

1. Record Nr.	UNISA996466296603316
Titolo	OpenMP: Heterogenous Execution and Data Movements [[electronic resource]] : 11th International Workshop on OpenMP, IWOMP 2015, Aachen, Germany, October 1-2, 2015, Proceedings // edited by Christian Terboven, Bronis R. de Supinski, Pablo Reble, Barbara M. Chapman, Matthias S. Müller
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2015
ISBN	3-319-24595-3
Edizione	[1st ed. 2015.]
Descrizione fisica	1 online resource (XI, 274 p. 146 illus. in color.)
Collana	Programming and Software Engineering ; ; 9342
Disciplina	004
Soggetti	Microprocessors Programming languages (Electronic computers) Computer system failures Computer hardware Algorithms Software engineering Processor Architectures Programming Languages, Compilers, Interpreters System Performance and Evaluation Computer Hardware Algorithm Analysis and Problem Complexity Software Engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Preface -- Organization -- Contents -- Applications -- PAGANtec: OpenMP Parallel Error Correction for Next-Generation Sequencing Data -- 1 Introduction -- 2 Related Work -- 2.1 k-mer Graph and Error Correction -- 2.2 Parallelization Options -- 3 PAGANtec Architecture -- 3.1 Graph Structure -- 3.2 Correction Strategies -- 3.3 Correcting Errors -- 4 Parallelization -- 4.1 Performance Analysis -- 5 Conclusion -- References -- Composing

Low-Overhead Scheduling Strategies for Improving Performance of Scientific Applications -- 1 Introduction -- 2 Scheduling Strategies -- 3 Techniques for Composing Scheduling Strategies -- 3.1 uSched -- 3.2 slackSched -- 3.3 vSched -- 3.4 ComboSched -- 4 Code Transformation -- 5 Results -- 6 Related Work -- 7 Conclusions -- References -- Exploiting Fine- and Coarse-Grained Parallelism Using a Directive Based Approach -- 1 Introduction -- 2 Related Work -- 3 Background: OpenMP Accelerator Model -- 4 An Offloading Model for a Cluster -- 4.1 Definitions -- 4.2 Execution Model -- 4.3 Memory Model -- 5 Implementation -- 5.1 Runtime Support -- 6 Preliminary Results -- 7 Discussion -- 8 Conclusions -- References -- Accelerator Applications -- Experiences of Using the OpenMP Accelerator Model to Port DOE Stencil Applications -- 1 Introduction -- 2 OpenMP 4.0's Accelerator Support -- 3 Applications -- 4 Porting to GPUs -- 4.1 Baseline Performance on CPU and GPU -- 4.2 Increasing Parallelism -- 4.3 Loop Scheduling -- 4.4 Exploiting Memory Hierarchy -- 4.5 Reducing Memory Movement Between Host and Device -- 4.6 Manual Tuning for GPU Performance -- 4.7 Productivity -- 5 Related Work -- 6 Discussion and Future Work -- References -- Evaluating the Impact of OpenMP 4.0 Extensions on Relevant Parallel Workloads -- 1 Introduction and Motivation -- 2 Application Parallelization -- 2.1 Facesim -- 2.2 Fluidanimate. 2.3 Streamcluster -- 3 Evaluation -- 3.1 Performance Evaluation -- 3.2 Programmability -- 4 Related Work -- 5 Conclusions -- References -- First Experiences Porting a Parallel Application to a Hybrid Supercomputer with OpenMP 4.0 Device Constructs -- 1 Introduction -- 2 OpenMP Device Constructs -- 2.1 Data Regions -- 3 A High Level View of the Porting Method -- 3.1 Fusing Local Data Regions -- 4 Porting NekBone -- 5 Conclusions -- References -- Tools -- Lessons Learned from Implementing OMPD: A Debugging Interface for OpenMP -- 1 Introduction -- 2 Prior Work -- 3 The OpenMP Debugging Interface -- 3.1 OMPT: A Runtime Interface for OpenMP Tools -- 3.2 Why Distinguish OMPD from OMPT? -- 3.3 The OMPD Architecture -- 4 Use Cases of OMPD -- 4.1 OpenMP-Aware Stack Trace -- 4.2 Stepping in and Out of a Parallel Region -- 5 OMPD Callback Interface -- 5.1 Functions for Operating System Interaction -- 5.2 Resolving Structures for Target Architecture -- 5.3 Access Application Memory -- 5.4 Debugger's Context Argument -- 6 OMPD API Function Specifications -- 6.1 Providing Information on Compatible Runtime Library -- 6.2 API Specification for Breakpoints -- 6.3 Missing Function to Identify Master -- 7 Future Challenges -- 7.1 Context Pointer for Accelerators -- 7.2 Addressing Accelerator Threads -- 7.3 Return Codes -- 8 Conclusions -- References -- False Sharing Detection in OpenMP Applications Using OMPT API -- 1 Introduction -- 2 Motivation -- 3 Related Work -- 4 OMPT- Application Programming Interface for Tools -- 5 Our Approach -- 5.1 OMPT for Capturing Unique Patterns -- 5.2 Hardware Performance Information -- 5.3 Binary Classifier for False Sharing Detection -- 5.4 Feature Selection -- 6 Experimentation and Results -- 6.1 Training Phase -- 6.2 Validation of the Approach -- 7 Conclusion and Future Work -- References. Exception Handling with OpenMP in Object-Oriented Languages -- 1 Introduction -- 2 Related Work -- 3 Problem Overview -- 3.1 Current Situation -- 3.2 Problem Definition -- 4 Cancellations -- 5 Exception Handling -- 5.1 Overview of Categorization -- 5.2 Local Exception Handling -- 5.3 Global Exception Handling -- 6 Implementation -- 6.1 Adaptable Synchronization Barrier -- 6.2 Dynamic Work Redistribution -- 6.3 Exception from Synchronization Regions -- 6.4 Global Exception Throwing -- 7 Evaluation -- 7.1 Usability -- 7.2 Performance -- 8

Conclusion -- References -- Extensions -- On the Algorithmic Aspects of Using OpenMP Synchronization Mechanisms II: User-Guided Speculative Locks -- 1 Introduction -- 2 Related Work -- 3 User-Guided Locking API with TSX -- 3.1 Intel Transactional Synchronization Extensions -- 3.2 Using the User-Guided Locking API -- 4 Applying Intel TSX to the Test Code -- 4.1 A Brief Review of the Algorithm -- 4.2 The Role of TSX -- 5 Experimental Results -- 5.1 Convergence -- 5.2 Transactional Memory Statistics -- 5.3 Performance Measurement -- 6 Conclusions and Future Work -- References -- Using Transactional Memory to Avoid Blocking in OpenMP Synchronization Directives -- 1 Introduction -- 2 Avoiding Blocking in OpenMP -- 2.1 Critical Sections -- 2.2 Barrier/Taskwait -- 3 Evaluation -- 3.1 Experimental Setup -- 3.2 Results -- 4 Limitations and Related Work -- 5 Conclusion -- References -- A Case Study of OpenMP Applied to Map/Reduce-Style Computations -- 1 Introduction -- 2 Related Work -- 3 Map-Reduce Programming Model -- 3.1 Phoenix++ Implementation -- 3.2 OpenMP Facilities for Map/Reduce-Style Computations -- 4 OpenMP Implementations -- 4.1 Histogram -- 4.2 Linear Regression -- 4.3 K-Means Clustering -- 4.4 Word Count -- 4.5 String Match -- 4.6 Matrix Multiply -- 4.7 Principal Component Analysis -- 5 Evaluation -- 5.1 Analysis.

5.2 Coding Style Comparison -- 5.3 Implications to OpenMP -- 6 Conclusion -- References -- Compiler and Runtime -- Enabling Region Merging Optimizations in OpenMP -- 1 Introduction -- 2 Region Merging and Control -- 2.1 Region Merging Validity in OpenMP -- 2.2 Syntax Extensions to Support Merging -- 3 Results and Evaluation -- 3.1 Back to Back Regions -- 3.2 Parallel Regions with Intervening Serial Regions -- 3.3 Lulesh -- 4 Related Work -- 5 Conclusion -- References -- Towards Task-Parallel Reductions in OpenMP -- 1 Introduction -- 2 Related Work -- 3 Discussion -- 3.1 Updates of a Reduction Variable Outside a Reduction Context -- 3.2 Over-Specifying the Reduction Identifier -- 3.3 Supporting Untied Tasks -- 3.4 Supporting Nested Taskgroups -- 3.5 Cancellation, Dependencies and Merged Tasks -- 4 Syntax Additions -- 5 Evaluation -- 5.1 System Environment -- 5.2 Benchmark Descriptions -- 5.3 Performance Results on Intel Xeon Processors -- 5.4 Performance Results on Intel Xeon Phi Coprocessors -- 6 Conclusions and Future Work -- References -- OpenMP 4.0 Device Support in the OMPi Compiler -- 1 Introduction -- 2 Background -- 3 Compiler Transformations -- 3.1 Target Data -- 3.2 Target -- 3.3 Declare Target -- 4 Runtime Support -- 4.1 Data Environment Handling -- 5 The Epiphany Accelerator as a Device -- 5.1 Runtime Organization -- 5.2 Experiments -- 6 Discussion and Current Status -- References -- Energy -- Application-Level Energy Awareness for OpenMP -- 1 Introduction -- 2 Motivation -- 3 OpenMPE -- 4 Compilation and Runtime System -- 5 Evaluation -- 6 Related Work -- 7 Conclusion and Future Work -- References -- Evaluating the Energy Consumption of OpenMP Applications on Haswell Processors -- 1 Introduction -- 2 Related Works -- 3 Basic Characteristics -- 3.1 Energy-Saving Features of Haswell -- 3.2 Load-Dependent Behavior -- 4 Optimization Steps. 4.1 Wait Strategies -- 4.2 Iterative Clock Adjustment -- 4.3 Evaluation -- 5 Conclusion -- References -- Parallelization Methods for Hierarchical SMP Systems -- 1 Introduction -- 2 The Test Code -- 3 SIMD Building Blocks -- 4 Nested Threading -- 5 Code Variants -- 5.1 Baseline -- 5.2 Hand Decomposed -- 5.3 Nested Parallelism -- 5.4 Hand Nested -- 5.5 Crew and Teams -- 5.6 SBB -- 6 Performance Experiments -- 7 Conclusions and Future Work -- References -- Supporting Indirect Data Mapping in OpenMP -- 1 Introduction -- 2 The OpenMP 4.0 Data Environment -- 2.1

Mapping Syntax -- 2.2 Presence -- 3 Map Refinements -- 3.1 Data only Array Sections -- 3.2 Type-Based Implicit Mappings -- 4 Clause Grouping and Binding -- 5 Conclusion -- References -- Author Index.

Sommario/riassunto

This book constitutes the refereed proceedings of the 11th International Workshop on OpenMP, held in Aachen, Germany, in October 2015. The 19 technical full papers presented were carefully reviewed and selected from 22 submissions. The papers are organized in topical sections on applications, accelerator applications, tools, extensions, compiler and runtime, and energy.
