

1. Record Nr.	UNISA996601561403316
Autore	Diehl Patrick
Titolo	Asynchronous Many-Task Systems and Applications : Second International Workshop, WAMTA 2024, Knoxville, TN, USA, February 14-16, 2024, Proceedings
Pubbl/distr/stampa	Cham : , : Springer International Publishing AG, , 2024 ©2024
ISBN	3-031-61763-0
Edizione	[1st ed.]
Descrizione fisica	1 online resource (196 pages)
Collana	Lecture Notes in Computer Science Series ; ; v.14626
Altri autori (Persone)	SchuchartJoseph Valero-LaraPedro BosilcaGeorge
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Contents -- Speaking Pygion: Experiences Writing an Exascale Single Particle Imaging Code -- 1 Introduction -- 2 Related Work -- 3 SpiniFEL -- 4 Pygion Implementation -- 5 Results -- 6 Conclusion -- References -- Futures for Dynamic Dependencies - Parallelizing the H-LU Factorization -- 1 Introduction -- 2 Background -- 3 Future-Based Algorithm -- 4 Definition of Futures -- 5 Pseudocode and Discussion -- 6 Related Work -- 7 Conclusion -- References -- Evaluating PaRSEC Through Matrix Computations in Scientific Applications -- 1 Introduction -- 2 Related Work -- 3 The PaRSEC Runtime System -- 4 Applications as Testbed -- 5 Performance Results and Analysis -- 5.1 Experimental Settings -- 5.2 Load Balancing -- 5.3 GPU Efficiency -- 5.4 Scalability -- 6 Conclusion and Future Work -- References -- Distributed Asynchronous Contact Mechanics with DARMA/vt -- 1 Introduction -- 2 Prior Work -- 3 Algorithm -- 3.1 Update and Tree Build -- 3.2 Broadphase -- 3.3 Midphase and Ghosting -- 3.4 Narrowphase -- 3.5 Load Balancing -- 4 Results -- 5 Conclusion -- References -- IRIS Reimagined: Advancements in Intelligent Runtime System for Task-Based Programming -- 1 Introduction -- 2 Background: IRIS -- 3 Related Work -- 4 IRIS Re-imagined -- 4.1 Vendor-Specific Kernels --

4.2 Foreign Function Interface (FFI) -- 4.3 Distributed Data Memory Management (DMEM) -- 4.4 Heterogeneous Build Environment for IRIS Applications -- 4.5 Hunter -- 4.6 DAGGER -- 5 Results -- 5.1 FFI -- 5.2 DMEM -- 5.3 DAGGER -- 6 Conclusion -- References -- MatRIS: Addressing the Challenges for Portability and Heterogeneity Using Tasking for Matrix Decomposition (Cholesky) -- 1 Introduction -- 2 Background: Cholesky Decomposition, IRIS, and MatRIS -- 2.1 Cholesky Decomposition -- 2.2 IRIS -- 2.3 MatRIS -- 3 Related Work -- 4 Cholesky Decomposition in MatRIS.

4.1 Abstractions for Memory and Computation -- 4.2 Kernel APIs for Cholesky -- 4.3 Tiled Cholesky in MatRIS -- 5 Experiments -- 5.1 Portability, Scalability, and Utilization of Cholesky -- 5.2 Multi-GPU Scalability of Cholesky -- 5.3 Comparison of Cholesky with Vendor Libraries -- 5.4 Heterogeneous Scheduling Opportunities -- 6 Conclusion -- References -- ParSweet: A Suite of Codes for Benchmarking and Testing Mutex-Based Parallel Systems -- 1 Introduction -- 2 Mutex Implementations -- 3 Parallel Codes -- 3.1 Sets -- 3.2 Maps -- 4 Benchmarks and Tests -- 4.1 Machines -- 4.2 Lock Benchmark -- 4.3 Set Benchmark (SetByLock) -- 4.4 Map Benchmark (MapByLock) -- 5 Results -- 5.1 Locks -- 5.2 SetByLocks -- 5.3 MapByLocks -- 6 Conclusion and Future Work -- References -- Rethinking Programming Paradigms in the QC-HPC Context -- 1 Introduction -- 2 Quantum Programming Tools -- 3 Task Modeling in Quantum Computation -- 4 Perspective on the Role of Quantum Technology -- References -- Dynamic Tuning of Core Counts to Maximize Performance in Object-Based Runtime Systems -- 1 Introduction -- 2 Background and Implementation -- 2.1 Implementation of Tuning Core Counts in Charm++ -- 2.2 Additional Changes to Charm++ features -- 2.3 Turning Cores Off Without Suspending -- 2.4 Programming API -- 3 Evaluation -- 3.1 System and Benchmarks -- 3.2 Tuning Physical/virtual Core Count for Performance (and Energy and Power Savings) -- 3.3 Overheads -- 4 Related Work -- 5 Conclusion and Future Work -- References -- Enhancing Sparse Direct Solver Scalability Through Runtime System Automatic Data Partition -- 1 Introduction -- 2 Task-Based Sparse Factorization -- 3 Implementation Within the PaStiX Solver -- 4 Experiments -- 5 Related Work -- 6 Conclusion -- References -- Experiences Porting Shared and Distributed Applications to Asynchronous Tasks: A Multidimensional FFT Case-Study.

1 Introduction -- 2 Related Work -- 3 Methods -- 3.1 Fast Fourier Transform -- 3.2 Parallelization -- 3.3 Different Implementations -- 4 Software Framework -- 4.1 HPX -- 4.2 FFTW -- 5 Results -- 5.1 Overheads -- 5.2 FFTW Backend -- 5.3 Distributed -- 6 Conclusion and Outlook -- References -- An Abstraction for Distributed Stencil Computations Using Charm++ -- 1 Introduction -- 2 Background -- 3 Methodology -- 3.1 Frontend -- 3.2 Backend -- 4 Performance Results -- 5 Related Work -- 6 Future Work -- 7 Conclusion -- References -- DLA-Future: A Task-Based Linear Algebra Library Which Provides a GPU-Enabled Distributed Eigensolver -- 1 Introduction -- 2 DLA-Future -- 2.1 Eigensolver Implementation Description -- 2.2 Implementation Challenges -- 3 Results -- 3.1 Eigensolver -- 3.2 Integration in CP2K -- 4 Conclusion -- References -- ALPI: Enhancing Portability and Interoperability of Task-Aware Libraries -- 1 Introduction -- 2 Background -- 2.1 Task-Based Runtime Systems -- 2.2 Task-Aware Libraries -- 3 The ALPI Interface -- 4 Implementing TAMPI Using the ALPI Interface -- 5 Interoperability Between TA-X Libraries -- 6 Conclusions -- References -- Evolving APGAS Programs: Automatic and Transparent Resource Adjustments at Runtime -- 1

Introduction -- 2 Background -- 3 Evolving APGAS Programs -- 3.1
Lifecycle -- 3.2 Programmer Abstractions -- 3.3 Heuristics -- 3.4
Example: GLB Library -- 4 Evaluation -- 4.1 EvoTree Benchmark -- 4.2
Experiments -- 5 Related Work -- 6 Conclusion -- References --
Optimizing Parallel System Efficiency: Dynamic Task Graph Adaptation
with Recursive Tasks -- 1 Introduction -- 2 Granularity Challenges
Within the STF Model -- 3 Just-in-Time Task Splitting in StarPU -- 4
Study Case: Cholesky Factorisation -- 5 Conclusion -- References.
HPX with Spack and Singularity Containers: Evaluating Overheads for
HPX/Kokkos Using an Astrophysics Application -- 1 Introduction -- 2
Related Work -- 3 Software Stack -- 3.1 Notable Octo-Tiger
Dependencies -- 3.2 Octo-Tiger -- 3.3 Build and Dependency
Management -- 4 Workflow -- 4.1 Challenges in Compiling and
Running Within Containers -- 5 Performance Differences -- 5.1
Supercomputer Fugaku (A64FX) -- 5.2 DeepBayou -- 6 Conclusion and
Outlook -- References -- Author Index.
