

Table of Contents

<hr/>	
Session 1: Large scale parallelism	
<hr/>	
QUARC: An Array Programming Approach to High Performance Computing	1
<i>Diptorup Deb, Robert J. Fowler and Allan Porterfield</i>	
Utilizing Concurrency Data Access: A New Theory	16
<i>Xian-He Sun</i>	
ParFuse: Parallel and Compositional Analysis of Message Passing Programs	21
<i>Sriram Ananthakrishnan, Greg Bronevetsky, Mark Baranowski and Ganesh Gopalakrishnan</i>	
Fast Approximate Distance Queries in Unweighted Graphs using Bounded Asynchrony ...	36
<i>Adam Fidel, Francisco Coral Sabido, Colton Riedel, Nancy Amato and Lawrence Rauchwerger</i>	
Energy Avoiding Matrix Multiply	51
<i>Kelly Livingston, Aaron Landwehr, Jose Monsalve Diaz, Stephane Zuckerman, Benoit Meister and Guang Gao</i>	
<hr/>	
Session 2: Resilience and persistence	
<hr/>	
Language Support for Reliable Memory Regions	66
<i>Saurabh Hukerikar and Christian Engelmann</i>	
Harnessing Parallelism in Multicore Systems to Expedite and Improve Function Approximation	78
<i>Aurangzeb and Rudolf Eigenmann</i>	
Adaptive Software Caching for Efficient NVRAM Data Persistence	82
<i>Pengcheng Li and Dhruva Chakrabarti</i>	
<hr/>	
Session 3A: Compiler analysis and optimization	
<hr/>	
Polyhedral Compiler Technology in Collaboration with Autotuning Important to Domain-Specific Frameworks for HPC	86
<i>Mary Hall and Protonu Basu</i>	
An Extended Polyhedral Model for SPMD Programs and its use in Static Data Race Detection	90
<i>Prasanth Chatarasi, Jun Shirako, Martin Kong and Vivek Sarkar</i>	
Polygonal Iteration Space Partitioning	105
<i>Aniket Shivam, Alexandru Nicolau, Alex V. Veidenbaum, Mario Mango Furnari and Ro Cammarota</i>	
<hr/>	
Session 3B: Compiler analysis and optimization	
<hr/>	
Automatically Optimizing Stencil Computations on Many-core NUMA Architectures	120
<i>Pei-Hung Lin, Qing Yi, Daniel Quinlan, Chunhua Liao and Yongqing Yan</i>	

Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation	135
<i>Tao Schardl</i>	
Formalizing Structured Control Flow Graphs	136
<i>Amit Sabne, Putt Sakdhnagool and Rudolf Eigenmann</i>	
<hr/> Session 4: Dynamic computation and languages <hr/>	
Automatic Vectorization for MATLAB	140
<i>Hanfeng Chen, Alexander Krolik, Erick Lavoie and Laurie Hendren</i>	
Analyzing Parallel Programming Models for Magnetic Resonance Imaging	155
<i>Forest Danford, Eric Welch, Julio Cárdenas-Rodríguez and Michelle Strout</i>	
The Importance of Fine-Grain Synchronization for Many-Core Systems	170
<i>Tongsheng Geng, Stéphane Zuckerman, Jose Monsalve, Alfredo Goldman, Sami Habib, Jean-Luc Gaudiot and Guang R. Gao</i>	
Optimizing LOBPCG: Sparse Matrix Loop and Data Transformations in Action	185
<i>Khalid Ahmad, Anand Venkat and Mary Hall</i>	
<hr/> Session 5: GPUs and private memory <hr/>	
LightHouse: An Automatic Code Generator for Graph Algorithms on GPUs	200
<i>Shashidhar G and Rupesh Nasre.</i>	
Locality-aware Task-parallel Execution on GPUs	215
<i>Jad Hbeika and Milind Kulkarni</i>	
Automatic Copying of Pointer-Based Data Structures	230
<i>Tong Chen, Zehra Sura and Hyojin Sung</i>	
Automatic Local Memory Management for Multicores Having Global Address Space	245
<i>Kouhei Yamamoto, Tomoya Shirakawa, Yoshitake Oki, Akimasa Yoshida, Keiji Kimura and Hironori Kasahara</i>	
<hr/> Session 6: Run-time and performance analysis <hr/>	
Mapping Medley: Adaptive Parallelism Mapping with Varying Optimization Goals	260
<i>Murali Emani</i>	
The Contention Avoiding Concurrent Priority Queue	275
<i>Kjell Winblad and Konstantinos Sagonas</i>	
Evaluating Performance of Task and Data Coarsening in Concurrent Collections	290
<i>Chenyang Liu and Milind Kulkarni</i>	