguer, M. Wirsing (editors):
Specification and Semantics, Dagstuhl-Seminar-Report; 64; 24.05.-28.05.93 (9321)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Algebra, Dagstuhl-Seminar-Report; 65; 07.06.-11.06.93 (9323)

Ch. Lengauer, P. Quinton, Y. Robert, L. Thiele (editors):
Parallelization Techniques for Uniform Algorithms, Dagstuhl-Seminar-Report; 66; 21.06.-25.06.93 (9325)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 67; 28.06.-02.07.93 (9326)

Ph. Flajolet, R. Kemp, H. Prodinger (editors):
"Average-Case"-Analysis of Algorithms, Dagstuhl-Seminar-Report; 68; 12.07.-16.07.93 (9328)

J.W. Gray, A.M. Pitts, K. Sieber (editors):
Interactions between Category Theory and Computer Science, Dagstuhl-Seminar-Report; 69; 19.07.-23.07.93 (9329)

D. Gabbay, H.-J. Ohlbach (editors):
Automated Practical Reasoning and Argumentation, Dagstuhl-Seminar-Report; 70; 23.08.-27.08.93 (9334)

A. Danthine , W. Effelsberg, O. Spaniol, (editors):
Architecture and Protocols for High-Speed Networks, Dagstuhl-Seminar-Report; 71; 30.08.-03.09.93 (9335)

R. Cole, E. W. Mayr, F. Meyer a.d.Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 72; 13.09.-17.09.93 (9337)

V. Marek, A. Nerode, P.H. Schmitt (editors):
Non-Classical Logics in Computer Science, Dagstuhl-Seminar-Report; 73; 20.-24.09.93 (9338)

A. Odlyzko, C.P. Schnorr, A. Shamir (editors):
Cryptography, Dagstuhl-Seminar-Report; 74; 27.09.-01.10.93 (9339)

J. Angeles, G. Hommel, P. Kovács (editors):
Computational Kinematics, Dagstuhl-Seminar-Report; 75; 11.10.-15.10.93 (9341)

T. Lengauer, M. Sarrafzadeh, D. Wagner (editors):
Combinatorial Methods for Integrated Circuit Design, Dagstuhl-Seminar-Report; 76; 18.10.-22.10.93 (9342)