1. Record Nr. UNINA9910143568803321

Titolo	High-performance computing [[electronic resource]] : paradigm and infrastructure / / edited by Laurence T. Yang, Minyi Guo
Pubbl/distr/stampa	Hoboken, N.J., : J. Wiley, c2006
ISBN	1-280-31131-2 9786610311316 0-470-36083-6 0-471-73271-0 0-471-73270-2
Descrizione fisica	1 online resource (818 p.)
Collana	Wiley series on parallel and distributed computing
Altri autori (Persone)	YangLaurence Tianruo GuoMinyi
Disciplina	004/.35 502.85435
Soggetti	High performance computing Parallel processing (Electronic computers) Electronic data processing - Distributed processing Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	HIGH-PERFORMANCE COMPUTING; Contents; Preface; Contributors; PART 1 Programming Model; 1 ClusterGOP: A High-Level Programming Environment for Clusters; 1.1 Introduction; 1.2 GOP Model and ClusterGOP Architecture; 1.2.1 The ClusterGOP Architecture; 1.3 VisualGOP; 1.4 The ClusterGOP Library; 1.5 MPMD programming Support; 1.6 Programming Using ClusterGOP; 1.6.1 Support for Program Development; 1.6.2 Performance of ClusterGOP; 1.7 Summary; 2 The Challenge of Providing A High-Level Programming Model for High-Performance Computing; 2.1 Introduction; 2.2 HPC Architectures 2.2.1 Early Parallel Processing Platforms2.2.2 Current HPC Systems; 2.3 HPC Programming Models: The First Generation; 2.3.1 The Message Passing Interface (MPI); 2.3.2 High Performance Fortran (HPF); 2.4 The Second Generation of HPC Programming Models; 2.4.1 OpenMP; 2.4.2 Other Shared-Memory APIs; 2.4.3 Is A Standard High-Level API for HPC

in Sight?; 2.5 OpenMP for DMPs; 2.5.1 A Basic Translation to GA; 2.5.2 Implementing Sequential Regions; 2.5.3 Data and Work Distribution in GA; 2.5.4 Irregular Computation Example; 2.6 Experiments with OpenMP on DMPs; 2.7 Conclusions
3 SAT: Toward Structured Parallelism Using Skeletons3.1 Introduction; 3.2 SAT: A Methodology Outline; 3.2.1 Motivation and Methodology; 3.2.2 Abstraction View: Basic Skeletons and Compositions; 3.2.3 Performance View: Collective Operations; 3.2.4 SAT: Combining Abstraction with Performance; 3.3 Skeletons and Collective Operations; 3.3.1 The H Skeleton and Its Standard Implementation; 3.3.2 Transformations for Performance View; 3.4 Case Study: Maximum Segment SUM (MSS); 3.5 Performance Aspect in SAT; 3.5.1 Performance Predictability; 3.5.2 Absolute Performance; 3.6 Conclusions and Related Work
4 Bulk-Synchronous Parallelism: An Emerging Paradigm of High-Performance Computing4.1 The BSP Model; 4.1.2 BSP Versus Traditional Parallelism; 4.1.3 Memory Efficiency; 4.1.4 Memory Management; 4.1.5 Heterogeneity; 4.1.6 Subset Synchronization; 4.1.7 Other Variants of BSP; 4.2 BSP Programming; 4.2.1 The BSPlib Standard; 4.2.2 Beyond BSPlib; 4.3 Conclusions; 5 Cilk Versus MPI: Comparing Two Parallel Programming Styles on Heterogeneous Systems; 5.1 Introduction; 5.1.1 Message-Passing Run-Time Systems; 5.1.2 Cilk's Dataflow Model; 5.1.3 Terminology; 5.2 Experiments; 5.2.1 Programs; 5.2.2 Test Bed
5.3 Results5.3.1 Fibonacci; 5.3.2 Traveling Salesman Problem; 5.3.3 N-Queens Problem; 5.3.4 Matrix Multiplication; 5.3.5 Finite Differencing; 5.3.6 Program Complexity; 5.4 Conclusion; 6 Nested Parallelism and Pipelining in OpenMP; 6.1 Introduction; 6.2 OpenMP Extensions for Nested Parallelism; 6.2.1 Parallelism Definition; 6.2.2 Thread Groups; 6.2.3 Evaluation of the Proposal; 6.3 OpenMP Extensions For Thread Synchronization; 6.3.1 Precedence Relations; 6.3.2 Evaluation of the Proposal; 6.4 Summary; 7 OpenMP for Chip Multiprocessors; 7.1 Introduction; 7.2 3SoC Architecture Overview
7.2.1 Quads

Sommario/riassunto

The state of the art of high-performance computingProminent researchers from around the world have gathered to present the state-of-the-art techniques and innovations in high-performance computing (HPC), including:* Programming models for parallel computing: graph-oriented programming (GOP), OpenMP, the stages and transformation (SAT) approach, the bulk-synchronous parallel (BSP) model, Message Passing Interface (MPI), and Cilk* Architectural and system support, featuring the code tiling compiler technique, the MigThread application-level migration and checkpointing package, th