LNCS 3111

Torben Hagerup
Jyrki Katajainen (Eds.)

Algorithm Theory – SWAT 2004

9th Scandinavian Workshop on Algorithm Theory
Humlebæk, Denmark, July 2004
Proceedings

Springer

Torben Hagerup   Jyrki Katajainen (Eds.)

Algorithm Theory – SWAT 2004

9th Scandinavian Workshop on Algorithm Theory
Humlebæk, Denmark, July 8-10, 2004
Proceedings

Springer

Volume Editors

Torben Hagerup
Universität Augsburg, Institut für Informatik
86135 Augsburg, Germany
E-mail: hagerup@informatik.uni-augsburg.de

Jyrki Katajainen
University of Copenhagen, Department of Computing
Universitetsparken 1, 2100 Copenhagen East, Denmark
E-mail: jyrki@diku.dk

Library of Congress Control Number: 2004107864

CR Subject Classification (1998): F.2, E.1, G.2, I.3.5, C.2

ISSN 0302-9743
ISBN 3-540-22339-8 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11014997 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3111

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Lecture Notes in Computer Science

For information about Vols. 1–3022

please contact your bookseller or Springer-Verlag

Vol. 3125: D. Kozen (Ed.), Mathematics of Program Construction. X, 401 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004. (Subseries LNAI).

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004. (Subseries LNAI).

Vol. 3118: K. Miesenberger, J. Klaus, W. Zagler, D. Burger (Eds.), Computer Helping People with Special Needs. XXIII, 1191 pages. 2004.

Vol. 3116: C. Rattray, S. Maharaj, C. Shankland (Eds.), Algebraic Methodology and Software Technology. XI, 569 pages. 2004.

Vol. 3114: R. Alur, D.A. Peled (Eds.), Computer Aided Verification. XII, 536 pages. 2004.

Vol. 3113: J. Karhumäki, H. Maurer, G. Paun, G. Rozenberg (Eds.), Theory Is Forever. X, 283 pages. 2004.

Vol. 3112: H. Williams, L. MacKinnon (Eds.), New Horizons in Information Management. XII, 265 pages. 2004.

Vol. 3111: T. Hagerup, J. Katajainen (Eds.), Algorithm Theory – SWAT 2004. XI, 506 pages. 2004.

Vol. 3109: S.C. Sahinalp, S. Muthukrishnan, U. Dogrusoz (Eds.), Combinatorial Pattern Matching. XII, 486 pages. 2004.

Vol. 3107: J. Bosch, C. Krüger (Eds.), Software Reuse: Methods, Techniques and Tools. XI, 339 pages. 2004.

Vol. 3105: S. Göbel, U. Spierling, A. Hoffmann, I. Iurgel, O. Schneider, J. Dechau, A. Feix (Eds.), Technologies for Interactive Digital Storytelling and Entertainment. XVI, 304 pages. 2004.

Vol. 3104: R. Kralovic, O. Sykora (Eds.), Structural Information and Communication Complexity. X, 303 pages. 2004.

Vol. 3103: K. Deb (Ed.), Genetic and Evolutionary Computation - GECCO 2004. XLIX, 1439 pages. 2004.

Vol. 3102: K. Deb (Ed.), Genetic and Evolutionary Computation - GECCO 2004. L, 1445 pages. 2004.

Vol. 3101: M. Masoodian, S. Jones, B. Rogers (Eds.), Computer Human Interaction. XIV, 694 pages. 2004.

Vol. 3099: J. Cortadella, W. Reisig (Eds.), Applications and Theory of Petri Nets 2004. XI, 505 pages. 2004.

Vol. 3098: J. Desel, W. Reisig, G. Rozenberg (Eds.), Lectures on Concurrency and Petri Nets. VIII, 849 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004. (Subseries LNAI).

Vol. 3096: G. Melnik, H. Holz (Eds.), Advances in Learning Software Organizations. X, 173 pages. 2004.

Vol. 3094: A. Nürnberger, M. Detyniecki (Eds.), Adaptive Multimedia Retrieval. VIII, 229 pages. 2004.

Vol. 3093: S.K. Katsikas, S. Gritzalis, J. Lopez (Eds.), Public Key Infrastructure. XIII, 380 pages. 2004.

Vol. 3092: J. Eckstein, H. Baumeister (Eds.), Extreme Programming and Agile Processes in Software Engineering. XVI, 358 pages. 2004.

Vol. 3091: V. van Oostrom (Ed.), Rewriting Techniques and Applications. X, 313 pages. 2004.

Vol. 3089: M. Jakobsson, M. Yung, J. Zhou (Eds.), Applied Cryptography and Network Security. XIV, 510 pages. 2004.

Vol. 3086: M. Odersky (Ed.), ECOOP 2004 – Object-Oriented Programming. XIII, 611 pages. 2004.

Vol. 3085: S. Berardi, M. Coppo, F. Damiani (Eds.), Types for Proofs and Programs. X, 409 pages. 2004.

Vol. 3084: A. Persson, J. Stirna (Eds.), Advanced Information Systems Engineering. XIV, 596 pages. 2004.

Vol. 3083: W. Emmerich, A.L. Wolf (Eds.), Component Deployment. X, 249 pages. 2004.

Vol. 3080: J. Desel, B. Pernici, M. Weske (Eds.), Business Process Management. X, 307 pages. 2004.

Vol. 3079: Z. Mammeri, P. Lorenz (Eds.), High Speed Networks and Multimedia Communications. XVIII, 1103 pages. 2004.

Vol. 3078: S. Cotin, D.N. Metaxas (Eds.), Medical Simulation. XVI, 296 pages. 2004.

Vol. 3077: F. Roli, J. Kittler, T. Windeatt (Eds.), Multiple Classifier Systems. XII, 386 pages. 2004.

Vol. 3076: D. Buell (Ed.), Algorithmic Number Theory. XI, 451 pages. 2004.

Vol. 3074: B. Kuijpers, P. Revesz (Eds.), Constraint Databases and Applications. XII, 181 pages. 2004.

Vol. 3073: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), Intelligence and Security Informatics. XV, 536 pages. 2004.

Vol. 3072: D. Zhang, A.K. Jain (Eds.), Biometric Authentication. XVII, 800 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004. (Subseries LNAI).

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004. (Subseries LNAI).

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004. (Subseries LNAI).

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004. (Subseries LNAI).

Vol. 3066: S. Tsumoto, R. S lowiński, J. Komorowski, J.W. Grzymala-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004. (Subseries LNAI).

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004. (Subseries LNAI).

Vol. 3064: D. Bienstock, G. Nemhauser (Eds.), Integer Programming and Combinatorial Optimization. XI, 445 pages. 2004.

Vol. 3063: A. Llamosí, A. Strohmeier (Eds.), Reliable Software Technologies - Ada-Europe 2004. XIII, 333 pages. 2004.

Vol. 3062: J.L. Pfaltz, M. Nagl, B. Böhlen (Eds.), Applications of Graph Transformations with Industrial Relevance. XV, 500 pages. 2004.

Vol. 3061: F.F. Ramas, H. Unger, V. Larios (Eds.), Advanced Distributed Systems. VIII, 285 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004. (Subseries LNAI).

Vol. 3059: C.C. Ribeiro, S.L. Martins (Eds.), Experimental and Efficient Algorithms. X, 586 pages. 2004.

Vol. 3058: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 233 pages. 2004.

Vol. 3057: B. Jayaraman (Ed.), Practical Aspects of Declarative Languages. VIII, 255 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004. (Subseries LNAI).

Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreasen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 500 pages. 2004. (Subseries LNAI).

Vol. 3054: I. Crnkovic, J.A. Stafford, H.W. Schmidt, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 311 pages. 2004.

Vol. 3053: C. Bussler, J. Davies, D. Fensel, R. Studer (Eds.), The Semantic Web: Research and Applications. XIII, 490 pages. 2004.

Vol. 3052: W. Zimmermann, B. Thalheim (Eds.), Abstract State Machines 2004. Advances in Theory and Practice. XII, 235 pages. 2004.

Vol. 3051: R. Berghammer, B. Möller, G. Struth (Eds.), Relational and Kleene-Algebraic Methods in Computer Science. X, 279 pages. 2004.

Vol. 3050: J. Domingo-Ferrer, V. Torra (Eds.), Privacy in Statistical Databases. IX, 367 pages. 2004.

Vol. 3049: M. Bruynooghe, K.-K. Lau (Eds.), Program Development in Computational Logic. VIII, 539 pages. 2004.

Vol. 3047: F. Oquendo, B. Warboys, R. Morrison (Eds.), Software Architecture. X, 279 pages. 2004.

Vol. 3046: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1016 pages. 2004.

Vol. 3045: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1040 pages. 2004.

Vol. 3044: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1140 pages. 2004.

Vol. 3043: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1180 pages. 2004.

Vol. 3042: N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, L. Merakos (Eds.), NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. XXXIII, 1519 pages. 2004.

Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), Current Topics in Artificial Intelligence. XIV, 689 pages. 2004. (Subseries LNAI).

Vol. 3039: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 1271 pages. 2004.

Vol. 3038: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 1311 pages. 2004.

Vol. 3037: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 745 pages. 2004.

Vol. 3036: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 713 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004. (Subseries LNAI).

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004. (Subseries LNAI).

Vol. 3033: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVIII, 1076 pages. 2004.

Vol. 3032: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVII, 1112 pages. 2004.

Vol. 3031: A. Butz, A. Krüger, P. Olivier (Eds.), Smart Graphics. X, 165 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004. (Subseries LNAI).

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004. (Subseries LNAI).

Vol. 3028: D. Neuenschwander, Probabilistic and Statistical Methods in Cryptology. X, 158 pages. 2004.

Vol. 3027: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004. XI, 628 pages. 2004.

Vol. 3026: C.V. Ramamoorthy, R. Lee, K.W. Lee (Eds.), Software Engineering Research and Applications. XV, 377 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004. (Subseries LNAI).

Vol. 3024: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3023: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 611 pages. 2004.

Preface

This volume contains the papers presented at SWAT 2004, the 9th Scandinavian Workshop on Algorithm Theory, which was held on July 8–10, 2004, at the Louisiana Museum of Modern Art in Humlebæk on the Øresund coast north of Copenhagen. The SWAT workshop, in reality a full-fledged conference, has been held biennially since 1988 and rotates among the five Nordic countries, Denmark, Finland, Iceland, Norway, and Sweden. The previous meetings took place in Halmstad (1988), Bergen (1990), Helsinki (1992), Århus (1994), Reykjavik (1996), Stockholm (1998), Bergen (2000), and Turku (2002). SWAT alternates with the Workshop on Algorithms and Data Structures (WADS), held in odd-numbered years.

The call for papers invited contributions on all aspects of algorithm theory. A total of 121 submissions was received — an overall SWAT high. These underwent thorough reviewing, and the program committee met in Copenhagen on March 20–21, 2004, and selected 40 papers for presentation at the conference. The program committee was impressed with the quality of the submissions and, given the constraints imposed by the choice of conference venue and duration, had to make some tough decisions. The scientific program was enriched by invited presentations by Gerth Stølting Brodal (University of Aarhus) and Charles E. Leiserson (Massachusetts Institute of Technology).

Two satellite events were held immediately before SWAT 2004: the Workshop on On-Line Algorithms (OLA 2004), organized by members of the Department of Mathematics and Computer Science at the University of Southern Denmark, and the Summer School on Experimental Algorithmics, organized by the Performance Engineering Laboratory in the Department of Computing at the University of Copenhagen. More information about SWAT 2004 and its satellite events is available at the conference web site `http://swat.diku.dk/`.

We thank all of the many persons whose efforts contributed to making SWAT 2004 a success. These include the members of the steering, program, and organizing committees, the invited speakers, the authors who submitted papers, the numerous referees who assisted the program committee, and the support staff at the conference. Our special thanks go to Frank Kammer (University of Augsburg), who maintained the submission server, and Sebastian Berg and his associates (Less is more ApS), who implemented and hosted the web-based conference shop. We also thank our institutional and industrial sponsors for their support.

May 2004

Torben Hagerup
Jyrki Katajainen

SWAT Steering Committee

Svante Carlsson, Blekinge Institute of Technology
Magnús M. Halldórsson, University of Iceland
Rolf Karlsson, Lund University
Andrzej Lingas, Lund University
Erik Meineche Schmidt, University of Aarhus
Jan Arne Telle, University of Bergen
Esko Ukkonen, University of Helsinki

SWAT 2004 Conference Chair

Jyrki Katajainen, University of Copenhagen

SWAT 2004 Program Committee

Arne Andersson, Uppsala University
Therese Biedl, University of Waterloo
Otfried Cheong, TU Eindhoven
Artur Czumaj, New Jersey Institute of Technology
Camil Demetrescu, University of Rome "La Sapienza"
Amos Fiat, Tel Aviv University
Gudmund Skovbjerg Frandsen, University of Aarhus
Torben Hagerup, University of Augsburg (chair)
Magnús M. Halldórsson, University of Iceland
Jyrki Katajainen, University of Copenhagen
Kim Skak Larsen, University of Southern Denmark
Rolf Niedermeier, University of Tübingen
Marina Papatriantafilou, Chalmers University of Technology
Cynthia A. Phillips, Sandia National Laboratories
Jan Arne Telle, University of Bergen
Jukka Teuhola, University of Turku

SWAT 2004 Organizing Committee

Joan Boyar, University of Southern Denmark
Christopher Derek Curry, University of Copenhagen
Lene Monrad Favrholdt, University of Southern Denmark
Karin Outzen, University of Copenhagen
Anna Pagh, IT University of Copenhagen
Rasmus Pagh, IT University of Copenhagen

Referees

Geir Agnarsson
Lars Arge
Boris Aronov
Yossi Azar
Jørgen Bang-Jensen
Luca Becchetti
Mark de Berg
Jorma Boberg
Joan Boyar
Broňa Brejová
Gerth Stølting Brodal
Dan Brown
David Bunde
Jesper Makholm Byskov
Tiziana Calamoneri
Robert Carr
Lenka Carr Motyckova
Barun Chandra
Benny Chor
Graham Cormode
Peter Damaschke
Erik Demaine
Uma Devi
Frederic Dorn
Leah Epstein
Rolf Fagerberg
Lene Monrad Favrholdt
Henning Fernau
Irene Finocchi
Fedor Fomin
Pasi Fränti
Toshihiro Fujito
Rajeev Gandhi
Naveen Garg
Burkay Genc
Anders Gidenstam
Jens Gramm
Roberto Grossi
Joachim Gudmundsson
Jiong Guo
Phuong Hoai Ha
Hjálmtýr Hafsteinsson
Harri Hakonen
Bjarni Vilhjálmur Halldórsson
Danny Halperin
Tero Harju
Rafi Hassin
Herman Haverkort
Pinar Heggernes
Peter Höfner
Falk Hüffner
Jouni Järvinen
Claus Jensen
Marek Karpinski
Michael Kaufmann
Chang Sik Kim
Ralf Klasing
Timo Knuutila
Jens Svalgaard Kohrt
Boris Koldehofe
Jochen Könemann
Guy Kortsarz
Evangelos Kranakis
Marc van Kreveld
Danny Krizanc
Piotr Krysta
Risto Lahdelma
Stefano Leonardi
Alex López-Ortiz
Anna Lubiw
Bolette Ammitzbøll Madsen
Fredrik Manne
Peter Bro Miltersen
Shuichi Miyazaki
Olli Nevalainen
Tomi Armas Pasanen
Maurizio Patrignani
Christian Nørgaard Storm Pedersen
Seth Pettie
Derek Philips
David Pisinger
Giuseppe Prencipe
Francesco Quaglia
Danny Raz
Liam Roditty
Frederik Rønn
Guenter Rote
Wojciech Rytter
Christian Scheideler
Hadas Shachnai
Micha Sharir
Riccardo Silvestri
Sivaramakrishnan Sivasubramanian
Bjarke Skjernaa
Christian Sloper
Michiel Smid
Christian Sohler
Bettina Speckmann
Yannis Stamatiou
Håkan Sundell
Arie Tamir
Dimitrios Thilikos
Bjarne Toft
Takeshi Tokuyama
Philippas Tsigas
Antoine Vigneron
Tomáš Vinař
Xin Wang
Sebastian Wernicke
Sue Whitesides
Hezy Yeshurun
Martin Zachariasen
Christos Zaroliagis
Yi Zhang
Hairong Zhao
Uri Zwick

Sponsors

DIKU

University of Southern Denmark

Department of Mathematics and Computer Science

IT University of Copenhagen

The Danish Natural Science Research Council

<FIRST> GRADUATE SCHOOL

LESS IS MORE

NOKIA Connecting People

SAS

Table of Contents

Design and Analysis of Dynamic Multithreaded Algorithms

Charles E. Leiserson*

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, Massachusetts 02139, USA

Dynamic multithreaded languages provide low-overhead fork-join primitives to express parallelism. One such language is Cilk [3, 5], which was developed in the MIT Laboratory for Computer Science (now the MIT Computer Science and Artificial Intelligence Laboratory). Cilk minimally extends the C programming language to allow interactions among computational threads to be specified in a simple and high-level fashion. Cilk's provably efficient runtime system dynamically maps a user's program onto available physical resources using a randomized "work-stealing" scheduler, freeing the programmer from concerns of communication protocols and load balancing.

Cilk provides an abstract performance model for algorithmic analysis. This model characterizes the performance of a multithreaded algorithm in terms of two quantities: its ***work*** T_1, which is the total time needed to execute the computation serially, and its ***critical-path length*** T_∞, which is its execution time on an infinite number of processors. Cilk provides instrumentation that allows a user to measure these two quantities. Cilk's scheduler executes a Cilk computation on P processors in expected time

$$T_P = T_1/P + O(T_\infty) \ ,$$

assuming an ideal parallel computer. This equation resembles "Brent's theorem" [1, 4] and is optimal to within a constant factor, since T_1/P and T_∞ are both lower bounds.

The fork-join primitives of a dynamic multithreaded language encourage the design of divide-and-conquer parallel algorithms. Consequently, recurrences are a natural way to express the work and critical-path length of a multithreaded algorithm. Recurrence analysis can be used to develop good algorithms for matrix multiplication, sorting, and a host of other problems.

The algorithmic technology behind dynamic multithreading has been applied to develop MIT's championship computer-chess programs, ⋆Socrates and

* Support for this research was provided in part by the Defense Advanced Research Projects Agency (DARPA) under Grant F30602-97-1-0270, by the National Science Foundation under Grants EIA-9975036 and ACI-032497, and by the Singapore-MIT Alliance.

T. Hagerup and J. Katajainen (Eds.): SWAT 2004, LNCS 3111, pp. 1–2, 2004.

Cilkchess [2]. The methodology of analyzing work and critical-path length provides a clean theoretical model which works in practice.

See `http://supertech.csail.mit.edu/cilk` for more background on Cilk and to download the Cilk manual and software release.

References

1. Richard P. Brent. The parallel evaluation of general arithmetic expressions. *Journal of the ACM*, 21(2):201–206, April 1974.
2. Don Dailey and Charles E. Leiserson. Using Cilk to write multiprocessor chess programs. *Advances in Computer Games*, 2001.
3. Matteo Frigo, Charles E. Leiserson, and Keith H. Randall. The implementation of the Cilk-5 multithreaded language. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 212–223, 1998.
4. R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, March 1969.
5. Supercomputing Technologies Group, MIT Computer Science and Artificial Intelligence Laboratory. *Cilk 5.3.2 Reference Manual*, November 2001.

Cache-Oblivious Algorithms and Data Structures

Gerth Stølting Brodal*

BRICS**, Department of Computer Science, University of Aarhus
IT-parken, Åbogade 34, DK-8200 Århus N, Denmark
gerth@daimi.au.dk

Abstract. Frigo, Leiserson, Prokop and Ramachandran in 1999 introduced the ideal-cache model as a formal model of computation for developing algorithms in environments with multiple levels of caching, and coined the terminology of *cache-oblivious algorithms*. Cache-oblivious algorithms are described as standard RAM algorithms with only one memory level, i.e. without any knowledge about memory hierarchies, but are analyzed in the two-level I/O model of Aggarwal and Vitter for an arbitrary memory and block size and an optimal off-line cache replacement strategy. The result are algorithms that automatically apply to multi-level memory hierarchies. This paper gives an overview of the results achieved on cache-oblivious algorithms and data structures since the seminal paper by Frigo *et al.*

1 Introduction

Modern computers are characterized by having a memory system consisting of a hierarchy of several levels of memory, where each level is acting as a cache for the next level [46]. The typical memory levels of current machines are registers, level 1 cache, level 2 cache, level 3 cache, main memory, and disk. While the sizes of the levels increase with the distance from the CPU the access times to the levels also get larger, most dramatically when going from main memory to disk. To circumvent dramatic performance loss data is moved between the memory levels in blocks (cache lines or disk blocks). As a consequence of this organization of the memory, the memory access pattern of an algorithm has a major influence on its practical running time. A basic rule commonly stated in the literature for achieving good running times is to ensure *locality of reference* in the developed algorithms.

1.1 I/O Model

Several models have been proposed in recent years to model modern memory hierarchies. The most successful of these models (in terms of number of publications) is the two-level I/O model introduced by Aggarwal and Vitter in 1988 [6]:

* Supported by the Carlsberg Foundation (contract number ANS-0257/20).
** Basic Research in Computer Science, www.brics.dk, funded by the Danish National Research Foundation.

T. Hagerup and J. Katajainen (Eds.): SWAT 2004, LNCS 3111, pp. 3–13, 2004.