More information about this series at http://www.springer.com/series/7407
Preface

This volume presents the papers accepted for the 28th International Workshop on Languages and Compilers for Parallel Computing (LCPC), held during September 9–11, 2015, in Raleigh, North Carolina, USA. Following a long tradition, LCPC 2015 offered a valuable forum for sharing research on all aspects of concurrency: parallel languages, parallel programming models, compilers, runtime systems, and tools. LCPC 2015 in addition encouraged work that went beyond the scope of scientific computing and enabled parallel programming in new areas, such as mobile computing and data centers.

LCPC 2015 received 44 abstract submissions, 37 of which turned into full submissions. Each full submission received three independent reviews from the Program Committee, and some submissions received an additional review from an external expert reviewer. The Program Committee met to discuss each of the full submissions, and decided to accept 19 regular papers. The accepted papers cover a range of important topics on parallel computing, including programming models, communication and latency, optimizing frameworks, parallelizing compilers, correctness and reliability, applications and data structures. LCPC 2015 additionally included four posters on preliminary research results and reflection of past experiences.

LCPC 2015 was fortunate to include two keynote talks. Paul H.J. Kelly from Imperial College, London, gave a talk titled “Synthesis Versus Analysis: What Do We Actually Gain from Domain-Specificity?” Kelly reflected on the extensive experiences that he and his collaborators had in domain-specific performance optimizations, and offered a series of insights on the profitability of domain-specific optimizations. The second talk, presented by Padma Raghavan from Pennsylvania State University, was titled “Toward Programming Models for Parallel Processing of Sparse Data Sets.” Raghavan discussed the utilization of fine-grain parallelism while reducing the latencies of data accesses for data sets with many dimensions that are sparse.

LCPC 2015 held a panel on “Implications of Emerging Memory Technology (e.g., Persistent Memory, Stacked Memory, Processing in Memory) to the Research on Compilers and Programming Systems.” The panelists include five experts on the topic: Dhruva R. Chakrabarti from HP Labs, Rudolf Eigenmann from Purdue University, David Padua from UIUC, Yan Solihin from NCSU, and Youtao Zhang from the University of Pittsburgh. The panel stimulated discussions on the new challenges and opportunities that emerging memory technology brings to compilers and programming systems research.

We would like to thank all the participants of LCPC 2015 for making it a success. The hard work by the Program Committee and external reviewers in reviewing the submissions is key to ensuring a high-quality technical program. We are indebted to the Steering Committee for the strong support. We give our special thanks to Lawrence Rauchwerger for helping us with the workshop registration and many other organizational issues. We are grateful for the financial support by Cisco, Huawei, Intel,
NVIDIA, and NetApp. Finally, the workshop would not have been a success without the excellent work by the volunteers: Amir Bahmani, Richa Budhiraja, Guoyang Chen, Anwesha Das, Anwesha Das, Yufei Ding, Neha Gholkar, Neha Gholkar, Lin Ning, Xing Pan, Apoorv Parle, Shanil Puri, Tao Qian, Tao Wang, Tiancong Wang, Zhipeng Wei, Bagus Wibowo, and Qi Zhu.

December 2015

Xipeng Shen
Frank Mueller
James Tuck
Organization

Workshop Chairs

Xipeng Shen North Carolina State University, USA
Frank Mueller North Carolina State University, USA
James Tuck North Carolina State University, USA

Workshop Committee

James Brodman Intel Corporation, USA
Calin Cascaval Qualcomm Research Silicon Valley Center, USA
Marcelo Cintra Intel Corporation, USA
Chen Ding University of Rochester, USA
Michael Garland NVIDIA Research, USA
Mike Hind IBM Research, USA
Hironori Kasahara Waseda University, Japan
Xiaoming Li University of Delaware, USA
Sam Midkiff Purdue University, USA
Pablo Montesinos Ortego Qualcomm Research Silicon Valley Center, USA
Peng Tu Intel Corporation, USA
Bo Wu Colorado School of Mines, USA
Qing Yi University of Colorado, Colorado Springs, CO, USA
Jidong Zhai Tsinghua University, China
Huiyang Zhou North Carolina State University, USA

Poster Selection Committee

Aparna University of California, Irvine, CA, USA
Chandramowlishwaran
Xu Liu College of William and Mary, USA
Xipeng Shen North Carolina State University, USA
Zhijia Zhao University of California, Riverside, CA, USA

Steering Committee

Rudolf Eigenmann Purdue University, USA
Alex Nicolau University of California, Irvine, CA, USA
David Padua University of Illinois, USA
Lawrence Rauchwerger Texas A&M University, USA
# Contents

## Programming Models

Size Oblivious Programming with *InfiniMem* ........................................ 3  
*Sai Charan Koduru, Rajiv Gupta, and Iulian Neamtiu*

Low-Overhead Fault-Tolerance Support Using DISC Programming Model ........... 20  
*Mehmet Can Kurt, Bin Ren, and Gagan Agrawal*

Efficient Support for Range Queries and Range Updates Using Contention  
Adapting Search Trees ........................................................................... 37  
*Konstantinos Sagonas and Kjell Winblad*

## Optimizing Framework

Polyhedral Optimizations for a Data-Flow Graph Language ............................ 57  
*Alina Sbirlea, Jun Shirako, Louis-Noël Pouchet, and Vivek Sarkar*

Concurrent Cilk: Lazy Promotion from Tasks to Threads in C/C++ ................. 73  
*Christopher S. Zakian, Timothy A.K. Zakian, Abhishek Kulkarni,  
Buddhika Chamith, and Ryan R. Newton*

Interactive Composition of Compiler Optimizations ..................................... 91  
*Brandon Nesterenko, Wenwen Wang, and Qing Yi*

Asynchronous Nested Parallelism for Dynamic Applications in Distributed Memory ................................................................. 106  
*Ioannis Papadopoulos, Nathan Thomas, Adam Fidel, Dielli Hoxha,  
Nancy M. Amato, and Lawrence Rauchwerger*

## Parallelizing Compiler

Multigrain Parallelization for Model-Based Design Applications  
Using the OSCAR Compiler .................................................................... 125  
*Dan Umeda, Takahiro Suzuki, Hiroki Mikami, Keiji Kimura,  
and Hironori Kasahara*

HYDRA: Extending Shared Address Programming for Accelerator Clusters ........ 140  
*Putt Sakdhnagool, Amit Sabne, and Rudolf Eigenmann*

Petal Tool for Analyzing and Transforming Legacy MPI Applications ........... 156  
*Hadia Ahmed, Anthony Skjellum, and Peter Pirkelbauer*
Communication and Locality

Automatic and Efficient Data Host-Device Communication for Many-Core Coprocessors .......................................................... 173
Bin Ren, Nishkam Ravi, Yi Yang, Min Feng, Gagan Agrawal, and Srimat Chakradhar

Topology-Aware Parallelism for NUMA Copying Collectors ............... 191
Khaled Alnowaiser and Jeremy Singer

An Embedded DSL for High Performance Declarative Communication with Correctness Guarantees in C++ ................................. 206
Nilesh Mahajan, Eric Holk, Arun Chauhan, and Andrew Lumsdaine

Parallel Applications and Data Structures

PNNU: Parallel Nearest-Neighbor Units for Learned Dictionaries .......... 223
H.T. Kung, Bradley McDanel, and Surat Teerapittayanon

Coarse Grain Task Parallelization of Earthquake Simulator GMS Using OSCAR Compiler on Various Cc-NUMA Servers ......................... 238
Mamoru Shimaoka, Yasutaka Wada, Keiji Kimura, and Hironori Kasahara

Conc-Trees for Functional and Parallel Programming ............................. 254
Aleksandar Prokopec and Martin Odersky

Correctness and Reliability

Practical Floating-Point Divergence Detection .................................... 271
Wei-Fan Chiang, Ganesh Gopalakrishnan, and Zvonimir Rakamarić

SMT Solving for the Theory of Ordering Constraints ............................ 287
Cunjing Ge, Feifei Ma, Jeff Huang, and Jian Zhang

An Efficient, Portable and Generic Library for Successive Cancellation Decoding of Polar Codes .................................................... 303
Adrien Cassagne, Bertrand Le Gal, Camille Leroux, Olivier Aumage, and Denis Barthou

Author Index .............................................................................. 319