Lynn Choi
Yunheung Paek
Sangyeun Cho (Eds.)

Advances in Computer Systems Architecture

LNCS 4697

12th Asia-Pacific Conference, ACSAC 2007
Seoul, Korea, August 2007
Proceedings

Springer

Lynn Choi Yunheung Paek
Sangyeun Cho (Eds.)

Advances in Computer Systems Architecture

12th Asia-Pacific Conference, ACSAC 2007
Seoul, Korea, August 23-25, 2007
Proceedings

Springer

Volume Editors

Lynn Choi
Korea University
School of Electrical Engineering
Anam-Dong, Sungbuk-Ku, Seoul, Korea
E-mail: lchoi@korea.ac.kr

Yunheung Paek
Seoul National University
School of Electrical Engineering
Seoul, Korea
E-mail: ypaek@snu.ac.kr

Sangyeun Cho
University of Pittsburgh
Department of Computer Science
Pittsburgh, PA 15260, USA
E-mail: cho@cs.pitt.edu

Library of Congress Control Number: 2007932678

CR Subject Classification (1998): B.2, B.4, B.5, C.2, C.1, D.4

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-74308-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-74308-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12109494 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4697

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–4549

please contact your bookseller or Springer

Vol. 4697: L. Choi, Y. Paek, S. Cho (Eds.), Advances in Computer Systems Architecture. XIII, 400 pages. 2007.

Vol. 4682: D.-S. Huang, L. Heutte, M. Loog (Eds.), Advanced Intelligent Computing Theories and Applications. XXVII, 1373 pages. 2007. (Sublibrary LNAI).

Vol. 4671: V. Malyshkin (Ed.), Parallel Computing Technologies. XIV, 635 pages. 2007.

Vol. 4660: S. Džeroski, J. Todorovski (Eds.), Computational Discovery of Scientific Knowledge. X, 327 pages. 2007. (Sublibrary LNAI).

Vol. 4651: F. Azevedo, P. Barahona, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. VIII, 185 pages. 2007. (Sublibrary LNAI).

Vol. 4647: R. Martin, M. Sabin, J. Winkler (Eds.), Mathematics of Surfaces XII. IX, 509 pages. 2007.

Vol. 4645: R. Giancarlo, S. Hannenhalli (Eds.), Algorithms in Bioinformatics. XIII, 432 pages. 2007. (Sublibrary LNBI).

Vol. 4643: M.-F. Sagot, M.E.M.T. Walter (Eds.), Advances in Bioinformatics and Computational Biology. XII, 177 pages. 2007. (Sublibrary LNBI).

Vol. 4639: E. Csuhaj-Varj'u, Z. 'Esik (Eds.), Fundamentals of Computation Theory. XIV, 508 pages. 2007.

Vol. 4634: H.R. Nielson, G. Filé (Eds.), Static Analysis. XI, 469 pages. 2007.

Vol. 4632: R. Alhajj, H. Gao, X. Li, J. Li, O.R. Zaïane (Eds.), Advanced Data Mining and Applications. XV, 634 pages. 2007. (Sublibrary LNAI).

Vol. 4628: L.N. de Castro, F.J. Von Zuben, H. Knidel (Eds.), Artificial Immune Systems. XII, 438 pages. 2007.

Vol. 4627: M. Charikar, K. Jansen, O. Reingold, J.D.P. Rolim (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 626 pages. 2007.

Vol. 4626: R.O. Weber, M.M. Richter (Eds.), Case-Based Reasoning Research and Development. XIII, 534 pages. 2007. (Sublibrary LNAI).

Vol. 4624: T. Mossakowski, U. Montanari, M. Haveraaen (Eds.), Algebra and Coalgebra in Computer Science. XI, 463 pages. 2007.

Vol. 4622: A. Menezes (Ed.), Advances in Cryptology - CRYPTO 2007. XIV, 631 pages. 2007.

Vol. 4619: F. Dehne, J.-R. Sack, N. Zeh (Eds.), Algorithms and Data Structures. XVI, 662 pages. 2007.

Vol. 4618: S.G. Akl, C.S. Calude, M.J. Dinneen, G. Rozenberg, H.T. Wareham (Eds.), Unconventional Computation. X, 243 pages. 2007.

Vol. 4617: V. Torra, Y. Narukawa, Y. Yoshida (Eds.), Modeling Decisions for Artificial Intelligence. XII, 502 pages. 2007. (Sublibrary LNAI).

Vol. 4616: A. Dress, Y. Xu, B. Zhu (Eds.), Combinatorial Optimization and Applications. XI, 390 pages. 2007.

Vol. 4615: R. de Lemos, C. Gacek, A. Romanovsky (Eds.), Architecting Dependable Systems IV. XIV, 435 pages. 2007.

Vol. 4613: F.P. Preparata, Q. Fang (Eds.), Frontiers in Algorithmics. XI, 348 pages. 2007.

Vol. 4612: I. Miguel, W. Ruml (Eds.), Abstraction, Reformulation, and Approximation. XI, 418 pages. 2007. (Sublibrary LNAI).

Vol. 4611: J. Indulska, J. Ma, L.T. Yang, T. Ungerer, J. Cao (Eds.), Ubiquitous Intelligence and Computing. XXIII, 1257 pages. 2007.

Vol. 4610: B. Xiao, L.T. Yang, J. Ma, C. Muller-Schloer, Y. Hua (Eds.), Autonomic and Trusted Computing. XVIII, 571 pages. 2007.

Vol. 4609: E. Ernst (Ed.), ECOOP 2007 – Object-Oriented Programming. XIII, 625 pages. 2007.

Vol. 4608: H.W. Schmidt, I. Crnkovic, G.T. Heineman, J.A. Stafford (Eds.), Component-Based Software Engineering. XII, 283 pages. 2007.

Vol. 4607: L. Baresi, P. Fraternali, G.-J. Houben (Eds.), Web Engineering. XVI, 576 pages. 2007.

Vol. 4606: A. Pras, M. van Sinderen (Eds.), Dependable and Adaptable Networks and Services. XIV, 149 pages. 2007.

Vol. 4605: D. Papadias, D. Zhang, G. Kollios (Eds.), Advances in Spatial and Temporal Databases. X, 479 pages. 2007.

Vol. 4604: U. Priss, S. Polovina, R. Hill (Eds.), Conceptual Structures: Knowledge Architectures for Smart Applications. XII, 514 pages. 2007. (Sublibrary LNAI).

Vol. 4603: F. Pfenning (Ed.), Automated Deduction – CADE-21. XII, 522 pages. 2007. (Sublibrary LNAI).

Vol. 4602: S. Barker, G.-J. Ahn (Eds.), Data and Applications Security XXI. X, 291 pages. 2007.

Vol. 4600: H. Comon-Lundh, C. Kirchner, H. Kirchner (Eds.), Rewriting, Computation and Proof. XVI, 273 pages. 2007.

Vol. 4599: S. Vassiliadis, M. Berekovic, T.D. Hämäläinen (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XVIII, 466 pages. 2007.

Vol. 4598: G. Lin (Ed.), Computing and Combinatorics. XII, 570 pages. 2007.

Vol. 4597: P. Perner (Ed.), Advances in Data Mining. XI, 353 pages. 2007. (Sublibrary LNAI).

Vol. 4596: L. Arge, C. Cachin, T. Jurdziński, A. Tarlecki (Eds.), Automata, Languages and Programming. XVII, 953 pages. 2007.

Vol. 4595: D. Bošnački, S. Edelkamp (Eds.), Model Checking Software. X, 285 pages. 2007.

Vol. 4594: R. Bellazzi, A. Abu-Hanna, J. Hunter (Eds.), Artificial Intelligence in Medicine. XVI, 509 pages. 2007. (Sublibrary LNAI).

Vol. 4592: Z. Kedad, N. Lammari, E. Métais, F. Meziane, Y. Rezgui (Eds.), Natural Language Processing and Information Systems. XIV, 442 pages. 2007.

Vol. 4591: J. Davies, J. Gibbons (Eds.), Integrated Formal Methods. IX, 660 pages. 2007.

Vol. 4590: W. Damm, H. Hermanns (Eds.), Computer Aided Verification. XV, 562 pages. 2007.

Vol. 4589: J. Münch, P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. XII, 414 pages. 2007.

Vol. 4588: T. Harju, J. Karhumäki, A. Lepistö (Eds.), Developments in Language Theory. XI, 423 pages. 2007.

Vol. 4587: R. Cooper, J. Kennedy (Eds.), Data Management. XIII, 259 pages. 2007.

Vol. 4586: J. Pieprzyk, H. Ghodosi, E. Dawson (Eds.), Information Security and Privacy. XIV, 476 pages. 2007.

Vol. 4585: M. Kryszkiewicz, J.F. Peters, H. Rybinski, A. Skowron (Eds.), Rough Sets and Intelligent Systems Paradigms. XIX, 836 pages. 2007. (Sublibrary LNAI).

Vol. 4584: N. Karssemeijer, B. Lelieveldt (Eds.), Information Processing in Medical Imaging. XX, 777 pages. 2007.

Vol. 4583: S.R. Della Rocca (Ed.), Typed Lambda Calculi and Applications. X, 397 pages. 2007.

Vol. 4582: J. Lopez, P. Samarati, J.L. Ferrer (Eds.), Public Key Infrastructure. XI, 375 pages. 2007.

Vol. 4581: A. Petrenko, M. Veanes, J. Tretmans, W. Grieskamp (Eds.), Testing of Software and Communicating Systems. XII, 379 pages. 2007.

Vol. 4580: B. Ma, K. Zhang (Eds.), Combinatorial Pattern Matching. XII, 366 pages. 2007.

Vol. 4579: B. M. Hämmerli, R. Sommer (Eds.), Detection of Intrusions and Malware, and Vulnerability Assessment. X, 251 pages. 2007.

Vol. 4578: F. Masulli, S. Mitra, G. Pasi (Eds.), Applications of Fuzzy Sets Theory. XVIII, 693 pages. 2007. (Sublibrary LNAI).

Vol. 4577: N. Sebe, Y. Liu, Y.-t. Zhuang, T.S. Huang (Eds.), Multimedia Content Analysis and Mining. XIII, 513 pages. 2007.

Vol. 4576: D. Leivant, R. de Queiroz (Eds.), Logic, Language, Information and Computation. X, 363 pages. 2007.

Vol. 4575: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.), Pairing-Based Cryptography – Pairing 2007. XI, 408 pages. 2007.

Vol. 4574: J. Derrick, J. Vain (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2007. XI, 375 pages. 2007.

Vol. 4573: M. Kauers, M. Kerber, R. Miner, W. Windsteiger (Eds.), Towards Mechanized Mathematical Assistants. XIII, 407 pages. 2007. (Sublibrary LNAI).

Vol. 4572: F. Stajano, C. Meadows, S. Capkun, T. Moore (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. X, 247 pages. 2007.

Vol. 4571: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition. XIV, 913 pages. 2007. (Sublibrary LNAI).

Vol. 4570: H.G. Okuno, M. Ali (Eds.), New Trends in Applied Artificial Intelligence. XXI, 1194 pages. 2007. (Sublibrary LNAI).

Vol. 4569: A. Butz, B. Fisher, A. Krüger, P. Olivier, S. Owada (Eds.), Smart Graphics. IX, 237 pages. 2007.

Vol. 4568: T. Ishida, S. R. Fussell, P. T. J. M. Vossen (Eds.), Intercultural Collaboration. XIII, 395 pages. 2007.

Vol. 4566: M.J. Dainoff (Ed.), Ergonomics and Health Aspects of Work with Computers. XVIII, 390 pages. 2007.

Vol. 4565: D.D. Schmorrow, L.M. Reeves (Eds.), Foundations of Augmented Cognition. XIX, 450 pages. 2007. (Sublibrary LNAI).

Vol. 4564: D. Schuler (Ed.), Online Communities and Social Computing. XVII, 520 pages. 2007.

Vol. 4563: R. Shumaker (Ed.), Virtual Reality. XXII, 762 pages. 2007.

Vol. 4562: D. Harris (Ed.), Engineering Psychology and Cognitive Ergonomics. XXIII, 879 pages. 2007. (Sublibrary LNAI).

Vol. 4561: V.G. Duffy (Ed.), Digital Human Modeling. XXIII, 1068 pages. 2007.

Vol. 4560: N. Aykin (Ed.), Usability and Internationalization, Part II. XVIII, 576 pages. 2007.

Vol. 4559: N. Aykin (Ed.), Usability and Internationalization, Part I. XVIII, 661 pages. 2007.

Vol. 4558: M.J. Smith, G. Salvendy (Eds.), Human Interface and the Management of Information, Part II. XXIII, 1162 pages. 2007.

Vol. 4557: M.J. Smith, G. Salvendy (Eds.), Human Interface and the Management of Information, Part I. XXII, 1030 pages. 2007.

Vol. 4556: C. Stephanidis (Ed.), Universal Access in Human-Computer Interaction, Part III. XXII, 1020 pages. 2007.

Vol. 4555: C. Stephanidis (Ed.), Universal Access in Human-Computer Interaction, Part II. XXII, 1066 pages. 2007.

Vol. 4554: C. Stephanidis (Ed.), Universal Acess in Human Computer Interaction, Part I. XXII, 1054 pages. 2007.

Vol. 4553: J.A. Jacko (Ed.), Human-Computer Interaction, Part IV. XXIV, 1225 pages. 2007.

Vol. 4552: J.A. Jacko (Ed.), Human-Computer Interaction, Part III. XXI, 1038 pages. 2007.

Vol. 4551: J.A. Jacko (Ed.), Human-Computer Interaction, Part II. XXIII, 1253 pages. 2007.

Vol. 4550: J.A. Jacko (Ed.), Human-Computer Interaction, Part I. XXIII, 1240 pages. 2007.

Preface

On behalf of the program and organizing committee members of this conference, we are pleased to present you with the proceedings of the 12th Asia-Pacific Computer Systems Architecture Conference (ACSAC 2007), which was hosted in Seoul, Korea on August 23-25, 2007. This conference has traditionally been a forum for leading researchers in the Asian, American and Oceanian regions to share recent progress and the latest results in both architectural and system issues. In the past few years the conference has become more international in the sense that the geographic origin of participants has become broader to include researchers from all around the world, including Europe and the Middle East.

This year, we received 92 paper submissions. Each submission was reviewed by at least three primary reviewers along with up to three secondary reviewers. The total number of completed reviews reached 333, giving each submission 3.6 reviews on average. All the reviews were carefully examined during the paper selection process, and finally 26 papers were accepted, resulting in an acceptance rate of about 28%. The selected papers encompass a wide range of topics, with much emphasis on hardware and software techniques for state-of-the-art multicore and multithreaded architectures. In addition to the regular papers, the technical program of the conference included eight invited papers from world-class renowned researchers and featured two keynotes by Pen-Chung Yew (University of Minnesota) and Kunio Uchiyama (Hitachi), addressing *a compiler framework for speculative multithreading* and *power-efficient heterogeneous multicore chip development,* respectively. We sincerely hope that the proceedings will serve as a valuable reference for researchers and developers alike.

Putting together ACSAC 2007 was a team effort. First of all, we would like to express our special gratitude to the authors and speakers for providing the contents of the program. We would also like to thank the program committee members and external reviewers for diligently reviewing the papers and providing suggestions for their improvements. We believe that you will find the outcome of their efforts in this book. In addition, we extend our thanks to the organizing committee members and student volunteers, who contributed enormously to various aspects of conference administration. Finally, we would like to express special thanks to Chris Jesshope and Jinling Xue for sharing their experience and offering fruitful feedback in the early stages of preparing the conference.

June 2007

Lynn Choi
Yunheung Paek
Sangyeun Cho

Conference Organization

General Co-chairs

Lynn Choi	Korea University, Korea
Sung Bae Park	Samsung Electronics, Korea

Program Co-chairs

Yunheung Paek	Seoul National University, Korea
John Morris	University of Auckland, New Zealand
Sangyeun Cho	University of Pittsburgh, USA

Publicity Chair

Ki-Seok Chung	Hanyang University, Korea

Publication Chair

Hwangnam Kim	Korea University, Korea

Local Arrangement Chair

Sung Woo Chung	Korea University, Korea

Finance Chair

Yunmook Nah	Dankook University, Korea

Registration Chair

Youngho Choi	Konkuk University, Korea

Steering Committee

Jesse Z. Fang	Intel, USA
James R. Goodman	University of Auckland, New Zealand
Gernot Heiser	National ICT, Australia

Kei Hiraki	Tokyo University, Japan
Chris Jesshope	University of Amsterdam, Netherlands
Feipei Lai	National Taiwan University, Taiwan
John Morris	University of Auckland, New Zealand
Amos Omondi	Yonsei University, Korea
Ronald Pose	Monash University, Australia
Stanislav Sedukhin	University of Aizu, Japan
Mateo Valero	Universitat Politecnica de Catalunya, Spain
Jingling Xue	University of New South Wales, Australia
Pen-Chung Yew	University of Minnesota, USA

Program Committee

Jin Young Choi	Korea University, Korea
Bruce Christianson	University of Hertfordshire, UK
Sung Woo Chung	Korea University, Korea
Oliver Diessel	University of New South Wales, Australia
Colin Egan	University of Hertfordshire, UK
Skevos Evripidou	University of Cyprus, Cyprus
Wong Weng Fai	National University of Singapore, Singapore
Michael Freeman	University of York, UK
Guang G. Gao	University of Delaware, USA
Jean-Luc Gaudiot	University of California at Irvine, USA
Alex Gontmakher	Technion, Israel
Gernot Heiser	National ICT, Australia
Wei-Chung Hsu	University of Minnesota, USA
Suntae Hwang	Kookmin University, Korea
Chris Jesshope	University of Amsterdam, Netherlands
Jeremy Jones	Trinity College, Ireland
Norman P. Jouppi	Hewlett Packard, USA
Cheol Hong Kim	Chonnam University, Korea
Doohyun Kim	Kunkook University, Korea
Feipei Lai	National Taiwan University, Taiwan
Hock Beng Lim	Nanyang Technological University, Singapore
Philip Machanick	University of Queensland, Australia
Worawan Marurngsith	Thammasat University, Thailand
Henk Muller	University of Bristol, UK
Sukumar Nandi	Indian Institute of Technology Guwahati, India
Tin-Fook Ngai	Intel China Research Center, China
Amos Omondi	Yonsei University, Korea
L M Patnaik	Indian Institute of Science Bangalore, India
Andy Pimentel	University of Amsterdam, Netherlands
Ronald Pose	Monash University, Australia
Stanislav G. Sedukhin	University of Aizu, Japan
Won Shim	Seoul National University of Technology, Korea
Mark Smotherman	Clemson University, USA

K. Sridharan	Indian Institute of Technology Madras, India
Rajeev Thakur	Argonne National Laboratory, USA
Mateo Valero	Universitat Politecnica de Catalunya, Spain
Lucian N. Vintan	University of Sibiu, Romania
Chengyong Wu	ICT, Chinese Academy of Sciences, China
Zhi-Wei Xu	ICT, Chinese Academy of Sciences, China
Jingling Xue	University of New South Wales, Australia
Pen-Chung Yew	University of Minnesota, USA

External Reviewers

Nidhi Aggarwal
Nadeem Ahmed
Christopher Ang
Elizabeth M. Belding-Royer
Darius Buntinas
Francisco Cazorla
José M. Cela
Yang Chen
Doosan Cho
Peter Chubb
Ian Clough
Toni Cortés
Kyriacou Costas
Adrián Cristal
Abhinav Das
Amitabha Das
Michel Dubois
Bin Fan
Jinyun Fang
Yu-Chiann Foo
John Glossner
Sandeep K. Gupta
Rubén Conzález
Rogeli Grima
Jizhong Han
Paul Havinga
Michael Hicks
Houman Homayoun
Kai Hwang
Lei Jin
Jonghee Kang
Kamil Kedzierski
Daeho Kim
Jinpyo Kim
John Kim
Chung-Ta King
Tei-Wei Kuo
Ihor Kuz
Koen Langendoen
Robert Latham
Sanghwan Lee
Heung-No Lee
Hyunjin Lee
Graham Leedham
Binghao Li
Huiyun Li
Kuan-Ching Li
Wei Li
Adam Postula
Chen Liu
Shaoshan Liu
Jie Ma
Luke Macpherson
Pramod K. Meher
Neill Miller
Miquel Moreto
Naveen Muralimanohar
Sudha Natarajan
Venkatesan Packirisamy
Chanik Park
Jagdish Patra
Vladimir Pervouchine
Vinod Prasad
Ken Robinson
Esther Salamí
Oliverio J. Santana
Michael Schelansker
Bill Scherer
Bertil Schmidt
Ahmed Sherif
Todor P. Stefanov
Mark Thompson
Jordi Torres
Nian-Feng Tzeng
Lei Wang
Yulu Yang
Jia Yu
Patryk Zadarnowski
Ahmed Zekri
Ge Zhang
Jony Zhang
Longbing Zhang
Youtao Zhang

Student Volunteers

Yong-Soo Bae
Jae Kyun Jung
Daeho Kim
Hyun-Joon Lee
Kiyeon Lee
Sang-Hoon Lee
Keunhee Yeo
Jonghee Youn

Table of Contents

A Compiler Framework for Supporting Speculative Multicore Processors

Pen-Chung Yew

University of Minnesota at Twin Cities

As multi-core technology is currently being deployed in computer industry primarily to limit power consumption and improve throughput, continued performance improvement of a *single application* on such systems remains an important and challenging task. Because of the shortened on-chip communication latency between cores, using thread-level parallelism (TLP) to improve the number of instructions executed per clock cycle, *i.e.*, to improve ILP performance, has shown to be effective for many *general-purpose* applications. However, because of the program characteristics of these applications, effective speculative schemes at both thread- and instruction-level are crucial.

Processors that support speculative multithreading have been proposed for sometime now. However, efforts have only begun recently to develop compilation techniques for this type of processors. Some of these techniques would require efficient architectural support. The jury is still out on how much performance improvement could be achieved for general-purpose applications on this kind of architectures.

In this talk, we focus on a compiler framework that supports thread-level parallelism with the help of control and data speculation for general-purpose applications. This compiler framework has been implemented on the Open64 compiler that includes support for efficient data dependence and alias profiling, loop selection schemes, as well as speculative compiler optimizations and effective recovery code generation schemes to exploit thread-level parallelism in loops and the remaining code regions.

L. Choi, Y. Paek, and S. Cho (Eds.): ACSAC 2007, LNCS 4697, pp. 1–1, 2007.

Power-Efficient Heterogeneous Multicore Technology for Digital Convergence

Kunio Uchiyama

Hitachi, Ltd

In recent mobile phones, car navigation systems, digital TVs, and other consumer electronic devices, there is a trend toward digital convergence in which a single device has the ability to process various kinds of applications. At the same time, considering the processing of media content, these devices must be capable of encoding and decoding video images and audio data based on MPEG2, MPEG4, H.264, VC-1, MP3, AAC, WMA, RealAudio, and other formats. Moreover, the latest DVD recorders have the ability to automatically generate digests of video images by using audio and image recognition technology. These kinds of digital convergence devices must be able to flexibly process various kinds of data—media, recognition, data, communications, and so on—and the SoC (System-on-Chip) that is embedded in the devices must deliver superior performance while consuming very small power.

To meet these needs, a power-efficient heterogeneous multi-core technology for the SoC used in consumer electronic devices has been developed. Primary objectives in developing this technology are to: (1) establish a robust heterogeneous multicore architecture that integrates a number of different types of power-efficient processor cores; (2) incorporate dynamic reconfigurable processors to leverage parallelism at the operation level; and (3) create a new software development environment for efficiently developing programs tailored for the heterogeneous multicore architecture. This combination of attributes will give us the superior performance/power ratio and flexibility, while satisfying the enormous demand for digital convergence devices.

The power-thrifty processors used in the heterogeneous multicore architecture essentially include a local memory and an intelligent data transfer unit. Each local memory functions as a distributed shared memory for the entire chip. Processing is speeded up by enabling operations within processors in parallel with data transfers between processors. Dynamic reconfigurable processors called Flexible Engines (FEs) have been implemented as a special type of processor core. The FE executes various arithmetic algorithms fast while dynamically changing the functions and interconnections among 32 arithmetic elements.

A prototype heterogeneous multicore chip has been developed using 90nm technology based on the architecture described above. Four low-power CPU cores are integrated along with two FEs on the 96mm^2 chip. The CPU core operates at 600 MHz and has a performance of 1.08 GIPS or 4.2 GFLOPS, while the FE operates at 300 MHz, and can perform up to 19.2 GOPS. The chip as a whole delivers a performance of 4.32 GIPS, 16.8 GOPS, and 38.4 GOPS with a power dissipation of less than several watts.

When a program is executed on the heterogeneous multicore chip, the program is divided up into sub-programs, which are processed by the processor cores on the chip

L. Choi, Y. Paek, and S. Cho (Eds.): ACSAC 2007, LNCS 4697, pp. 2–3, 2007.

that are best suited to the task based on the attributes of each sub-program part. Multimedia programs such as encoding audio data have been executed on various combinations of CPUs and FEs, and the performance and the power consumption of the various configurations have been evaluated.

A new software development environment has been created for the efficient development of programs specifically tailored for the heterogeneous multicore architecture. Using the new platform, programs are broken up into sub-program parts. The object code for the portions executed by CPU cores is generated using a usual compiler. An FE compiler has been developed for the sub-program parts executed by FEs, and the compiler generates configuration data and sequence control codes tailored for the FEs. A graphical interface editor for optimizing FE libraries has also been developed. It not only enables programmers to write FE programs directly but also enables the programs to be verified by simulation.

A part of the introduced research has been supported by NEDO "Advanced heterogeneous multiprocessor," "Multicore processors for real-time consumer electronics," and "Heterogeneous multi-core technology for information appliances."

StarDBT: An Efficient Multi-platform Dynamic Binary Translation System

Cheng Wang, Shiliang Hu, Ho-seop Kim, Sreekumar R. Nair, Mauricio Breternitz Jr., Zhiwei Ying, and Youfeng Wu

Programming Systems Lab, Intel Corporation
2200 Mission College Blvd
Santa Clara, CA 95052, USA
{cheng.c.wang,shiliang.hu,ho-seop.kim,sreekumar.r.nair,
mauricio.breternitz.jr,victor.ying,youfeng.wu}@intel.com

Abstract. This paper describes the design and implementation of a research dynamic binary translation system, StarDBT, which runs many real-world applications. StarDBT is a multi-platform translation system that is capable of translating application level binaries on either Windows or Linux OSes. A system-level variant of StarDBT can also run on a bare machine by translating the whole system code. We evaluate performance of a user-mode system using both SPEC2000 and some challenging Windows applications. StarDBT runs the SPEC2000 benchmark competitively to other state-of-the-art binary translators. For Windows applications that are typically multi-threaded GUI-based interactive applications with large code footprint, the StarDBT system provides acceptable performance in many cases. However, there are important scenarios in which dynamic translation still incurs significant runtime overhead, raising issues for further research. The major overheads are caused by the translation overhead of large volume of infrequently-executed code and by the emulation overhead for indirect branches.

Keywords: Dynamic binary translation, performance evaluation.

1 Introduction

Dynamic binary translation (DBT) has many attractive applications in computer system designs. For example, it can be used to support legacy binary code [4]; support ISA virtualization [1]; enable innovative co-designed microarchitectures [7], [13], and many others [3], [10], [14], [15], [19], [20]. However, DBT technology also comes with its costs: translation overhead, emulation overhead and potentially other runtime overheads. It is an interesting research topic to obtain insights for designing systems featuring binary translation.

To evaluate DBT design and application, we developed a multi-platform DBT system, named StarDBT. StarDBT translates from IA (Intel Architecture, a.k.a 'x86') to IA, including from IA32 to Intel64. As a multi-platform system, StarDBT can run as a user-mode module that resides in user process space. Currently, we have OS-specific support for user-mode DBT on both Linux and Windows x64 platforms. Furthermore, StarDBT can also serve as a system level DBT that runs directly on

L. Choi, Y. Paek, and S. Cho (Eds.): ACSAC 2007, LNCS 4697, pp. 4–15, 2007.