Publication Series of the John von Neumann Institute for Computing (NIC)
NIC Series Volume 37

John von Neumann Institute for Computing (NIC)

Parallel Computing: Architectures, Algorithms and Applications

Book of Abstracts

edited by
Gerhard Joubert
Christian Bischof
Frans Peters
Thomas Lippert
Martin Bücker
Paul Gibbon
Bernd Mohr

ParCo 2007 Conference, 4. - 7. September 2007
organized by
Forschungszentrum Jülich
RWTH Aachen University

NIC Series Volume 37

ISBN 978-3-9810843-3-7

Die Deutsche Bibliothek – CIP-Cataloguing-in-Publication-Data
A catalogue record for this publication is available from Die Deutsche Bibliothek.

Publisher: NIC-Directors

Distributor: NIC-Secretariat
Research Centre Jülich
52425 Jülich
Germany

Internet: www.fz-juelich.de/nic

Printer: Graphische Betriebe, Forschungszentrum Jülich

NIC Series Volume 37
ISBN 978-3-9810843-3-7

Preface

ParCo2007 marks a quarter of a century of ParCo conferences: the longest running series of international meetings on the development and application of high speed parallel computing technologies in Europe.

The aim of this year's conference, which is jointly organised by the Forschungszentrum (Research Centre) Jülich and the RWTH Aachen University in Germany, is to give an overview of the state-of-the-art of developments with regard to compute system architectures, algorithms and applications as well as future trends in high performance computing. The conference addresses all aspects of parallel computing, including: applications in physical, life and engineering sciences; the design, analysis and implementation of parallel algorithms; hardware and software technologies, programming languages, development environments and performance tools.

The scientific program consists of 70 contributed papers covering the above topics, preceded each morning by one of 4 invited papers by Maria Ramalho-Natario (European Commission), Barbara Chapman (University of Houston), Marek Behr (RWTH Aachen University) and Satoshi Matsuoka (Tokyo Institute of Technology). In addition to the regular programme there are also 5 Mini-Symposia on the following special topics:

- Parallel Computing with FPGAs
- The Future of OpenMP in the Multi-Core Era
- Scalability and Usability of HPC Programming Tools
- DEISA: Extreme Computing in an Advanced Supercomputing Environment
- Scaling Science Applications on Blue Gene

A special word of thanks is due to our sponsors: Forschungszentrum Jülich, IBM Germany, ParTec Cluster Competence Center, and RWTH Aachen University. Their support enabled the organization of a number of highlights, e.g., student awards, and keeping the conference fee at the same level as in recent years.

Gerhard Joubert, TU Clausthal, Germany (Conference Chair)
Frans Peters, Philips Research, The Netherlands (Finance Chair)
Thomas Lippert, FZ Jülich, Germany (Organising and Symposium Committee Chair)
Christian Bischof, RWTH Aachen University, Germany (Program Committee Chair)
Martin Bücker, RWTH Aachen University, Germany (PC Subchair Algorithms)
Paul Gibbon, FZ Jülich, Germany (PC Subchair Applications)
Bernd Mohr, FZ Jülich, Germany (PC Subchair Software and Architectures)

Timetable

	Tuesday September 4, 2007 Jülich	**Wednesday** September 5, 2007 Jülich	**Thursday** September 6, 2007 Aachen	**Friday** September 7, 2007 Jülich
09:00	**Registration & Opening Session**	**Registration & Opening Session**	**Registration & Opening Session**	**Registration & Opening Session**
09:30		**Invited Talk** Barbara Chapman	**Invited Talk** Marek Behr	**Invited Talk** Satoshi Matsuoka
10:00	**Invited Talk** Maria Ramalho-Natario			
10:30		**Coffee**	**Coffee**	**Coffee**
11:00	**Coffee**	**Session 4** A4: Parallel Computing with FPGAs B4: Numerical Algorithms I	**Session 7** A7: Scheduling B7: Performance Analysis I C7: Biomedical Applications	**Session 9** A9: Parallel Tools and Middleware B9: Image Processing and Visualization C9: Fluid Dynamics Simulation D9: Hyperscalable Applications
11:30	**Session 1** A1: Electronic Structure Simulation B1: Parallel Performance Tools			
12:00	**Mini Symposia 1** C1: The Future of OpenMP D1: Scaling Science Applications	**Mini Symposia 4** C4: Scalability/Usability of Tools D4: DEISA Extreme Computing	**Mini Symposia 7** D7: Parallel Computing with FPGAs E7: Vendor Session	
12:30	**Lunch**	**Lunch**	**Lunch**	**Lunch**
13:00				
13:30				
14:00	**Session 2** A2: Particle + Atomic Simulation B2: Performance Modeling and Analysis	**Session 5** A5: Parallel Programming Models B5: Numerical Algorithms II	**Session 8** A8: Fault Tolerance B8: Performance Analysis II C8: MHD and Turbulence Simulation	
14:30		**Mini Symposia 5** C5: Scalability/Usability of Tools D5: DEISA Extreme Computing		
15:00	**Mini Symposia 2** C2: The Future of OpenMP D2: Scaling Science Applications		**Mini Symposia 8** D8: Parallel Computing with FPGAs E8: Vendor Session	
15:30		**Coffee**		
16:00	**Coffee**	**Session 6** A6: Parallel Data Distr. and I/O B6: Parallel Automatic Differentiation		
16:30	**Session 3** A3: Image Reconstruction B3: Parallel Algorithms C3: Parallel Computing with OpenMP	**Mini Symposia 5** C6: Scalability/Usability of Tools D6: DEISA Extreme Computing		
17:00				
17:30	**Mini Symposia 3** D3: Scaling Science Applications			

Contents

Invited Talks

European E-Infrastructure: Promoting Global Virtual Research Communities
Maria Ramalho-Natario **3**

Programming in the Multicore Era
Barbara Chapman **4**

Simulation of Heart-Assist Devices
Marek Behr **5**

Towards Petascale Grids as a Foundation of E-Science
Satoshi Matsuoka **6**

Session "Electronic Structure Simulation"

Domain Decomposition for Electronic Structure Computations
Guy Bencteux, Maxime Barrault, Eric Cancès, William Hager, Claude Le Bris **9**

Scalable Dynamic Adaptations for Electronic Structure Calculations
Masha Sosonkina **10**

Session "Parallel Performance Tools"

Visualizing Parallel Functional Program Runs: Case Studies with the Eden Trace Viewer
Jost Berthold and Rita Loogen **13**

Automatic Phase Detection of MPI Applications
Marc Casas, Rosa M. Badia, Jesús Labarta **14**

Session "Particle and Atomistic Simulation"

Load Balanced Parallel Simulation of Particle-Fluid DEM-SPH Systems with Moving Boundaries
Florian Fleissner, Peter Eberhard **17**

Communication and Load Balancing of Force-Decomposition Algorithms for Parallel Molecular Dynamics
Godehard Sutmann, Florian Janoschek **18**

Aspects of a Parallel Molecular Dynamics Software for Nano-Fluidics
Martin Bernreuther, Martin Buchholz, Hans-Joachim Bungartz **19**

Massively Parallel Quantum Computer Simulations: Towards Realistic Systems
Marcus Richter, Guido Arnold, Binh Trieu, Thomas Lippert **20**

Session "Performance Modeling and Analysis"

Distribution of Periscope Analysis Agents on ALTIX 4700
Michael Gerndt, Sebastian Strohhäcker **23**

Analysis of the Weather Research and Forecasting (WRF) Model on Large-Scale Systems.
Darren J. Kerbyson, Kevin J. Barker, Kei Davis **24**

Analytical Performance Models of Parallel Programs in Clusters
Diego R. Martínez, Vicente Blanco, Marcos Boullón, José Carlos Cabaleiro, Tomás F. Pena **25**

Computational Force: A Unifying Concept for Scalability Analysis
Robert W. Numrich **26**

Session "Image Reconstruction"

A Parallel Workflow for the Reconstruction of Molecular Surfaces
Daniele D'Agostino, Ivan Merelli, Andrea Clematis, Luciano Milanesi, Alessandro Orro **29**

HPC Simulation of Magnetic Resonance Imaging
Tony Stöcker, Kaveh Vahedipour, N. Jon Shah **30**

A Load Balancing Framework in Multithreaded Tomographic Reconstruction
José Antonio Álvarez, Javier Roca Piera, José Jesús Fernández **31**

Session "Parallel Algorithms"

Parallelisation of Block Recursive Matrix Multiplication in Prefix Computations
Michael Bader, Sebastian Hanigk, Thomas Huckle **35**

Parallel Exact Inference
Yinglong Xia, Viktor K. Prasanna **36**

Efficient Parallel String Comparison
Peter Krusche and Alexander Tiskin **37**

Session "Parallel Computing with OpenMP"

Implementing Data-Parallel Patterns for Shared Memory with OpenMP
Michael Suess, Claudia Leopold **41**

Generic Locking and Deadlock-Prevention with C++
Michael Suess, Claudia Leopold **42**

Parallelizing a Real-Time Steering Simulation for Computer Games with OpenMP
Bjoern Knafla, Claudia Leopold **43**

Session "Parallel Computing with FPGAs"

IANUS: Scientific Computing on an FPGA-Based Architecture
Francesco Belletti, Maria Cotallo, Andres Cruz, Luis Antonio Fernández, Antonio Gordillo, Andrea Maiorano, Filippo Mantovani, Enzo Marinari, Victor Martín-Mayor, Antonio Muñoz-Sudupe, Denis Navarro, Sergio Pérez-Gaviro, Mauro Rossi, Juan Jesus Ruiz-Lorenzo, Sebastiano Fabio Schifano, Daniele Sciretti, Alfonso Tarancón, Raffaele Tripiccione, Jose Luis Velasco **47**

Optimizing Matrix Multiplication on Heterogeneous Reconfigurable Systems
Ling Zhuo, Viktor K. Prasanna **48**

Session "Numerical Algorithms I"

Strategies for Parallelizing the Solution of Rational Matrix Equations
José M. Badía, Peter Benner, Maribel Castillo, Heike Faßbender, Rafael Mayo, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí **51**

A Heterogeneous Pipelined Parallel Algorithm for Minimum Mean Squared Error Estimation with Ordered Successive Interference Cancellation
Francisco-Jose Martínez-Zaldívar, Antonio. M. Vidal-Maciá, Alberto González **52**

Aitken-Schwarz Acceleration with Auxiliary Background Grids
Frank Hülsemann **53**

Session "Parallel Programming Models"

A Framework for Performance-Aware Composition of Explicitly Parallel Components
Christoph W. Kessler, Welf Löwe **57**

A Framework for Prototyping and Reasoning about Distributed Systems
Marco Aldinucci, Marco Danelutto, Peter Kilpatrick **58**

Formal Semantics Applied to the Implementation of a Skeleton-Based Parallel Programming Library
Joel Falcou, Jocelyn Sérot **59**

Session "Numerical Algorithms II"

OpenMP Implementation of the Householder Reduction for Large Complex Hermitian Eigenvalue Problems
Andreas Honecker, Josef Schüle **63**

Multigrid Smoothers on Multicore Architectures
Carlos García, Manuel Prieto, Fransisco Tirado **64**

Parallelization of Multilevel Preconditioners Constructed from Inverse-Based ILUs on on Shared-Memory Multiprocessors
José I. Aliaga, Matthias Bollhöfer, Alberto F. Martín, Enrique S. Quintana-Ortí **65**

Session "Parallel Data Distribution and I/O"

Optimization Strategies for Data Distribution Schemes in a Parallel File System
Jan Seidel, Rudolf Berrendorf, Ace Crngarov, Marc-André Hermanns **69**

Parallel Redistribution of Multidimensional Data
Tore Birkeland, Tor Sørevik **70**

Parallel I/O Aspects in $\mathrm{PIMA(GE)^2}$ **Lib**
Andrea Clematis, Daniele D'Agostino, Antonella Galizia **71**

Session "Parallel Automatic Differentiation"

Parallelism in Structured Newton Computations
Thomas F. Coleman, Wei Xu **75**

Automatic Computation of Sensitivities for a Parallel Aerodynamic Simulation
Arno Rasch, H. Martin Bücker, Christian H. Bischof **76**

Parallel Jacobian Accumulation
Ebadollah Varnik, Uwe Naumann **77**

Session "Scheduling"

Layer-Based Scheduling Algorithms for Multiprocessor-Tasks with Precedence Constraints
Jörg Dümmler, Raphael Kunis, Gudula Rünger **81**

Unified Scheduling of I/O- and Computation-Jobs for Climate Research Environments
N. Peter Drakenberg, Sven Trautmann **82**

Session "Performance Analysis I"

Analyzing Cache Bandwidth on the Intel Core 2 Architecture
Robert Schöne, Wolfgang E. Nagel, Stefan Pflüger **85**

Analyzing Mutual Influences of High Performance Computing Programs on SGI Altix 3700 and 4700 Systems with PARbench
Rick Janda, Matthias S. Müller, Wolfgang E. Nagel, Bernd Trenkler **86**

Low-level Benchmarking of a New Cluster Architecture
Norbert Eicker, Thomas Lippert **87**

Session "Biomedical Applications"

Parallel Ensemble of Decision Trees for Neuronal Sources Localization of the Brain Activity
Elena Popova **91**

Experimenting Grid Protocols to Improve Privacy Preservation in Efficient Distributed Image Processing
Antonella Galizia, Federica Viti, Daniele D'Agostino, Ivan Merelli, Luciano Milanesi, Andrea Clematis **92**

Efficient Parallel Simulations in Support of Medical Device Design
Marek Behr, Mike Nicolai, Markus Probst **93**

Session "Fault Tolerance"

Mitigating the Post-Recovery Overhead in Fault Tolerant Systems
Guna Santos, Angelo Duarte, Dolores Rexachs, Emilio Luque **97**

Towards Fault Resilient Global Arrays
Vinod Tipparaju, Manoj Krishnan, Bruce Palmer, Fabrizio Petrini, Jarek Nieplocha **98**

Using AOP to Automatically Provide Distribution, Fault Tolerance, and Load Balancing to the CORBA–$\mathcal{LC}$ Component Model
Diego Sevilla, José M. García, Antonio Gómez **99**

VirtuaLinux: Virtualised High-Density Clusters with no Single Point of Failure
Marco Aldinucci, Marco Danelutto, Massimo Torquati, Francesco Polzella, Gianmarco Spinatelli, Marco Vanneschi, Alessandro Gervaso, Manuel Cacitti, Pierfrancesco Zuccato **100**

Session "Performance Analysis II"

Comparative Study of Concurrency Control on Bulk-Synchronous Parallel Search Engines
Carolina Bonacic, Mauricio Marin **103**

Gb Ethernet Protocols for Clusters: An OpenMPI, TIPC, GAMMA Case Study
Stylianos Bounanos, Martin Fleury **104**

Performance Measurements and Analysis of the BlueGene/L MPI Implementation
Michael Hofmann, Gudula Rünger **105**

Potential Performance Improvement of Collective Operations in Current UPC Implementations
Rafik A. Salama, Ahmed Sameh **106**

Session "MHD and Turbulence Simulation"

Massively Parallel Simulations of Solar Flares and Plasma Turbulence
Lukas Arnold, Christoph Beetz, Jürgen Dreher, Holger Homann, Christoph Schwarz, Rainer Grauer **109**

Object-Oriented Programming and Parallel Computing in Radiative Magneto-hydrodynamics Simulations
Vladimir Gasilov, Sergei D'yachenko, Olga Olkhovskaya, Alexei Boldarev, Elena Kartasheva, Sergei Boldyrev **110**

Parallel Simulation of Turbulent Magneto-hydrodynamic Flows
Axelle Viré, Dmitry Krasnov, Bernard Knaepen, Thomas Boeck **111**

Pseudo-Spectral Modeling in Geodynamo
Maxim Reshetnyak, Bernhard Steffen **112**

Session "Parallel Tools and Middleware"

Design and Implementation of a General-Purpose API of Progress and Performance Indicators
Ivan Rodero, Francesc Guim, Julita Corbalan, Jesus Labarta **115**

Efficient Object Placement including Node Selection in a Distributed Virtual Machine
Jose M. Velasco, David Atienza, Katzalin Olcoz, Francisco Tirado **116**

Memory Debugging of MPI-Parallel Applications in Open MPI
Rainer Keller, Shiqing Fan, Michael Resch **117**

Session "Image Processing and Visualization"

Lessons Learned Using a Camera Cluster to Detect and Locate Objects
Daniel Stødle, Phuong Hoai Ha, John Markus Bjørndalen, Otto J. Anshus **121**

Parallel Morphological Neural Networks for Hyperspectral Image Classification on Fully Heterogeneous and Homogeneous Networks of Workstations
Javier Plaza, Antonio Plaza, Rosa Pérez, Pablo Martínez **122**

Hybrid Parallelization for Interactive Exploration in Virtual Environments
Marc Wolter, Marc Schirski, Torsten Kuhlen **123**

Session "Fluid Dynamics Simulation"

Parallelisation of a Geothermal Simulation Package: A Case Study on Four Multicore Architectures
Andreas Wolf, Volker Rath, H. Martin Bücker **127**

A Lattice Gas Cellular Automata Simulator with Cell Broadband Engine^{TM}
Yusuke Arai, Ryo Sawai, Yoshiki Yamaguchi, Tsutomu Maruyama, Moritoshi Yasunaga **128**

Session "Hyperscalable Applications"

Massively Parallel All Atom Protein Folding in a Single Day
Abhinav Verma, Srinivasa M. Gopal, Alexander Schug, Jung S. Oh, Konstantin V. Klenin, Kyu H. Lee, Wolfgang Wenzel **131**

Simulations of QCD in the Era of Sustained Tflop/s Computing
Thomas Streuer, Hinnerk Stüben **132**

Optimizing Lattice QCD Simulations on BlueGene/L
Stefan Krieg **133**

Mini-Symposium
"The Future of OpenMP in the Multi-Core Era"

OpenMP 3.0
J. Mark Bull **137**

OpenMP for Clusters
Larry Meadows **138**

Getting OpenMP Up to Speed
Ruud van der Pas **139**

PerfOMP: A Runtime Performance Monitoring API for OpenMP
Van Bui, Oscar Hernandez, Barbara Chapman, Rick Kufrin, Danesh Tafti, Pradeep Gopalkrishnan **140**

Affinity Matters! OpenMP on Multicore and ccNUMA Architectures
Dieter an Mey, Christian Terboven **141**

Mini-Symposium "Scaling Science Applications on Blue Gene"

Turbulence in Laterally Extended Systems
Jörg Schumacher, Matthias Pütz **145**

Large Simulations of Shear Flow in Mixtures via the Lattice Boltzmann Equation
Kevin Stratford, Jean Christophe Desplat **146**

Simulating Materials with Strong Correlations on BlueGene
Erik Koch **147**

DL_POLY_3: Parallel Performance and Large Scale Simulations
Ilian T. Todorov **148**

Massively Parallel Simulation of Cardiac Electrical Wave Propagation on Blue Gene
Jeffrey J. Fox, Gregery T. Buzzard, Robert Miller, Fernando Siso-Nadal **149**

Petascale Atmospheric General Circulation Models for CCSM
Henry M. Tufo **150**

Blue Gene/P: The Next Generation Enabling Breakthrough Simulation Based Engineering and Science
Kirk E. Jordan **151**

Mini-Symposium "Scalability and Usability of HPC Programming Tools"

Benchmarking the Stack Trace Analysis Tool for BlueGene/L
Gregory L. Lee, Dong H. Ahn, Dorian C. Arnold, Bronis R. de Supinski, Barton P. Miller, Martin Schulz **155**

Scalable, Automated Performance Analysis with TAU and PerfExplorer
Kevin A. Huck, Allen D. Malony **156**

Developing Scalable Applications with Vampir
Matthias S. Müller, Holger Brunst, Matthias Jurenz, Andreas Knüpfer, Wolfgang E. Nagel **157**

Scalable Collation and Presentation of Call-Path Profile Data with CUBE
Markus Geimer, Björn Kuhlmann, Farzona Pulatova, Felix Wolf, Brian Wylie **158**

Coupling DDT and Marmot for Debugging of MPI Applications
Bettina Krammer, Valentin Himmler, David Lecomber **159**

Compiler Support for Efficient Profiling and Tracing
Oscar Hernandez, Barbara Chapman **160**

Comparing Intel Thread Checker and Sun Thread Analyzer
Christian Terboven **161**

Continuous Runtime Profiling of OpenMP Applications
Karl Fürlinger, Shirley Moore **162**

Understanding Memory Access Bottlenecks on Multi-core
Josef Weidendorfer **163**

Mini-Symposium "DEISA: Extreme Computing in an Advanced Supercomputing Environment"

DEISA: Enabling Cooperative Extreme Computing in Europe
Victor Alessandrini **167**

Effective Methods for Accessing Resources in a Distributed HPC Production System
Andrea Vanni **168**

GPFS - A Cluster File System
Klaus Gottschalk **169**

Development Strategies for Modern Predictive Simulation Codes
Alice Koniges, Robert Anderson, Aaron Fisher, Brian Gunney, Nathan Masters **170**

Submission Scripts for Scientific Simulations on DEISA
Gavin J. Pringle, Terry M. Sloan, Elena Breitmoser, Odysseas Bournas and Arthur S. Trew **171**

Application Enabling in DEISA: Hyperscaling of Turbulence Codes Supporting ITER
Hermann Lederer, Reinhard Tisma, Roman Hatzky, Alberto Bottino, Frank Jenko **172**

First Principles Simulations of Plasma Turbulence within DEISA
Frank Jenko, Alberto Bottino, Tobias Görler, and Emanuele Poli **173**

Heavy Particle Transport in Turbulent Flows
Alessandra S. Lanotte, Luca Biferale, Jérémie Bec, Massimo Cencini, Stefano Musacchio, Federico Toschi **174**

Membranes Under Tension: Atomistic Modeling of the Membrane-Embedded Synaptic Fusion Complex
Marc Baaden **175**

Mini-Symposium "Parallel Computing with FPGAs"

Parallel Computing with Low-Cost FPGAs: A Framework for COPACOBANA
Tim Güneysu, Christof Paar, Jan Pelzl, Gerd Pfeiffer, Manfred Schimmler, Christian Schleiffer **179**

Accelerating the Cube Cut Problem with an FPGA-Augmented Compute Cluster
Tobias Schumacher, Enno Lübbers, Paul Kaufmann, Marco Platzner **180**

A Run-Time Reconfigurable Cache Subsystem
Fabian Nowak, Rainer Buchty, Wolfgang Karl **181**

Novel Brain-Derived Algorithms Scale Linearly with Number of Processing Elements
Jeff Furlong, Andrew Felch, Jayram Moorkanikara Nageswaran, Nikil Dutt, Alex Nicolau, Alex Veidenbaum, Ashok Chandrashekar, Richard Granger **182**

Programmable Architectures for Realtime Music Decompression
Martin Botteck, Holger Blume, Jörg von Livonius, Martin Neuenhahn, Tobias G. Noll **183**

The HARWEST High Level Synthesis Flow to Design a Special-Purpose Architecture to Simulate the 3D Ising Model
Alessandro Marongiu, Paolo Palazzari **184**

Towards an FPGA Solver for the PageRank Eigenvector Problem
Séamas McGettrick, Dermot Geraghty, Ciarán McElroy **185**

Invited Talks

Maria Ramalho-Natario	Tuesday, 4 Sept.	10:00 am
Barbara Chapman	Wednesday, 5 Sept.	9:30 am
Marek Behr	Thursday, 6 Sept.	9:30 am
Satoshi Matsuoka	Friday, 7 Sept.	9:30 am

European E-Infrastructure: Promoting Global Virtual Research Communities

Maria Ramalho-Natario

European Commission, INFSO
Information Society and Media DG
Unit F3 "GÉANT and e-Infrastructure"
1049 Brussels
Belgium
E-mail: maria.ramalho-natario@ec.europa.eu

Abstract

The Framework Programme 7, through the specific Programme 'Capacities', will ensure that support to existing research infrastructures (e.g. high speed inter-networks, computing grids and digital repositories) will continue and will help to create new research infrastructures of pan-European interest. This talk will focus on the vision for FP7 of the development of ICT-based infrastructures, also named e-Infrastructure, insisting on the aims of the Calls in 2007 and early 2008 to support virtual global research communities. For the benefit of the audience, a parenthesis will be made on the plans for the creation of the European High-Performance Computing service in FP7.

Programming in the Multicore Era

Barbara Chapman

University of Houston, Texas
Department of Computer Science
Houston, TX 77204-3475
United States of America
E-mail: chapman@cs.uh.edu

Abstract

Dual-core machines are now actively marketed for desktop and home computing. Systems with a larger number of cores exist, and more are planned. Some cores are capable of executing multiple threads. At the very high end, programmers need to design codes for execution by thousands of processes or threads and have begun to consider how to write programs that can scale to hundreds of thousands of threads. Clearly, the future is multi- and many-core, as well as many-threaded. In the past, most application developers could rely on Moore's Law to provide them with steady performance improvements. But we have entered an era in which they may have to expend considerable effort if their codes are to exploit the processing power offered by next-generation platforms.

Existing shared memory parallel programming APIs were not necessarily designed for general-purpose computing or with many threads in mind. Distributed memory paradigms do not necessarily allow the expression of fine-grained parallelism or provide full exploitation of architectural features. The fact that threads share some resources in multicore systems makes it hard to reason about the impact of program modifications on performance and results may be surprising. Will programmers be able to use multicore platforms effectively?

In this presentation, we discuss the challenges posed by multicore technology. We then review recent work on programming languages that are potentially interesting for multicore platforms, and discuss on-going activities to extend compiler technology in ways that may help the multicore programmer.

Simulation of Heart-Assist Devices

Marek Behr

RWTH Aachen University
Chair for Computational Analysis of Technical Systems
52074 Aachen
Germany
E-mail: behr@cats.rwth-aachen.de

Abstract

Parallel computing is enabling computational engineering analyses of unprecedented complexity to be performed. This talk reports on parallel finite element flow simulations supporting the development of implantable ventricular assist devices in the form of continuous-flow axial pumps. These pumps offer simplicity and reliability needed in long-term clinical applications. Their design however poses continuing challenges, such as high shear stress levels, flow stagnation and onset of clotting, and loss of pump efficiency.

Elevated shear stress levels are particularly evident in mechanical biomedical devices. One of the adverse responses of the red blood cells to elevated shear is hemolysis, dependent on both dose and time. The distribution of the shear stress levels in a complex flow field in a rotary blood pump chamber as well as the duration of the blood cells' exposure to these pathological conditions are largely unknown. Device designers are often compelled to make decisions about the details of pump configuration guided only by the global, time- and space-averaged, indicators of the shear stress inside the pump, such as the hemolysis observations made on the exiting blood stream. This challenge of detailed analysis and reduction of shear stress levels while maintaining pump efficiency as well as the need to pinpoint any persistent stagnation areas in the flow field motivates our current computational work.

We describe the flow simulation methodology and apply it to the problem of anylysis of blood flow in an axial ventricular assist device, the MicroMed DeBakey LVAD. This pump consists of the flow straightener, a six-bladed impeller, and a six-bladed diffuser inside a cylindrical housing. The simulations must explore a range of impeller speed and various pressure conditions. The computations are performed on an IBM Blue Gene.

Using these simulations as an illustration, we will focus on the architecture of the MPI-based finite element code, and on steps taken to ensure reasonable parallel speed-up on 4096 CPUs, including performance analysis and bottleneck identification. In view of the need for design optimization, where unsteady flow fields as well as their sensitivities with respect to the design parameters must be computed repeatedly while seeking a minimum of an flow-dependent objective function, the use of thousands of CPUs is a critical factor that makes such optimization practical.

Towards Petascale Grids as a Foundation of E-Science

Satoshi Matsuoka

Tokyo Institute of Technology
The NAREGI Project
National Institute of Informatics
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550
Japan
E-mail: matsu@is.titech.ac.jp

Abstract

While there is general consensus that computing platforms underlying the grid infrastructures will continue to evolve, variance in the speed of technology acceleration in HPC is causing many of the assumptions made in the early days of grid to no longer hold. Such divergence in the metrics, as well as wider proliferation of related technologies such as Web2.0, will be changing the optimal design of the overall grid infrastructure towards more centralization in the data/computing centers, as we also have experienced in the past for the Internet. Still, some facilities will remain fundamentally distributed, as simple centralization in one location might not be feasible for various reasons. Based on our recent experiences with our TSUBAME supercomputer, which is currently Asia-Pac's fastest machine according to the Top500, and its next petascale generation design thereof, we will discuss the future design of multi-petascale grids with such machines being constituent massive resource nodes instead of vast distribution.

Session

“Electronic Structure Simulation”

Tuesday, September 4, 2007
11:30 to 12:30

Domain Decomposition for Electronic Structure Computations

Guy Bencteux[1,3], Maxime Barrault[1], Eric Cancès[3], William Hager[2], and Claude Le Bris[3]

[1] EDF-R&D,
Dept SINETICS, 92141 Clamart, France
E-mail: {guy.bencteux, maxime.barrault}@edf.fr

[2] Department of Mathematics, University of Florida
Gainesville FL 32611-8105, USA
E-mail: hager@math.ufl.edu

[3] 6 & 8, avenue Blaise Pascal, Cité Descartes
77455 Marne-La-Vallée Cedex 2, France
E-mail: {cances, lebris}@cermics.enpc.fr

Abstract

We present here an application of parallel computing in electronic structure computations. Concerned fields are materials science, chemistry and biology, where numerical simulation with quantum models is nowadays an ubiquituous tool.

When they are discretized, most often via a development on a Galerkin basis, these models go through the computation of an orthogonal projector, called the density matrix, size N_b, onto a subspace whose dimension is equal to the number of electrons in the system, N, with $N < N_b$. Most of the existing codes computes this subspace through a basis made of eigenvectors of the so called Fock (or Kohn-Sham) matrix. This step has a $O(N^3)$ scaling and constitutes one of the bottleneck of quantum simulations.

This work presents recent results given by the newly introduced[1], Multilevel Domain Decomposition method (MDD) that scales linearly with N and is based on domain decomposition paradigm.

The parallel implementation fully exploit the geographical decomposition of the problem. A Single Process Multiple Data model have been implemented, where each processor executes a single instance of the algorithm, with additional data structure containing the information needed from the contiguous parts of the problem. Major part of the communications is done between neighbouring processors, so communications are not a bottleneck towards a good scalability.

Results are presented on real hydrocarbons systems with up to 2 millions atoms, on various computing environment from laboratory cluster to Blue Gene/L machine.

1. M. Barrault, E. Cancès, W. W. Hager, and C. Le Bris. *Multilevel domain decomposition for electronic structure calculations*, J. Comp. Phys. **222**, 86–119, (2007).

Scalable Dynamic Adaptations for Electronic Structure Calculations

Masha Sosonkina

Ames Laboratory and Iowa State University,
Ames, IA, 50011,USA
E-mail: masha@scl.ameslab.gov

Abstract

Applications augmented with adaptive capabilities are becoming common in parallel computing environments which share resources, such as main memory, network, or disk I/O. For large-scale scientific applications, dynamic algorithmic adjustments to certain computationally intensive parts may facilitate efficient execution of the entire application when the availability of the computational resources changes. Application-specific knowledge, often best revealed during the run-time, is required to select and initiate algorithmic adaptations. In particular, General Atomic and Molecular Electronic Structure System (GAMESS) used for *ab-initio* molecular quantum chemistry calculations has two different implementations, *conventional* and *direct*, of the Self-Consistent Field (SCF) method, which is an iterative procedure to find an approximate solution to the Schrödinger equation. The conventional implementation may lead to a faster convergence. It stores large data sets defining integrals, and thus is viewed as I/O intensive. In contrast, the direct implementation may be preferred on computing platforms with thousands of processing elements since it recomputes integrals "on-the-fly".

To accomplish the dynamic switching between direct and conventional implementations, the middleware NICAN is integrated with GAMESS using only a few application source code changes. NICAN is a light-weight tool that decouples performance analysis and decision making from the application execution and invokes application adaptation functions in a timely manner. Adaptation decisions are based on the GAMESS execution time for the current as well as previous SCF iterations. For example, if the I/O channel contention starts to interfere with the conventional SCF for a certain molecule size—as seen, e.g., by a large jump in the SCF execution time—, GAMESS is prompted to switch to the direct SCF implementation at the next iteration. In general, any application-specific adaptation policy may be attached to NICAN as the *control* module via a specific control interface (port). The structure of the control port and its implementation for GAMESS are described in the paper. Dynamic adaptations may be combined with static adaptations based on characteristics of the underlying architecture, such as the number of cores per processor or I/O capabilities. Tests, performed on up to 128 processors for molecules of different sizes, show that, in the presence of the I/O resource contention, the performance of parallel GAMESS enhanced with adaptive algorithm changes may be improved substantially.

Session

“Parallel Performance Tools”

Tuesday, September 4, 2007
11:30 to 12:30

Visualizing Parallel Functional Program Runs: Case Studies with the Eden Trace Viewer

Jost Berthold and Rita Loogen

Philipps-Universität Marburg, Fachbereich Mathematik und Informatik
Hans Meerwein Straße, D-35032 Marburg, Germany
E-mail: {berthold, loogen}@informatik.uni-marburg.de

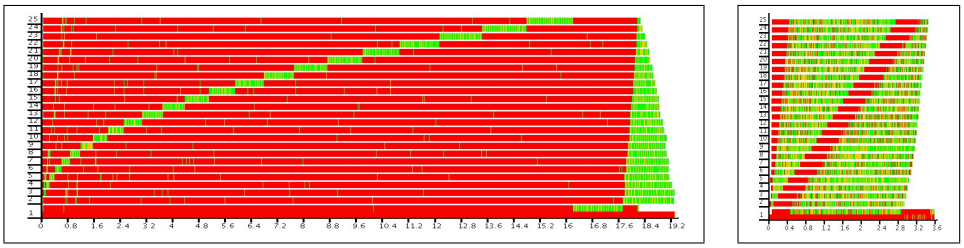
Abstract

Parallel functional languages offer a highly abstract view of parallelism. While less error-prone, this sometimes hampers program optimisations. The large gap between abstract language concepts and concrete runtime behaviour needs customized *high-level* tools for runtime analysis. This paper describes the Eden Trace Viewer (*EdenTV*), a post-mortem trace analysis and visualisation tool for the parallel functional language Eden[a]. EdenTV shows program executions in terms of Eden's abstract units of computation instead of providing a machine-oriented low level view like common tools for parallelism analysis do. The Eden runtime system writes selected *events*, into a trace file processed by EdenTV. Eden programs need not be changed to obtain the traces.

Case Study: Missing Demand. When using a lazy computation language, a crucial issue is to start the evaluation of needed subexpressions early enough and to fully evaluate them for later use. Evaluation strategies must then be applied to certain sub-results. On the sparse basis of runtime measurements, such an optimisation would be rather cumbersome. The EdenTV, accompanied by code inspection, makes inefficiencies obvious.

Example: (Warshall's algorithm) This algorithm computes shortest paths for all nodes of a graph from the adjacency matrix with a ring of processes. Each process computes the minimum distances from one node to every other node. The trace visualisations in Figure 1 show EdenTV's *Processes* view for two versions of the program on a Beowulf cluster, with an input graph of 500 nodes (aggregated on 25 processors). The programs differ by a single line which introduces additional demand for an early update on the local row.

[a] `http://www.mathematik.uni-marburg.de/~eden`



Runtime: 19.37 sec.
without additional demand control

Runtime: 3.62 sec.
with additional demand control

Figure 1. Warshall-Algorithm (500 node graph)

Automatic Phase Detection of MPI Applications

Marc Casas, Rosa M. Badia, and Jesús Labarta

Barcelona Supercomputing Center (BSC),
Technical University of Catalonia (UPC),
Campus Nord, Modul C6, Jordi Girona, 1-3, 08034 Barcelona, Spain
E-mail: {mcasas, rosa}@ac.upc.edu, jesus@cepba.upc.es

Abstract

In the last years, the performance of parallel platforms has increased amazingly. Thus, the study of the execution of applications in these platforms has become a hard and tedious work because the analysts have to spend a lot of time studying data about computation and communication periods of time. For studying a complete timestamped sequence of events of an application, that is, a tracefile of the whole application, a huge tracefile will be required. The size of these files can reach easily 10 or 20 Gigabytes. It is impossible to handle this amount of data with tools like Paraver[1]. As well as the problem of the size, the analysis of such tracefiles has another important problem: often, some parts of the trace are perturbed, so the analysis of these parts is misleading. A third problem is the identification of the most representative regions of the tracefile.

The analyst is then forced to control very carefully the process of tracing the application, enabling this process in the moments he wants to study and disabling it when he is not interested. The analyst must limit as much as possible the number of events of the tracefile (hardware counters, instrumented routines, etc...). This process is very large and tedious and the analyst must have a knowledge about the source code of the application he is studying.

For these reasons, several authors believe that the development and utilization of trace based techniques is not useful. However, techniques based on tracefiles allow the analyst to do a very detailed study of the variations on space (set of processes) and time that could affect notably the performance of the application. For that reason, is necessary to develop techniques that allow the analyst to handle large event traces.

The first goal of this paper is to use signal processing techniques (wavelet transform) in order to provide a very fast automatic detection of the phases of MPI applications' execution. The criterion of such signal processing techniques in order to perform the phase detection is to separate regions according to their frqeuency behavior, i. e., a region with a small iteration which is repeated many times will be separated from another region with no peridic behavior. The second goal is to use the information derived from signal processing techniques to acquire remarkable conclusions about the scalability of the applicatons and how could be improved. An important point of our work is that it enables the analyst to acquire some information without requiring knowlegde of the source code of the application. Our tool makes possible to obtain a very fast report of the general characteristics of the application's execution and, for that reason, makes easy the work of performance analyst.

1. `http://www.cepba.upc.es/Paraver/`

Session

“Particle and Atomistic Simulation”

Tuesday, September 4, 2007
14:00 to 16:00

Load Balanced Parallel Simulation of Particle-Fluid DEM-SPH Systems with Moving Boundaries

Florian Fleissner and Peter Eberhard

Institute of Engineering and Computational Mechanics,
University of Stuttgart, 70550 Stuttgart, Germany
E-mail: {fleissner, eberhard}@itm.uni-stuttgart.de

Abstract

We propose a new pure Lagrangian method for the parallel load balanced simulation of particle-fluid systems with moving boundaries or free surfaces. Our method is completely meshless and modells solid objects as well as the fluid as particles. By an Orthogonal Recursive Bisection we obtain a domain decomposition that is well suited for a PI-controller based load balancing. This controller approach is designed for being used on clusters of workstations as it can cope with load imbalances not only as emerging from the simulation dynamics but also from competitive processes of other users. In this paper we present the most important concepts such as neighborhood search, particle interactions, domain decomposition and controller based load balancing.

Communication and Load Balancing of Force-Decomposition Algorithms for Parallel Molecular Dynamics

Godehard Sutmann[1] **and Florian Janoschek**[2]

[1] John von Neumann Institute for Computing (NIC),
Central Institute for Applied Mathematics (ZAM)
Research Centre Jülich (FZJ)
D-52425 Jülich, Germany
E-mail: g.sutmann@fz-juelich.de

[2] Stuttgart University
Institute for Computational Physics, Pfaffenwaldring 27
D - 70569 Stuttgart, Germany
E-mail: FJanoschek@gmx.net

Abstract

Classical molecular dynamics simulations are often considered as the *method par excellence* to be ported to parallel computers, promising a good scaling behavior. This is at least true when considering homogeneous particle distributions which interact via short range interaction potentials. For long range interactions there exist efficient methods, which are, however, very much more involved to be brought onto parallel computers. For systems, which have non-homogenous particle distributions, the load on each processor is likely to become unbalanced, due to deviations in spatial particle distributions, resulting in different numbers of interactions which have to be calculated on each processor. The result is often a strong reduction of parallel scalability of the program.

In the present talk, a new approach is presented, which combines the philosophies of force-[1] and domain-decomposition[2] methods. The basic entity is the influence matrix of the system, giving information about which particle pairs interact with each other. Geometric proximity between particles is achieved via sorting them according to space filling curves. Load-balancing is achieved via partitioning the influence matrix in such a way that the work for each processor becomes approximately the same. Test calculations are presented, which indeed proove the better performance of the new method especially for inhomogenous systems.

1. R. Murty and D. Okunbor, *Efficient parallel algorithms for molecular dynamics simulations*, Parallel Comp., **25**, 217–230, (1999).
2. S. J. Plimpton, *Fast parallel algorithms for short-range molecular dynamics*, J. Comp. Phys. **117**, 1–19, (1995).

Aspects of a Parallel Molecular Dynamics Software for Nano-Fluidics

Martin Bernreuther[1], Martin Buchholz[2], and Hans-Joachim Bungartz[2]

[1] Höchstleistungsrechenzentrum Stuttgart,
70550 Stuttgart, Germany
E-mail: bernreuther@hlrs.de

[2] Institut für Informatik, Technische Universität München,
85748 Garching, Germany
E-mail: {buchholm, bungartz}@in.tum.de

Abstract

We have developed a program system for the simulation of fluid flow on the nano-scale in the field of process engineering. Different aspects of this systems (wich has been implemented using C++ and MPI) are shown. They are dealing with software engineering, datastructures for efficient access to parameters and parallelisation.

Fluid-flow simulations usually involve a large number of relatively small molecules. Our main focus is on multiphase systems for which the particle distribution is usually very heterogeneous (nucleation processes, e.g.) The systems we are simulating can be modelled using rigid molecular models assembled from sites with non-bonded short-range pair potentials. Each of the sites is described by a set of parameters which are required for the calculation of interactions between sites of the same type. For the interaction of unequal sites, mixed parameter sets have to be calculated. This has to be done for each possible pair of sites. We describe an approach to precalculate and store those mixed parameter sets in a stream, which allows efficient access and gives the flexibility to add new site types easily.

Another focus of our work has been on software engineering techniques. Using the adapter design pattern, we achieved a complete decoupling of the physical parts of the simulation (e.g. molecule models and interactions) from the data structures and the parallelisation. This eases the further concurrent development of the software and reduces the complexity of the different modules. It also gives us the opportunity to swap modules in a plug-in like fashion.

Finally, we demonstrate the advantages of a "pair ownership" of processes for the parallelisation which allows the joint calculation of macroscopic values and the forces on molecules.

Massively Parallel Quantum Computer Simulations: Towards Realistic Systems

Marcus Richter, Guido Arnold, Binh Trieu, and Thomas Lippert

Central Institute for Applied Mathematics,
Research Centre Jülich, D-52425 Jülich, Germany
E-mail: {m.richter, g.arnold, b.trieu, th.lippert}@fz-juelich.de

Abstract

Quantum computers have become of great interest primarily due to their potential of solving certain computationally hard problems such as factoring integers and searching databases faster than a conventional computer. Candidate technologies for realizing quantum computers include trapped ions, atoms in QED cavities, Josephson junctions, nuclear or electronic spins, quantum dots, and molecular magnets.

All these technologies have to overcome one main difficulty: decoherence – the loss of quantum information due to interaction with the environment. Therefore, quantum computers will need error correction, which requires at least several tens of qubits and the ability to perform hundreds of gate operations. This imposes a number of strict requirements, and narrows down the list of candidate physical systems. Simulating numbers of qubits in this range is important to numerically test the scalability of error correction codes and fault tolerant quantum computing schemes and their robustness to errors typically encountered in realistic quantum computer architectures.

For this reason, we have extended the *Massively Parallel Quantum Computer Simulator*[1] by a gate level error model which covers operational errors and decoherence. Applying this error model to the Quantum Fourier Transformation and Grover's quantum search algorithm, one finds that the QFT circuit is more robust to operational inaccuracies than Grover's algorithm on comparable scales. Critical parameters can be derived which give a first estimate of tolerable error thresholds.

At present ion traps are regarded as the most promising technology for the realization of quantum computers due to the long coherence time of trapped ions. We discuss Hamiltonian based dynamical ion-trap simulations which have been developed in collaboration with the experimental working group of Prof. Rainer Blatt. In contrast to standard approches no approximations like the rotating wave approximation or an expansion in the Lamb-Dicke parameter are required which allow for very accurate simulations. This permits to indentify critical system parameters which limit the stability of the experiment.

1. K. De Raedt, K. Michielsen, H. De Raedt, B. Trieu, G. Arnold, M. Richter, Th. Lippert, H. Watanabe and N. Ito, *Massively Parallel Quantum Computer Simulator*, Comp. Phys. Comm. **176**, 121–136 (2007).

Session

“Performance Modeling and Analysis”

Tuesday, September 4, 2007
14:00 to 16:00

Distribution of Periscope Analysis Agents on ALTIX 4700

Michael Gerndt, Sebastian Strohhäcker

Technische Universität München, Fakultät für Informatik I10
Boltzmannstr.3, 85748 Garching, Germany
E-mail: gerndt@in.tum.de

Abstract

Performance analysis tools help users in writing efficient codes for current high performance machines. Since the architectures of today's supercomputers with thousands of processors expose multiple hierarchical levels to the programmer, program optimization cannot be performed without experimentation.

Performance analysis tools can provide the user with measurements of the program's performance and thus can help him in finding the right transformations for performance improvement. Since measuring performance data and storing those data for further analysis in most tools is not a very scalable approach, most tools are limited to experiments on a small number of processors.

Periscope[1] is the first distributed online performance analysis tool. It consists of a set of autonomous agents that search for performance properties. Each agent is responsible for analyzing a subset of the application's processes and threads. The agents request measurements of the monitoring system, retrieve the data, and use the data to identify performance properties. This approach eliminates the need to transport huge amounts of performance data through the parallel machine's network and to store those data in files for further analysis.

The focus of this paper is on the distribution of application processes and analysis agents in Periscope. Large scale experiments with Periscope are executed in form of batch jobs where in addition to the processors for the application additional processors are allocated for the analysis agents. The number of additional processors is currently decided by the programmer. If the analysis agents were overloaded, the programmer might decide to use more processors for the analysis in a next experiment.

During startup of the experiment, Periscope determines the mapping of application processes and analysis agents to the processors. It is the goal, to place analysis agents near to the controlled processes to reduce the communication overhead. This paper describes the concepts and the implementation used on the ALTIX 4700 supercomputer at LRZ for placement.

1. M. Gerndt, K. Fürlinger, and E. Kereku, *Advanced techniques for performance analysis*, in: Parallel Computing: Current & Future Issues of High-End Computing, Proc. ParCo 2005, G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata, (Eds.), NIC Series **33** ISBN 3-00-017352-8, pp. 15–26 (2006).

Analysis of the Weather Research and Forecasting (WRF) Model on Large-Scale Systems.

Darren J. Kerbyson, Kevin J. Barker, and Kei Davis

Performance and Architecture Lab (PAL),
Los Alamos National Laboratory,
Los Alamos, NM USA
E-mail: {djk, kjbarker, kei.davis}@lanl.gov

Abstract

In this work we analyze the performance of the Weather Research and Forecasting (WRF) model using both empirical data and an analytic performance model.

The Weather Research and Forecasting (WRF) model is a community mesoscale numerical weather prediction system with nearly 5,000 users, developed by a consortium of government agencies and the research community. It is used for both operational forecasting research and atmospheric research, and is capable of modeling events such as storm systems and hurricanes. Features of WRF include dynamical cores based on finite difference methods and many options for physical parameterizations.

WRF has been ported to various platforms and can utilize thousands of processors in parallel. Future computational requirements are expected to increase as a consequence of both increased resolution and the use of increasingly detailed physics models. The performance model of WRF that we have developed allows us to accurately predict the performance of WRF on near-future large-scale systems that may contain many hundreds of thousands of processors.

In this work we analyze the performance of the current version of WRF (version 2.2) on two very different large-scale systems: a cluster of 256 Opteron nodes (1,024 processing cores) interconnected by 4x SDR Infiniband, and a small Blue Gene/L system containing 1,024 nodes (2,048 processing cores) interconnected by a proprietary 3-D torus. This comparison allows us to draw conclusions concerning the system sizes required to achieve an equivalent level of performance on WRF. We then develop a performance model of WRF that is validated against these two systems and that exhibits high prediction accuracy. The model is used to compare larger-sized configurations of the two systems that cannot currently be measured. Finally, we compare the predicted performance of comparison between possible future configurations of Blue Gene and Infiniband/Opteron clusters.

An important aspect of this work is the capture of key performance characteristics into an analytical performance model. This model is parameterized in terms of the main application inputs (iteration count, number of grid points in each dimension, the time step per iteration, etc.) as well as system parameters (processor count, communication topology, latencies and bandwidths, etc.). The model also takes as input the time per cell when using all processing cores in a single node—this can be measured on an available system, or determined for a future system using a processor simulator. The utility of the model is its capability of predicting for larger-scale systems that are not available for measurement, and for accurate prediction for future or hypothetical systems.

Analytical Performance Models of Parallel Programs in Clusters

Diego R. Martínez[1], Vicente Blanco[2], Marcos Boullón[1], José Carlos Cabaleiro[1], and Tomás F. Pena[1]

[1] Dept. of Electronics and Computer Science
University of Santiago de Compostela, Spain
E-mail: {diegorm, marcos, caba, tomas}@dec.usc.es

[2] Dept. de Statistics and Computer Science
La Laguna University, Spain
E-mail: vicente.blanco@ull.es

Abstract

This paper presents a framework based on an user driven methodology to obtain analytical models on parallel systems and, in particular, clusters. The proposed framework uses both measurement and analytical modeling and provides an easy to use tool that allow the analyst to obtain analytical models to characterize the performance of parallel applications in clusters. Analytical models are obtained by a statistical analysis of measurements from real parallel executions. The behavior of an application can be characterized in terms of parameters such as the problem size, the number of process or the effective network capacity.

The framework consists of two interconnected stages. The first stage is devoted to instrumentation, where information about the performance of the execution of the parallel application is monitored and stored. This stage is based on CALL, which is a profiling tool for interacting with the code in an easy, simple and direct way.

The second stage uses this information to obtain an analytical model by means of statistical analysis. This stage is based on R, a well known language and environment for statistical analysis. Specific R functions were developed to process the data from multiple executions of a CALL experiment and to automatically perform analytical models of CALL experiments by means of an iterative fitting process. These functions are grouped into modules, so any change in a module produces a minimal impact on the others, and the capabilities of the analysis environment can be easily extended.

Three different parallel versions of the matrix product are used to show the automatic fit process, and the accuracy of the obtained models is compared with an algorithmic complexity study of the selected examples. The analytical models obtained provide accurate models as those based on a theoretical complexity analysis of the source.

Computational Force: A Unifying Concept for Scalability Analysis

Robert W. Numrich

Minnesota Supercomputing Institute
University of Minnesota
Minneapolis, MN 55455 USA
E-mail: rwn@msi.umn.edu

Abstract

Computational force, also called computational intensity, is a unifying concept for understanding the performance of parallel numerical algorithms. Dimensional analysis reduces a formula for execution time, from a paper by Stewart, to an exercise in differential geometry for a single efficiency surface. Different machines move on the surface along different paths defined by curvilinear coordinates that depend on ratios of hardware forces to software forces.

Session

“Image Reconstruction”

Tuesday, September 4, 2007
16:30 to 18:00

A Parallel Workflow for the Reconstruction of Molecular Surfaces

Daniele D'Agostino[1], Ivan Merelli[2], Andrea Clematis[1], Luciano Milanesi[2], and Alessandro Orro[1]

[1] Institute for Applied Mathematics and Information Technologies, National Research Council
Via De Marini 6, 16149 Genova, Italy
E-mail: {dago, clematis}@ge.imati.cnr.it

[2] Institute for Biomedical Technologies, National Research Council
Via Fratelli Cervi 93, 20090 Segrate (MI), Italy
E-mail: {ivan.merelli, luciano.milanesi, alessandro.orro}@itb.cnr.it

Abstract

The modeling of molecular surfaces and their visualization is assuming an increasing importance in many fields of Bioinformatics. One example is the study of molecule-molecule interactions, usually referred as molecular docking, that represents one of the subject of our research. In the paper we present a parallel workflow for the reconstruction of high resolution molecular surfaces, with the aim to develop a high performance docking screening system. The original contribution is represented by the high performance approach to the molecular surface reconstruction process and, in perspective, to molecular docking.

The workflow is made up by five operations: Grid Generation, Connolly Correction, Median Filter, Isosurface Extraction and Simplification. The input is represented by the atomic coordinates of a molecule, the outputs are both its volumetric description and the isosurface that, on the basis of the user selection, corresponds to the Van der Waals, Lee & Richards or Connolly surface.

The main feature of the parallel implementation of the workflow is the efficient production of high resolution surfaces. The experiments show an efficiency of 0.5 for the whole pipeline on a beowulf clusters. This is an important result considering that this value is achieved taking into account I/O operations and the whole workflow stages.

It is worthwhile to note that the use of the simplification operation permits the production of high quality surfaces with different levels of detail. In fact all the morphological information of the irregular zones of the surface, that are the most interesting ones, are preserved with great accuracy, while it is possible to greatly reduce the number of triangles in the other parts of the mesh. For molecular docking the use of simplified surfaces represents an important advantage, because such kind of analysis has a cost proportional to the size of the surfaces to process.

HPC Simulation of Magnetic Resonance Imaging

Tony Stöcker, Kaveh Vahedipour, and N. Jon Shah

Institute of Medicine,
Research Centre Jülich, 52425 Jülich, Germany
E-mail: {t.stoecker, k.vahedipour, n.j.shah}@fz-juelich.de

Abstract

High performance computer (HPC) simulations provide helpful insights to the process of magnetic resonance image (MRI) generation, e.g. for general MRI pulse sequence design[1] and optimisation, MRI artefact detection, validation of experimental results, hardware optimisation, MRI sample development and for education purposes. This abstract presents the recently developed simulator JEMRIS (Jülich Environment for Magnetic Resonance Imaging Simulation). JEMRIS is developed in C++ and the message passing is realised with the MPI library. In contrast to previous approaches known from the literature, JEMRIS provides generally valid numerical solutions for the case of classical MR spin physics governed by the Bloch equations.

A new object-oriented design pattern for the rapid implementation of MRI sequences was developed. Here the sequence is defined by a set of interacting objects inserted into a left-right ordered tree structure. The framework provides a GUI for the rapid development of the MRI sequence tree, which is stored in XML format and parsed into a C++ object for the simulation routines. Thus, the set-up of MR imaging sequences relying on the predefined modules is easily performed without any programming.

The parallel HPC implementation allows the treatment of huge spin ensembles resulting in realistic MR signals. On small HPC clusters, 2D simulations can be obtained in the order of minutes, whereas for 3D simulations it is recommended to use higher scaling HPC systems. The simulator can be applied to MRI research for the development of new pulse sequences. The framework already serves as a tool in research projects, as will be shown for the important example of multidimensional spatially selective excitation. The impact of these newly designed pulses in real experiments has to be tested in the near future. Further, the rapidly evolving field of medical image processing might benefit of a gold standard, serving as a testbed for the application of new methods and algorithms. Here, JEMRIS could provide a general framework for generating standardised and realistic MR images for many different situations and purposes.

1. E. M. Haacke, R. W. Brown, M. R. Thompson, and R. Venkatesan, *Magnetic Resonance Imaging: Physical Principles and Sequence Design*, Wiley & Sons, (1999).

A Load Balancing Framework in Multithreaded Tomographic Reconstruction

José Antonio Álvarez, Javier Roca Piera, and José Jesús Fernández

Departamento de Arquitectura de Computadores y Electrónica
Universidad de Almería, 04120 Almería, Spain
E-mail: {jaberme, jroca, jose}@ace.ual.es

Abstract

Hiding latencies as well as getting an optimal assignment of processors, are two issues for many scientific applications, specially if non dedicated clusters are used. Traditionally, high performance scientific applications are parallelized using MPI libraries. Typical implementations of MPI minimize dynamic features required to face latencies or shared resource usage. Switching from the MPI process model to a threaded programming model in the parallel environment, can help to achieve efficient overlapping and provide abilities for load balancing.

$BICAV$, our research group's tomographic reconstruction software, was ported to a multithreaded environment and provided with load balancing capabilities. In the design of a parallel and multithreaded strategy for applications related to image processing, such as 3D tomographic image reconstruction algorithms[1], data distributions have to be carefully devised so locality is preserved as much as possible. Our tomographic reconstruction software exhibits strong data dependences and hence emphasizes the influence of data locality preservation on the performance. The load balancing strategy presented, RakeLB (inspired on Rake[2] algorithm), seizes the facilities provided by AMPI [3](a framework where MPI processes can be embedded into user level threads). RakeLB aims to preserve data locality when dynamic workload reassignments are needed.

Through experiments, we illustrated how our application was directly benefited due to an optimal exploitation of concurrence, in its threaded version, in contrast to the MPI version which did not perform so well. The performance of this technique was analyzed and the dynamic load balancing strategy, RakeLB, which preserves data locality, was proved to be efficient for applications like $BICAV$.

1. A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, SIAM Society for Industrial and Applied Mathematics, (2001).
2. C. Fonlupt, P. Marquet, and J. L. Dekeyser, *Data-parallel load balancing strategies*, Parallel Computing **24**, 1665–1684, (1998)
3. Ch. Huang, O. Lawlor, and L. V. Kale, *Adaptive MPI*, Proceedings of the 16th International Workshop on Languages and Compilers for Parallel Computing (LCPC 03) , 306–322, (2003)

Session

“Parallel Algorithms”

Tuesday, September 4, 2007
16:30 to 18:00

Parallelisation of Block Recursive Matrix Multiplication in Prefix Computations

Michael Bader, Sebastian Hanigk, and Thomas Huckle

Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, 85748 Garching, Germany,
E-mail: {*bader, hanigk, huckle*}*@in.tum.de*

Abstract

We present a study on computational efficiency and scalability of the parallelisation of a matrix multiplication problem that occurs within an optimal-control-based quantum compiler. During the respective iterative optimisation scheme, the evolutions of the quantum system are computed at given time steps. The quantum evolutions are modelled by a sequence of complex-valued matrices, for which we need to compute all matrix products $A_1 \cdots A_k$ for given matrices $A_1, \ldots, A_m$ (so-called *prefix problem*).[2]

For parallelisation, both a coarse-grain and a fine-grain approach is available. The coarse grain approach computes partial matrix products as intermediate values, which are combined via a tree-structured algorithm (*parallel prefix* scheme). In contrast, a pure fine-grain approach only parallelises the subsequent matrix multiplications. Both approaches have inherent bottlenecks. Distribution of matrix multiplication to too many processors leads to small matrix blocks, where the respective serial implementations (usually BLAS routines) cannot reach their full speed. On the other hand, only a limited number of processors can be used for the parallel prefix scheme, and the scheme contains certain communication bottlenecks. In addition, it considerably increases the total computational work. Hence, we study a hybrid approach that combines these two parallelisation techniques.

We compare two different block-structured approaches to parallelise matrix multiplication: (1) The SRUMMA algorithm, which combines block-structured matrix multiplication with clever prefetching of matrix blocks to hide communication costs, and (2) a block-recursive scheme based on Peano curves[1] that in addition utilizes the space-filling curve's locality properties for parallelisation. We present performance studies on an Infiniband cluster with 128 processors. The Peano-curve approach proved to be superior especially for moderately sized matrices on many processors. As a result, the prefix problem may be parallelised by preferring parallel matrix multiplications over using the tree-like parallel prefix scheme. This is in contrast to a previous result[2], where a less scalable algorithm for parallel matrix multiplication was used.

1. M. Bader, C. Zenger, *Cache oblivious matrix multiplication using an element ordering based on a Peano curve*, Linear Algebra and its Applications **417** 2–3, (2006).
2. T. Gradl, A. Spörl, T. Huckle, S. J. Glaser, and T. Schulte-Herbrüggen, *Parallelising Matrix Operations on Clusters for an Optimal Control-Based Quantum Compiler*, in: Euro-Par 2006, Parallel Processing, LNCS **4128**, pp. 751–762, (2006).

Parallel Exact Inference

Yinglong Xia[1] **and Viktor K. Prasanna**[2]

[1] Computer Science Department,
University of Southern California, Los Angeles, U.S.A.
E-mail: yinglonx@usc.edu

[2] Ming Hsieh Department of Electrical Engineering,
University of Southern California, Los Angeles, U.S.A.
E-mail: prasanna@usc.edu

Abstract

In this paper, we present complete message-passing implementation that shows scalable performance while performing exact inference on arbitrary Bayesian networks. Our work is based on a parallel version of the classical technique of converting a Bayesian network to a junction tree before computing inference. We propose a parallel algorithm for constructing potential tables for a junction tree and explore the parallelism of rerooting technique for multiple evidence propagation. Our implementation also uses pointer jumping for parallel inference over the junction tree. For an arbitrary Bayesian network with n vertices using p processors, we show an execution time of $O(nk_m^2 + n^2w + (nw^2 + wN \log n + r^w wN + r^w N \log N)/p)$, where w is the clique width, r is the number of states of the random variables, k is the maximum node degree in the Bayesian network, k_m is the maximum node degree in the moralized graph and N is the number of cliques in the junction tree. Our implementation is shown to be scalable for $1 \leq p \leq n$ for moralization and clique identification, and $1 \leq p \leq N$ for junction tree construction, potential table construction, rerooting and evidence propagation. We have implemented the parallel algorithm using MPI on state-of-the-art clusters and our experiments show scalable performance.

Efficient Parallel String Comparison

Peter Krusche and Alexander Tiskin

Department of Computer Science
The University of Warwick, Coventry CV4 7AL, United Kingdom
E-mail: {peter, tiskin}@dcs.warwick.ac.uk

Abstract

The longest common subsequence (LCS) problem is a classical method of string comparison. Several coarse-grained algorithms for the LCS problem have been proposed in the past. However, none of these algorithms achieve scalable communication. In this paper, we propose the first coarse-grained LCS algorithm with scalable communication. Moreover, the algorithm is work-optimal, synchronisation-efficient, and solves a more general problem of semi-local string comparison, improving in at least two of these aspects on each of the predecessors.

Session

“Parallel Computing with OpenMP”

Tuesday, September 4, 2007
16:30 to 18:00

Implementing Data-Parallel Patterns for Shared Memory with OpenMP

Michael Suess and Claudia Leopold

University of Kassel, Research Group Programming Languages / Methodologies,
Wilhelmshöher Allee 73, D-34121 Kassel, Germany
E-mail: {msuess, leopold}@uni-kassel.de

Abstract

With the advent of multi-core processors, parallel programming for shared memory architectures is entering the mainstream. However, parallel programming in general is considered difficult. One solution to this problem lays in the use of libraries that hide parallelism from the programmer.

Although there are a variety of libraries available both commercially and in a research stage, what is missing from the picture is a pattern library in OpenMP. Design patterns implemented in OpenMP should be a viable learning aid to beginners in the field of concurrent programming. At the same time, they are able to encapsulate parallelism and hide it from the programmer if required. For this reason, the AthenaMP project[1] was created that implements various parallel programming patterns in C++ and OpenMP.

The main goal of AthenaMP is to provide implementations for a set of concurrent patterns, both low-level patterns like advanced locks (not described here), and higher-level patterns like the data-parallel patterns described in this paper. The patterns demonstrate solutions to parallel programming problems, as a reference for programmers, and additionally can be used directly as generic components. The code is also useful for compiler vendors testing their OpenMP implementations against more involved C++-code. A more extensive project description is provided by one of the authors in his weblog[a].

This paper reports on the implementations of several data-parallel patterns: `modify_each`, `transmute`, `combine`, `reduce`, and `filter`. We describe the patterns as well as our experiences implementing them in C++ and OpenMP. The results of two benchmarks showing no performance losses when compared to a pure OpenMP implementation are also included.

Another contribution of this paper is a description of implementation problems that we faced while developing our patterns: the insufficient state of the compilers with regard to OpenMP and C++, as well as the restricted ability to influence the scheduling of parallel loops at runtime.

1. M. Suess, *AthenaMP*, `http://athenamp.sourceforge.net/`, (2006).

[a] http://www.thinkingparallel.com/2006/11/03/a-vision-for-an-openmp-pattern-library-in-c/

Generic Locking and Deadlock-Prevention with C++

Michael Suess and Claudia Leopold

University of Kassel, Research Group Programming Languages / Methodologies,
Wilhelmshöher Allee 73, D-34121 Kassel, Germany
E-mail: {msuess, leopold}@uni-kassel.de

Abstract

Locks are one of the most important building blocks of concurrent programming today. As with the advent of multi-core processors, parallel programming starts to move into the mainstream, the problems associated with locks become more visible, especially with higher-level languages like C++. This paper addresses the following ones:

- lock initialization and destruction is not exception-safe and very C-ish. Lock destruction is forgotten frequently.
- setting and releasing locks is not exception-safe and C-ish, as well. Unsetting Locks may be forgotten in complicated code-paths with a lot of branches.
- deadlocks are possible when using multiple locks

To solve the first two problems, a common C++ idiom called *RAII* is used. RAII stands for *Resource Acquisition is Initialization* and combines acquisition/initialization and release/destruction of resources with construction and destruction of variables. Our solution to the first problem is called lock adapter and has the effect that locks are initialized and destroyed in constructors and destructors, respectively. Our solution for the second problem is already well known as guard objects or *scoped locking* and means that locks are set and unset in constructors and destructors, respectively.

The third problem is solved in two ways: first by extending the guard objects to multiple locks and internally choosing the locking order in a deterministic way, and second by introducing so-called leveled locks that enable the creation and automatic control of a lock hierarchy[1] that detects possible deadlocks at runtime.

This paper presents our solutions for the problems sketched above, along with simple benchmark results and the implementation problems we have encountered. Although the examples use OpenMP, the functionality presented is mostly independent of the parallel programming system used, as long as it meets certain conditions.

This work is part of the AthenaMP open source parallel pattern library[2]. Its main goal is to provide implementations for a set of concurrent patterns (both low-level patterns like advanced locks, and higher-level ones) using OpenMP and C++.

1. A. S. Tanenbaum, *Modern Operating Systems*. Prentice Hall, 2nd edition, (2001).
2. M. Suess, *AthenaMP*, `http://athenamp.sourceforge.net/`, (2006).

Parallelizing a Real-Time Steering Simulation for Computer Games with OpenMP

Bjoern Knafla and Claudia Leopold

University of Kassel, Research Group Programming Languages / Methodologies,
Wilhelmshoeher Allee 73, 34121 Kassel, Germany
E-mail: {bknafla, leopold}@uni-kassel.de

Abstract

Future computer games need parallel programming to exploit multi-core game consoles and PCs. To gain first insights into the parallelization of legacy game-like programs, we parallelized the C++ application OpenSteerDemo[1] with OpenMP. This paper reports on our experiences, including guidelines to help parallelizing legacy game codes.

OpenSteerDemo runs a main loop as it is typical for games. Each iteration simulates one time step, for which it first collects user input, then updates the state of the game world (update stage), and finally renders and displays this state (graphics stage). In the original sequential program, the update stage cycles through all agents. For each agent, the new state (position, velocity, direction) is computed based on the agent's previous state and information on the local environment (state of neighbor agents, nearby obstacles etc.). Right after that, the agent's state is updated.

To support parallelization, we divided the update stage into a simulation sub-stage and a modification sub-stage. The former computes the next state of each agent, but does not yet write this information. Since all accesses to the state are reads, the agents can be processed in parallel. A barrier separates the simulation sub-stage from the modification sub-stage, which accomplishes the updates, and can be executed in parallel, as well. The separation between the two sub-stages required extensive refactoring of the code to prevent unintended accesses to global variables.

Furthermore, the update stage is decoupled from the non-thread-safe (in the present implementation sequential) graphics stage by so-called render feeders. Render feeders collect state updates in a thread-specific storage, with no need for locks. Another parallelization problem we had to face were random numbers. In our parallel program, each agent has its own random number generator, which enables deterministic simulations, independent from the number of threads and the scheduling of agents to threads.

These changes required major refactorization of the code, but the modified program has a simple parallel structure and needs only two barriers for synchronization. Performance measurements on a dual-processor dual-core computer (4 threads) showed speedups of up to 2.84 for a main loop iteration (including sequential parts), and up to 3.54 for the parallel update stage.

1. C. W. Reynolds, *OpenSteer website*, `http://opensteer.sourceforge.net` (2004).

Session

“Parallel Computing with FPGAs”

Wednesday, September 5, 2007
11:00 to 12:30

IANUS: Scientific Computing on an FPGA-Based Architecture

Francesco Belletti[1,2], **Maria Cotallo**[3,4], **Andres Cruz**[3,4], **Luis Antonio Fernández**[5,4], **Antonio Gordillo**[6,4], **Andrea Maiorano**[1,4], **Filippo Mantovani**[1,2], **Enzo Marinari**[7], **Victor Martín-Mayor**[5,4], **Antonio Muñoz-Sudupe**[5,4], **Denis Navarro**[8,9], **Sergio Pérez-Gaviro**[3,4], **Mauro Rossi**[10], **Juan Jesus Ruiz-Lorenzo**[6,4], **Sebastiano Fabio Schifano**[1,2], **Daniele Sciretti**[3,4], **Alfonso Tarancón**[3,4], **Raffaele Tripiccione**[1,2], **and Jose Luis Velasco**[3,4]

[1] Dipartimento di Fisica, Università di Ferrara, I-44100 Ferrara (Italy)

[2] INFN, Sezione di Ferrara, I-44100 Ferrara (Italy)

[3] Departamento de Física Teórica, Facultad de Ciencias, Universidad de Zaragoza, 50009 Zaragoza (Spain)

[4] Instituto de Biocomputación y Física de Sistemas Complejos (BIFI), 50009 Zaragoza (Spain)

[5] Departamento de Física Teórica, Facultad de Ciencias Físicas, Universidad Complutense, 28040 Madrid (Spain)

[6] Departamento de Física, Facultad de Ciencia, Universidad de Extremadura, 06071, Badajoz (Spain)

[7] Dipartimento di Fisica, Università di Roma "La Sapienza", I-00100 Roma (Italy)

[8] Departamento de Ingenieria Electrónica y Comunicaciones, Universidad de Zaragoza, CPS, Maria de Luna 1, 50018 Zaragoza (Spain)

[9] Instituto de Investigación en Ingenieria de Aragón (I3A), Universidad de Zaragoza, Maria de Luna 3, 50018 Zaragoza (Spain)

[10] ETH Lab - Eurotech Group, I-33020 Amaro (Italy)

Abstract

This paper describes the architecture and FPGA-based implementation of a massively parallel processing system (IANUS), carefully tailored to the computing requirements of a class of simulation problems relevant in statistical physics. We first discuss the system architecture in general and then focus on the configuration of the system for Monte Carlo simulation of spin-glass systems. This is the first large-scale application of the machine, on which IANUS achieves impressive performance. Our architecture uses large-scale on chip parallelism ($\simeq 1000$ computing cores on each processor) so it is a relevant example in the quickly expanding field of many-core architectures.

Optimizing Matrix Multiplication on Heterogeneous Reconfigurable Systems

Ling Zhuo and Viktor K. Prasanna

Ming Hsieh Department of Electrical Engineering,
University of Southern California,
Los Angeles, USA
E-mail: {lzhuo, prasanna}@usc.edu

Abstract

With the rapid advances in technology, FPGAs have become an attractive option for acceleration of scientific applications. In particular, reconfigurable computing systems have been built which combine FPGAs and general-purpose processors to achieve high performance. Previous work assumes the nodes in such systems are homogeneous, containing both processors and FPGAs. However, in reality, the nodes can be heterogeneous, based on either FPGAs, processors, or both. In this paper, we model these heterogeneous reconfigurable systems using various parameters, including the computing capacities of the nodes, the size of memory, the memory bandwidth, and the network bandwidth. Based on the model, we propose a design for matrix multiplication that fully utilizes the computing capacity of a system and adapts to various heterogeneous settings. To illustrate our ideas, the proposed design is implemented on Cray XD1. Heterogeneous nodes are generated by using only the FPGAs or the processors in some nodes. Experimental results show that our design achieves up to 80% of the total computing capacity of the system and more than 90% of the performance predicted by the model.

Session

“Numerical Algorithms I”

Wednesday, September 5, 2007
11:00 to 12:30

Strategies for Parallelizing the Solution of Rational Matrix Equations

José M. Badía[1], Peter Benner[2], Maribel Castillo[1], Heike Faßbender[3], Rafael Mayo[1], Enrique S. Quintana-Ortí[1], and Gregorio Quintana-Ortí[1]

[1] Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I, 12.071–Castellón, Spain
E-mail: {badia, castillo, mayo, quintana, gquintan}@icc.uji.es.

[2] Fakultät für Mathematik, Technische Universität Chemnitz, 09107 Chemnitz, Germany
E-mail: benner@mathematik.tu-chemnitz.de.

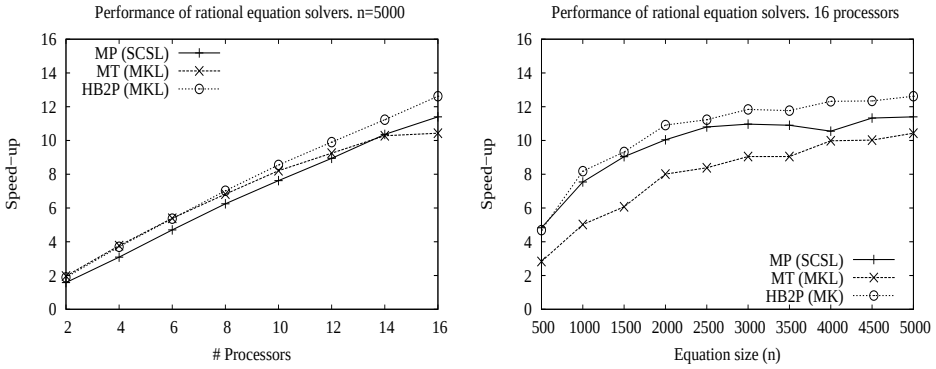
[3] Technische Universität Braunschweig, Institut Computational Mathematics, 38106 Braunschweig, Germany
E-mail: h.fassbender@tu-bs.de

Abstract

In this paper we compare different strategies to parallelize numerical methods. We have applied those strategies to solve the nonlinear rational matrix equation $X = Q + LX^{-1}L^T$, where $Q \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^{n \times n}$, and $X \in \mathbb{R}^{n \times n}$ is the sought-after solution, via a structure-preserving doubling algorithm (SDA). This equation arises in the analysis of stationary Gaussian reciprocal processes.

The parallelization strategies attack the problem at three levels of granularity: The MP algorithm computes all operations in the SDA using the parallel routines in ScaLAPACK. The MT algorithm is a "serial" code that uses BLAS and LAPACK, and extracts all parallelism from a multithreaded implementation of BLAS. The HB2P algorithm employs two processes that perform concurrently two groups of operations of the SDA. Each process invokes the kernels from a multithreaded version of BLAS.

The following results were obtained on a SGI Altix 350 with 16 Intel Itanium2 processors. The Figure shows the speed-up of the parallel implementations of the SDA measured with respect to an optimized sequential implementation of the SDA.



All the parallel algorithms are implemented using standard sequential and parallel linear algebra libraries and MPI, ensuring the portability of the codes.

A Heterogeneous Pipelined Parallel Algorithm for Minimum Mean Squared Error Estimation with Ordered Successive Interference Cancellation

Francisco-Jose Martínez-Zaldívar[1], **Antonio. M. Vidal-Maciá**[2], **and Alberto González**[1]

[1] Departamento de Comunicaciones, Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
E-mail: {fjmartin, agonzal}@dcom.upv.es

[2] Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
E-mail: avidal@dsic.upv.es

Abstract

This paper describes a pipelined parallel algorithm for Minimum Mean Square Error estimation (MMSE) with an Ordered Successive Interference Cancellation (OSIC) decoding procedure. The sequential algorithm is based on the square root version of the Information Filter used as a RLS solver. Its cost is compared with a square root Kalman Filter based algorithm, with better results. The parallel algorithm is suitable to be executed in either a distributed memory or a shared memory architecture multiprocessor, and in either a homogeneous or heterogeneous parallel system with high efficiency.

1. G. J. Foschini, *Layered space-time architecture for wireless communications in a fading environment when using multiple antennas*, Bell Labs Technical Journal **1**, 41–59, (1996).
2. B. Hassibi, *An efficient square-root algorithm for BLAST*, IEEE International Conference on Acoustics, Speech and Signal Processing **2**, 737–740, (2000)
3. Yang-Seok Choi, Peter J. Voltz, and Frank A. Cassara, *On channel estimation and detection for multicarrier signals in fast and selective Rayleigh fading channels*, IEEE Transactions on Communications **49**, , (2001)
4. Hufei Zhu, Zhongding Lei, and Francois P. S. Chin, *An improved square-root algorithm for BLAST*, IEEE Signal Processing Letters **11(9)**, , (2004)
5. Ali H. Sayed and Thomas Kailath, *A state-space approach to adaptive RLS filtering*, IEEE Signal Processing Magazine **11(3)**, 18–60, (1994)
6. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA, (1996)
7. V. Kumar, A. Gram, A. Gupta, and G. Karypis, *An Introduction to Parallel Computing: Design and Analysis of Algorithms*, Addison-Wesley, Harlow, UK, (2003).

Aitken-Schwarz Acceleration with Auxiliary Background Grids

Frank Hülsemann

EDF R&D,
1, Avenue du Général de Gaulle, 92141 Clamart, France
E-mail: frank.hulsemann@edf.fr

Abstract

The ease with which Schwarz domain decomposition methods allow existing single-threaded solvers to be (re-)used for parallel computations is undoubtedly one of their advantages. However, in order to be practical for large scale parallel computations, the slow convergence of the standard Schwarz algorithm has to be overcome by some form of acceleration.

We present the Aitken-Schwarz method which is an example of an extrapolation based acceleration technique which achieves fast convergence with low additional communication requirements. These properties are particularly desirable in metacomputing environments such as clusters of clusters up to clusters of supercomputers. The parallel performance of the approach has been demonstrated[1] by experiments involving three supercomputers on two continents. Since then, the research in the area of Aitken-Schwarz methods has shifted to algorithmic development in order to extend the scope of the approach[3].

The original Aitken-Schwarz method developed by Garbey and Tromeur-Dervout[2] was motivated by Fourier transformations on regular Cartesian grids. The method has since been extended to different types of grids such as general tensor product grids or Chimera-type overset grids. In this presentation, we show how to adapt the Aitken-Schwarz method to grids with refined stripes, as a step on the way to locally refined grids.

We recall the basic form of the Aitken-Schwarz algorithm for the concrete case of a linear, separable, second order differential operator on two subdomains. Then we show how auxiliary grids can reduce the computational cost of the setup phase and give numerical results illustrating the convergence behaviour of the proposed scheme.

1. N. Barberou, M. Garbey, M. Hess, M. Resch, T. Rossi, J. Toivanen, D. Tromeur-Dervout, *Aitken-Schwarz method for efficient metacomputing of elliptic equations*, in: Domain Decomposition Methods in Science and Engineering, I. Herrera, D. E. Keyes, O. B. Widlund, R. Yates, (eds.), UNAM, 349–356, (2003).
2. M. Garbey, D. Tromeur-Dervout, *Two Level Domain Decomposition for Multiclusters*, in: 12th Int. Conf. on Domain Decomposition Methods DD12, T. Chan, T. Kako, H. Kawarada, O. Pironneau, (eds), 325–340, (2001).
3. M. Garbey, *Acceleration of the Schwarz method for elliptic problems*, SIAM J. of Scientific Computing **26**, 1871–1893, (2005).

Session

“Parallel Programming Models”

Wednesday, September 5, 2007
14:00 to 15:30

A Framework for Performance-Aware Composition of Explicitly Parallel Components

Christoph W. Kessler[1] **and Welf Löwe**[2]

[1] PELAB, IDA, Linköpings universitet, S-58183 Linköping, Sweden, *E-mail: chrke@ida.liu.se*

[2] MSI, Växjö universitet, Växjö, Sweden, *E-mail: welf.lowe@msi.vxu.se*

Abstract

We present a novel framework for performance-aware static and dynamic composition of explicitly parallel software components in a SPMD environment such as Fork[1] or MPI where component functions can be called by groups of processors operating in parallel.

A component provider makes components performance-aware by adding metacode that enables, at component deployment time, an optimizing composition tool to predict expected execution times of the implemented component functions. Different components may declare to implement the same functionality, such that composition can choose, for each call, between multiple variants that have different performance behavior for different problem sizes and numbers of executing processors. Moreover, components containing parallel compositions of independent parallel subtasks that involve component calls, e.g. where exploiting nested parallelism or parallel divide-and-conquer, may profit from optimized scheduling as time information is available.

Optimized static composition is applicable where problem sizes are statically known; it amounts to finding an optimal schedule for a parallel task graph consisting of variant malleable tasks. Optimized dynamic composition results in a table-driven implementation that, for each parallel call of a performance-aware component, looks up the expected best implementation variant, given the current problem sizes and processor group size. Likewise, the expected best processor allocation and schedule are looked up at parallel compositions. The dispatch tables are computed off-line at component deployment time by an interleaved dynamic programming algorithm.

We provide a proof-of-concept implementation of dynamic composition for different sorting components written in Fork[1] and run on the SBPRAM[1] cycle-accurate simulator. The experimental results demonstrate significant performance gains for the table-driven composed implementation compared to the original component implementations. We discuss possible application scenarios and raise issues for future work.

Our approach could be used with any modular parallel programming environment that provides the interface concept. If dynamic composition is applied to an object-oriented environment, our framework actually constitutes a generalization of dynamic virtual method dispatch that is guided by performance metadata.

1. J. Keller, Ch. Kessler, and J. Träff, *Practical PRAM Programming*, Wiley Interscience, (2001).

A Framework for Prototyping and Reasoning about Distributed Systems

Marco Aldinucci[1], **Marco Danelutto**[1], **and Peter Kilpatrick**[2]

[1] Dept. Computer Science – University of Pisa – Italy
E-mail: {aldinuc, marcod}@di.unipi.it

[2] Dept. Computer Science – Queen's University Belfast – United Kingdom
E-mail: p.kilpatrick@qub.ac.uk

Abstract

The design of distributed systems is challenging and becomes more so with the emergence of systems, such as grids, whose architecture may change dynamically. Ideally, one would develop such systems in a framework that would allow construction of a high-level design of the target system, reasoning about the performance of the system (including comparison with alternative designs) and, finally, provide support for (semi-)automatic generation of implementation code from the design. The work described here targets such a framework.

It builds on earlier work in which we proposed the use of Misra and Cook's Orc notation together with lightweight reasoning as a foundation for such an approach[1]; and on an elaboration of that work to augment Orc specifications with metadata characterising non-functional aspects of a target system, thus providing a means to allow reasoning, to a certain degree, about runtime performance[2]. The addition of metadata permits the user to make assumptions about the design that would not be possible in the general case. This, together with a semi-formal style of reasoning drawing extensively upon insight and experience, allows significant analysis of systems at relatively small expense.

The contribution described here extends the earlier work by (i) providing a worked example of the use of metadata to allow communication cost comparison between two designs for a classical distributed computing scenario; and, (ii) introducing a tool for the generation of a distributed Java skeleton from an Orc specification. The user can complete the skeleton code produced by providing the functional (sequential) code implementing site and process internal logic, while the general orchestration logic, mechanisms, schedule, etc. are completely dealt with by the generated code. Finally, code for the example is generated and completed, and the cost comparison validated.

1. M. Aldinucci, M. Danelutto, and P. Kilpatrick, *Management in distributed systems: a semi-formal approach*, Proc. Euro-Par 2007 – Parallel Processing 13th Intl. Euro-Par Conference, LNCS No. 4641, Rennes (F), Springer, (2007).
2. M. Aldinucci, M. Danelutto, and P. Kilpatrick, *Adding metadata to Orc to support reasoning about grid programs*, Proc. CoreGRID Symposium 2007, pp. 205–214, Rennes (F), Springer, (2007). ISBN: 978-0-387-72497-3

This research is carried out under the FP6 NoE CoreGRID (EC Contract IST-2002-004265).

Formal Semantics Applied to the Implementation of a Skeleton-Based Parallel Programming Library

Joel Falcou and Jocelyn Sérot

LASMEA, UMR6602 UBP/CNRS, Campus des Cézeaux, 63177 Aubière, France.
E-mail: {joel.falcou, jocelyn.serot}@lasmea.univ-bpclermont.fr

Abstract

In a previous paper[1], we described QUAFF, a skeleton-based parallel programming library[2,3] which main originality is to rely on C++ *template* meta-programming[4,5] techniques to significantly reduce the overhead traditionally associated to object-oriented implementations of such libraries. The basic idea is to use the C++ *template* mechanism so that skeleton-based programs are actually run at compile-time and generate a new C+MPI code to be compiled and executed at run-time. The implementation mechanism supporting this compile-time approach to skeleton-based parallel programming was only sketched mainly because the operational semantics of the skeletons was not stated in a formal way, but "hardwired" in a set of complex meta-programs. As a result, changing this semantics or adding a new skeleton was difficult. In this paper, we give a formal model for the QUAFF skeleton system, describe how this model can efficiently be implemented using C++ meta-programming techniques and show how this helps overcoming the aforementioned difficulties. We also asses the impact of this implementation technique by measuring the overhead introduced by QUAFF on the completion time over hand-written C+MPI code for both single skeleton application and when skeletons are nested at arbitrary level. In both cases, we show that this overhead never exceeds 7%, contrary to other run-time implementations which overhead are usually far larger[6]. To our best knowledge, this work is the first to both rely on a formal approach to skeleton compilation while offering performances on par with hand-coded C+MPI implementations.

1. J. Falcou, J. Sérot, T. Chateau, and J.-T. Lapresté, *QUAFF: Efficient C++ Design for Parallel Skeletons*, Parallel Computing, **32**, 604–615, (2006).
2. M. Cole, *Algorithmic skeletons: structured management of parallel computation*, MIT Press, (1989).
3. M. Cole, *Bringing skeletons out of the closet: A pragmatic manifesto for skeletal parallel programming*, Parallel Computing, **3**, 389–406, (2004).
4. T. Veldhuizen, *Using C++ template metaprograms*, C++ Report, **7**, 36–43, (1995). Reprinted in *C++ Gems*, Stanley Lippman (ed.).
5. D. Abrahams and A. Gurtovoy, *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*, C++ in Depth Series, Addison-Wesley Professional, (2004).
6. H. Kuchen, *A skeleton library*, in: Euro-Par '02: Proc. 8th International Euro-Par Conference on Parallel Processing, London, pp. 620–629, Springer (2002).

Session

“Numerical Algorithms II”

Wednesday, September 5, 2007
14:00 to 15:30

OpenMP Implementation of the Householder Reduction for Large Complex Hermitian Eigenvalue Problems

Andreas Honecker[1] **and Josef Schüle**[2]

[1] Institut für Theoretische Physik, Georg-August-Universität Göttingen,
37077 Göttingen, Germany
E-mail: ahoneck@uni-goettingen.de

[2] Gauss-IT-Zentrum, Technische Universität Braunschweig, 38106 Braunschweig, Germany
E-mail: j.schuele@tu-bs.de

Abstract

The computation of the complete spectrum of a complex Hermitian matrix typically proceeds through a Householder step. If only eigenvalues are needed, this Householder step needs almost the complete CPU time. Here we report our own parallel implementation of this Householder step using different variants of C and OpenMP. As far as we are aware, this is the only existing parallel implementation of the Householder reduction for complex Hermitian matrices which supports packed storage mode. As an additional feature we have implemented checkpoints which allow us to go to dimensions beyond $100\,000$. We perform runtime measurements and show firstly that even in serial mode the performance of our code is comparable to commercial libraries and that secondly we can obtain good parallel speedup.

Multigrid Smoothers on Multicore Architectures

Carlos García, Manuel Prieto, and Fransisco Tirado

Dpto. de Arquitectura de Computadores y Automática
Universidad Complutense
28040 Madrid, Spain
E-mail: {*garsanca, mpmatias, ptirado*}*@dacya.ucm.es*

Abstract

We have addressed in this paper the implementation of red-black multigrid smoothers on high-end microprocessors. Most of the previous work about this topic has been focused on cache memory issues due to its tremendous impact on performance. In this paper, we have extended these studies taking *Multicore Processors (MCP)* into account. With the introduction of *MCP*, new possibilities arise, which makes a revision of the different alternatives highly advisable. We proposed an alternative mapping of these algorithms onto *MCP* systems that improves cooperation between threads. Performance results on an *Intel Core*^{TM}*2 Duo* based system reveal that our scheme can compete with and even outperform sophisticated schemes based on loop fusion and tiling transformations aimed at improving temporal locality.

Parallelization of Multilevel Preconditioners Constructed from Inverse-Based ILUs on on Shared-Memory Multiprocessors

José I. Aliaga[1], **Matthias Bollhöfer**[2], **Alberto F. Martín**[1], **and Enrique S. Quintana-Ortí**[1]

[1] Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I, 12.071–Castellón, Spain; *E-mail:* {*aliaga, martina, quintana*}*@icc.uji.es.*

[2] Institute of Computational Mathematics, TU-Braunschweig, D-38106 Braunschweig, Germany; *E-mail: m.bollhoefer@tu-braunschweig.de.*

Abstract

The solution of linear systems is ubiquitous in chemistry, physics, and engineering applications. When the coefficient matrix is large and sparse, iterative methods as, e.g., those based on Krylov subspaces, are traditionally employed. Among these methods, ILUPACK[a] (Incomplete LU decomposition PACKage) is a novel software package, based on approximate factorizations with improved accuracy, that contains routines to both compute and apply preconditioners using a Krylov subspace method.

In this paper we present an OpenMP parallel preconditioner based on ILUPACK. The first step in the development of a parallel preconditioner consists in splitting the process into tasks and identifying the dependencies among these. After that, tasks are mapped to threads deploying task pools, in an attempt to achieve dynamic load balancing while preserving the dependencies.

Tasks for the parallel algorithm and their dependencies can be identified by manipulating the elimination tree of the matrix permuted using a Multilevel Nested Dissection (MLND) ordering: If we condense each elimination subtree rooted at height $log_2(p)$, where p is the number of processors (threads), we obtain a set of tasks which is organized in a tree-like structure, with nodes representing tasks, and the ancestor-descendant relationship representing dependencies among them. This task tree defines a partition of the ordered matrix which identifies matrix blocks that can be factorized in parallel and matrix blocks whose factorization depends on other matrix computations.

Due to the properties of the MLND ordering, the major part of the computation is concentrated on the leaves of the task tree; therefore a good load balance in the computation of the tasks associated with the leaves is mandatory to achieve high parallel performance. Although MLND ordering performs a best-effort work, there are a few leaves that concentrate the major part of the computation. In order to attain a better load balance, our parallel algorithm splits those leaves with a higher weight into finer-grain tasks, and employs dynamic load balancing.

[a] `http://www.math.tu-berlin.de/ilupack`

Session

“Parallel Data Distribution and I/O”

Wednesday, September 5, 2007
16:00 to 17:30

Optimization Strategies for Data Distribution Schemes in a Parallel File System

Jan Seidel[1], **Rudolf Berrendorf**[1], **Ace Crngarov**[1], **and Marc-André Hermanns**[2]

[1] Department of Computer Science
University of Applied Sciences Bonn-Rhein-Sieg, 53754 St. Augustin, Germany
E-mail: mail@janseidel.net, {rudolf.berrendorf, ace.crngarov}@fh-bonn-rhein-sieg.de

[2] Central Institute for Applied Mathematics
Research Centre Jülich, 52425 Jülich, Germany
E-mail: m.a.hermanns@fz-juelich.de

Abstract

Parallel systems leverage parallel file systems to efficiently perform I/O to shared files. These parallel file systems utilize different client-server communication and file data distribution strategies to optimize the access to data stored in the file system. In many parallel file systems, clients access data that is striped across multiple I/O devices or servers. Striping, however, results in poor access performance if the application generates a different stride pattern.

This work analyzes optimization approaches of different parallel file systems and proposes new strategies for the mapping of clients to servers and the distribution of file data with special respect to strided data access. The main drawbacks of standard striping in parallel file systems are described that especially occur with complex nested-strided data distributions of clients. We present a new way of utilizing existing MPI file views to distribute file data based on application needs. This technique is implemented in a parallel file system for remote main memory, which is used for evaluation. In this evaluation we demonstrate the benefits of file data distributions based on client access patterns described by file views compared to striping distribution.

Parallel Redistribution of Multidimensional Data

Tore Birkeland and Tor Sørevik

Dept. of Mathematics, University of Bergen
Norway
E-mail: {*tore.birkeland, tor.sorevik*}*@math.uib.no*

Abstract

On a parallel computer with distributed memory, multidimensional arrays are usually mapped onto the nodes such that only one or more of the indexes become distributed. Global computation on data by traversing the reminding indexes may then be done without communication. However, when global computation is needed on all indexes redistribution of data is needed. When we have d-dimensional data, and $d > 2$, different choices of mapping the data onto an r-dimensional processor grid exists. Any integer $1 \leq r < d$ would be a feasible processor grid. Of course, the different mappings require different redistribution strategies.

In this paper we develop a redistribution algorithm which is a generalisation of the standard algorithm for $d = 2$ and $r = 1$. Our algorithm handles all allowable values of d and r, including the important case of $d = 3$ and $r = 2$ described by[1],[2].

We show by a complexity analysis and numerical experiments that while using a $1D$ processor grid is the most efficient for modest number of processors, using $2D$ processor grid has better scalability and hence work best for higher number of processors. As the current trends in hardware design put greater emphasis on scalability, we believe this to be a decisive feature for the real High Performance code of tomorrow.

Our implementation is made in C++ and MPI. MPI's Cartesian topology routines are used for setting up the communicator groups. The nitty, gritty index fiddling is hidden by using blitz++[3] and MPI datatypes to create the proper blocks of data to be sent and received.

1. M. Eleftheriou, B. G. Fitch, A. Rayshubskiy, T. J. C. Ward, and R. S. Germain, *Scalable framework for the 3D FFTs on the BlueGene/L supercomputer: Implementation and early performance measurements*, IBM J. Res. & Dev. **49**, 457–464, (2005).
2. A. Dubey and D. Tessera, *Redistribution strategies for portable parallel FFT: a case study*, Concurrency and Computation: Practice and Experience **13**, 209–220, (2001).
3. T. L. Veldhuizen, *Arrays in Blitz++*, in: Proc. 2nd International Scientific Computing in Object-Oriented Parallel Environments (ISCOPE'98), Springer, Heidelberg, (1998).

Parallel I/O Aspects in $\mathrm{PIMA(GE)^2}$ Lib

Andrea Clematis, Daniele D'Agostino, and Antonella Galizia

Institute for Applied Mathematics and Information Technologies,
National Research Council
Genoa, Italy
E-mail: {clematis, dago, antonella}@ge.imati.cnr.it

Abstract

The scientific evolution allows the analysis of different phenomena with great accuracy and this results in a growing production of data to process. During last years, parallel computing has become a feasible solution, but a critical point is the efficiency in the I/O management. The characterization of I/O overhead is very important to improve the performance of parallel applications[1], and the adoption of a parallel I/O in scientific applications is becoming a common practice. The image processing community has not completely appraised its use, although an increasing attention is paid to parallel computations.

In this paper we address these issues in parallel image processing applications. We developed $\mathrm{PIMA(GE)^2}$ Lib, the Parallel IMAGE processing GEnoa Library; it provides a robust implementation of the most common image processing low level operations. During the design of the library, we look for a proper logical organization in the execution of I/O operations. We fixed the I/O pattern in the data access, and we made several tests about the logical organization in I/O operations to determine the most efficient strategy to apply in $\mathrm{PIMA(GE)^2}$ Lib. To achieve this goal we compared a master slave approach implemented with MPI 1, and a parallel I/O using the functionalities of the MPI-IO provided by MPI 2. In particular we use the ROMIO implementation[2]. In both cases we tested the interaction with the most common file systems for parallel and distributed applications, that are Parallel Virtual File System[3], and Network File System[4], both open source. Also aspects related with data layout on disk and efficient data distribution to parallel processes, for 3D images analysis tasks, are considered.

This work represents an experimental study about the selection of a parallel I/O strategy. We show that MPI 2 parallel I/O, combined with PVFS 2, outperforms the other possibilities, providing a reduction of the I/O cost for parallel image processing applications. Furthermore, at the best of our knowledge, $\mathrm{PIMA(GE)^2}$ Lib is one of the few examples of image processing library where a parallel I/O is strategy is adopted.

1. J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, and A. White, *The Sourcebook of Parallel Computing*, Morgan Kaufmann, (2002).
2. ROMIO home page, `http://www.mcs.anl.gov/romio`
3. Parallel Virtual File System Version 2, `http://www.pvfs.org`
4. Sun Microsystems Inc., *NFS: Network File System Version 3 Protocol Specification*, Sun Microsystems Inc., Mountain View,CA, (1993).

Session

“Parallel Automatic Differentiation”

Wednesday, September 5, 2007
16:00 to 17:30

Parallelism in Structured Newton Computations

Thomas F. Coleman and Wei Xu

Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario, Canada. N2L 3G1
E-mail: tfcoleman@uwaterloo.ca
E-mail: wdxu@math.uwaterloo.ca

Abstracts

A structured computation is one that breaks down into a (partially ordered) straight-line sequence of (accessible) macro computational tasks. Many vector-valued functions, representing expensive computation, are also structured computations. Exploiting this structure, a Newton step computation can expose useful parallelism in many cases. This parallelism can be used to further speed up the overall computation of the Newton step.

1. T. F. Coleman and W. Xu, *Fast Newton computations*, in progress.
2. T. F. Coleman and A. Verma, *Structured and efficient Jacobian calculation*, in: editors, Computational Differentiation: Techniques, Applications and Tools, M. Berz, C.Bischof, G. Corliss and A. Griewank (eds.) , 149–159, Philadelphia, SIAM (1996)

Automatic Computation of Sensitivities for a Parallel Aerodynamic Simulation

Arno Rasch, H. Martin Bücker, and Christian H. Bischof

Institute for Scientific Computing
RWTH Aachen University, D–52056 Aachen, Germany
E-mail: {*rasch, buecker, bischof*}*@sc.rwth-aachen.de*

Abstract

Derivatives of functions given in the form of large-scale simulation codes are frequently used in computational science and engineering. Examples include design optimization, parameter estimation, solution of nonlinear systems, and inverse problems. In this note we address the computation of derivatives of a parallel computational fluid dynamics code by automatic differentiation. More precisely, we are interested in the derivatives of the flow field around a three-dimensional airfoil with respect to the angle of attack and the yaw angle. We discuss strategies for transforming MPI commands by the forward and reverse modes of automatic differentiation and report performance results on a Sun Fire E2900.

Parallel Jacobian Accumulation

Ebadollah Varnik and Uwe Naumann

Department of Computer Science,
RWTH Aachen University, D-52056 Aachen, Germany
E-mail: {varnik, naumann}@stce.rwth-aachen.de

Abstract

The focus of this work is on the parallelization of the process of Jacobian accumulation using *vertex elimination*[1] . In automatic differentiation the accumulation of the Jacobian F' of a vector function $F : I\!R^n \rightarrow I\!R^m$ implemented as a computer program can be regarded as a transformation of its *linearized computational graph* into a subgraph of the directed complete bipartite graph $K_{n,m}$. This transformation can be performed by applying a vertex elimination technique. Following the notation in Griewank's book[2] we assume that F can be decomposed into a *code list* by assigning the result of any scalar *elemental* function to a unique *code list variable*. The code list induces a directed acyclic computational graph G. For given input values the computational graph is linearized by attaching the values of the *local partial derivatives* to the corresponding edges. The Jacobian F' can be computed on the linearized computational graph by elimination of all intermediates vertices resulting in a bipartite graph with labels on the remaining edges representing exactly the nonzero entries of F'. Following the chain rule an intermediate vertex can be eliminated by multiplying the edge labels over all paths connecting pairwise the corresponding predecessor and successor followed by adding these products. In order to parallelize the Jacobian accumulation by vertex elimination, we decompose the computational graph G of F into k subgraphs G_p. Application of the vertex elimination technique to a subgraph G_p yields the local Jacobian matrix F'_p. The reduction to the Jacobian F' is computed as the chained product of k local Jacobian matrices. We report on first numerical results of two parallel approaches to Jacobian accumulation using vertex elimination.

1. A. Griewank and S. Reese, *On the calculation of Jacobian matrices by the Markowitz rule*, in: Automatic Differentiation of Algorithms: Theory, Implementation, and Application, pp. 126–135, SIAM, Philadelphia, PA, (1991).
2. A. Griewank, *Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation*, Frontiers in Applied Mathematics **19**, SIAM, Philadelphia, (2000).

Session

“Scheduling”

Thursday, September 6, 2007
11:00 to 12:30

Layer-Based Scheduling Algorithms for Multiprocessor-Tasks with Precedence Constraints

Jörg Dümmler, Raphael Kunis, and Gudula Rünger

Chemnitz University of Technology,
Department of Computer Science, 09107 Chemnitz, Germany
E-mail: {djo, krap, ruenger}@cs.tu-chemnitz.de

Abstract

A current challenge in the development of parallel applications for distributed memory platforms is the achievement of a good scalability even for a high number of processors. The scalability is impacted by the use of collective communication operations, e. g. broadcast operations, whose runtime exhibits a logarithmic or linear dependence on the number of utilized processors. The multiprocessor-task (M-Task) programming model with precedence constraints can help to reduce this overhead.

The efficient execution of an M-Task application requires a schedule that is adapted to the target platform. A schedule that leads to a good performance on one platform may result in an inefficient execution on another platform. The choice of a suitable schedule is therefore an important factor in the development of M-Task applications. Additionally, the optimal schedule may depend on the size or the structure of the input data of the application. As a consequence, the schedule for an application that is run on a variety of target platforms or with varying input data may have to be recomputed many times. To release the programmer from the tedious task of determining a good schedule by hand, many scheduling algorithms for M-Task applications have been developed. Most of these approaches belong to the classes of *allocation-and-scheduling-based* or *layer-based* algorithms.

In this paper we examine *layer-based* scheduling algorithms. Based on the *layer-based* approach we develop an extension methodology to enable scheduling algorithms for independent M-Tasks to handle precedence constraints. We apply this methodology to three approximation algorithms (with performance guarantees of 2, $\sqrt{3}(1+\epsilon)$ and $\frac{3}{2}(1+\epsilon)$) and derive new scheduling algorithms for M-Tasks with precedence constraints. We compare these algorithms with existing *layer-based* approaches. In this comparison we consider the runtime of the scheduling algorithm itself and the expected execution time of the produced schedules. The comparison is based on a test set consisting of synthetic M-Task applications with 50 to 1000 M-Tasks scheduled for target platforms with 16 to 256 processors.

The results show that the runtime of the tested scheduling algorithms is approximately equal for low numbers of processors but there are notable differences for an increasing number of processors. The comparison of the makespans of the produced schedules shows that all M-Task scheduling algorithms clearly outperform a pure data parallel and a pure task parallel execution especially for a large number of processors. Additionally, our results show that the scheduling algorithm, which obtains the best schedules, changes from a low number of processors to a high number of processors but is independent of the number of M-Tasks. Finally, we derive a guideline for M-Task application developers describing which scheduling algorithm is most suitable in which situation.

Unified Scheduling of I/O- and Computation-Jobs for Climate Research Environments

N. Peter Drakenberg[1] **and Sven Trautmann**[2]

[1] German Climate Computing Center,
Bundesstraße 55, 20146 Hamburg, Germany
E-mail: drakenberg@dkrz.de

[2] Max Planck Institute for Meteorology
Bundesstraße 53, 20146 Hamburg, Germany
E-mail: sven.trautmann@zmaw.de

Abstract

Climate simulation is not only used to predict global warming and its consequences. Like weather prediction, climate simulation is of significant importance to many areas of human society. Notable current and future examples being prediction of outbreaks of infectious diseases such as cholera and malaria, prediction of agricultural conditions and surface water supply, and of course prediction of "difficult" seasonal weather conditions.

In order to achieve reasonable run times (and waiting times), climate simulations are typically run on high-performance computer systems (*e.g.*, vector supercomputers or large clusters). In addition to using large amounts of processor time, climate simulations also consume and produce large volumes of data. For the small to medium resolution models currently used, the typical size of starting configurations for a simulation is approximately 1 GB. Each model run (*e.g.*, using ECHAM5, ARPEGE or GISS) typically uses 32 Opteron/Xeon-type processor cores and produces 1.5–1.7 GB of data per hour of run-time. High resolution models and models forced by observation data are connected with significantly higher amounts of input and output data, but are so far only rarely used. This will change in the near future (particularly w.r.t. higher resolutions) if climate researchers get more used to the resources provided by cluster systems and their characteristics.

The cluster system we use provides more than 1000 processor cores, a parallel file system (*Lustre*) with a capacity of about 130 Terabytes and runs the Linux operating system on all nodes. The cluster has had to be integrated in an existing environment with a shared file system (Sun QFS/SamFS) providing about 75 Terabyte of storage capacity and an underlying storage system with over 4 Petabyte capacity. As a user front-end to the cluster we use the Sun Grid Engine (SGE). The SGE is a resource management tool, the purpose of which is to accept *jobs* submitted by users, and schedule the jobs to run on appropriate systems available to it and in accordance with defined resource management policies.

In the full version of this paper we start by presenting and explaining a sample climate model simulation run. We then describe the file system structure and give benchmark results w.r.t. *IOZone*, `cp` and `dd`. Based on the benchmark results we present how data moving jobs can be tightly integrated into the Sun Grid Engine and be scheduled to minimize impact on computation jobs. For this purpose we define a so-called consumable attribute `ioload` for the SGE and implement a custom load sensor for all of Lustre's object storage servers.

Session

“Performance Analysis I”

Thursday, September 6, 2007
11:00 to 12:30

Analyzing Cache Bandwidth on the Intel Core 2 Architecture

Robert Schöne, Wolfgang E. Nagel, and Stefan Pflüger

Center for Information Services and
High Performance Computing,
Technische Universität Dresden
01062 Dresden, Germany
E-mail: {robert.schoene, wolfgang.nagel, stefan.pflueger}@tu-dresden.de

Abstract

Intel Core 2 processors are used in servers, desktops, and notebooks. They combine the Intel64 Instruction Set Architecture with a new microarchitecture, based on Intel Core and are proclaimed by their vendor as the "world's best processors". In this paper, measured bandwidths between the computing cores and the different caches are presented.
To gain these performance results, the STREAM benchmark[1] was adapted to fit the BenchIT[2] interface. While the STREAM benchmark implements a simple and well-understood measurement kernel, BenchIT provides a comfortable benchmarking environment and a variable problem size for a single execution of the same algorithm.
The resulting benchmark has been extended to produce more exact performance results but it was also optimized for the tested Core 2 Duo processor. To achieve this goal, different compilers, compilerflags but also preprocessor statements and code restructuring have been checked. Finally, the concluding version was run on several other processors to achieve an overview of the different performances.

1. J. D. McCalpin, *Memory Bandwidth and Machine Balance in Current High Performance Computers*, IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, (Dec. 1995).
2. G. Juckeland, S. Börner, M. Kluge, S. Kölling, W. E. Nagel, S. Pflüger, H. Röding, S. Seidl, T. William, and Robert Wloch, *BenchIT - Performance Measurement and Comparison for Scientific Applications*, Proc. ParCo2003, pp. 501–508, (2004), `http://www.benchit.org/DOWNLOAD/DOC/parco2003_paper.pdf`.

Analyzing Mutual Influences of High Performance Computing Programs on SGI Altix 3700 and 4700 Systems with PARbench

Rick Janda, Matthias S. Müller, Wolfgang E. Nagel, and Bernd Trenkler

Center of Information Services and
High Performance Computing
Dresden University of Technology
01162 Dresden, Germany
E-mail: rick.janda@zuehlke.com
E-mail: {matthias.mueller, wolfgang.nagel, bernd.trenkler}@tu-dresden.de

Abstract

Nowadays, most high performance computing systems run in multiprogramming mode with several user programs simultaneously utilizing the available CPUs. Even though most current SMP systems are implemented as ccNUMA to reduce the bottleneck of main memory access, the user programs still compete in different ways for resources and influence the scheduler decisions with their generated load.

The paper presents the investigation results of the SGI Altix 3700Bx2 of the TU-Dresden and its successor system the Altix 4700 with the PARbench system. The PARbench system is a multiprogramming and multithreading benchmark system, which enables the user to assess the system behavior under typical production work load and identify bottlenecks and scheduling problems.

The Altix 3700 and 4700 with their directory based ccNUMA architecture are the largest SMP systems in the market and promise a high scalability combined with a good isolation of the several system nodes. Part of the conducted tests validates these promised features and analyzes the connection network and the utilized Intel Itanium 2 Madison (Altix 3700Bx2) and dual core Itanium 2 Montecito (Altix 4700) CPUs.

The paper will also show practical problems of the shared system bus by two CPUs each in the 3700 system and compare these situation with the internally shared system bus of the dual core CPUs in the 4700 system.

Further tests examine the load balancing behavior and it's consequences to OpenMP parallelized programs under overload.

Low-level Benchmarking of a New Cluster Architecture

Norbert Eicker[1] **and Thomas Lippert**[1,2]

[1] NIC/ZAM,
Forschungszentrum Jülich, 52425 Jülich, Germany
E-mail: {n.eicker, th.lippert}@fz-juelich.de

[2] Department C
Bergische Universität Wuppertal
42097 Wuppertal, Germany

Abstract

The JULI[a] project is the first in a series of projects carried out at the Zentralinstitut für Angewandte Mathematik (ZAM) of the Forschungszentrum Jülich (FZJ). The mission of these projects is to develop and evaluate new state-of-the-art cluster platforms to prepare the upgrade of FZJ's general purpose supercomputer system in 2008.

Following this philosophy, the JULI project aims at integrating very compact compute hardware with lowest power consumptions, smallest latency interprocessor communication, and best possible process management systems. As project partners from industry we have chosen IBM Development Laboratory Böblingen (Germany) for PPC based node hardware and various software components, QLogic as contributer of the InfiniPath interconnect and corresponding MPI library and ParTec with their ParaStation cluster middleware.

Within this paper we will present and discuss results of synthetic low-level benchmarks on JULI. This will shed light on the most important hardware features of the system, like effective latency and effective bandwidth of memory and communication hardware or performance of the FPU. These parameters are the basis for any performance estimate in high-performance computing.

The conclusions we can draw from JULI extend the project and the type of hardware involved. The finding that dual-core or multi-core architectures will complicate life in HPC significantly is not surprising: until now the growth-rate of processor performances was slightly larger than the increase of memory-bandwidth. With the appearance of architecture with multiple cores the amount of memory bandwidth required to balance the system for HPC increases by factors significantly larger than 1 for each generation of processors.

Furthermore, the JULI project shows that in HPC we have to minimize the distance between processor and HCA, i.e. we need to keep the infrastructure in between as simple as possible. In the case of the JULI cluster the latency penalty for inter-node communication already is significant: the time a CPU has to wait for data from an interconnect will become more and more expensive: JULI's latency-penalty of $1.0\ \mu$sec compared to other PCIe architectures corresponds to $\mathcal{O}(2500)$ cycles of each CPU, a number that amounts to $\mathcal{O}(40000)$ floating point operations.

[a] **JU**elich **LI**nux Cluster

Session

“Biomedical Applications”

Thursday, September 6, 2007
11:00 to 12:30

Parallel Ensemble of Decision Trees for Neuronal Sources Localization of the Brain Activity

Elena Popova

Moscow State University, Faculty of Computational Mathematics and Cybernetics,
Leninskie Gory, VMK Faculty, 119992 Moscow, Russia
E-mail: eapopova@gmail.com

Abstract

Parallel computing became the only way for analysis of large databases in biomedical problems. This report put emphasis on the problem of detection of psychoneurological illness using the space-time electroencephalography (EEG) recordings on the scalp. One of the important problems in this case is the localizing of the neuronal active sources using experimental measurements. Database has a mix type. EEG data consists on at least 21 space channels recording. The time window is 40 time points each and its number vary from 100 to 1000 windows. The need of parallel data processing in real time is obvious for the databases containing recording for many patients[1].

Parallel ensemble of decision trees[2] approach for the problem of neuronal sources localization in the brain is suggested. The main idea of a new method is to consider the sources parameters as the decision trees attributes in parallel instead of direct handling with row space-time measurements. Suggested algorithm is based on construction and classification of the training database for ensemble of trees depending on the value of residual relative energy (RRE) error. If the localization problem states for 2 or more dipoles the size of training set becomes a key point due to main memory. This paper describes the method of attribute selection and data distribution between processors for each decision tree construction. Parallel ensemble consists of individual trees which are trained on the database related to one time point in the current time window. If the time points are selected comparatively close the voting of constructed ensemble gives stable zone of dipole position.

The parallel algorithm of source localization is developed on SMP architectures. OpenMP approach is used for parallel implementation on SMP architectures and the hybrid model of parallel computations based on MPI and OpenMP is also discussed. The proposed methods of parallelization were tested on noisy model databases, on the real filtered data and rough EEG signals.

1. Y. O. Halchenko, S. J. Hanson, and B. A. Pearlmutter, *Multimodal Integration: fMRI, MRI, EEG, MEG*, J. Advanced Im. Pr. in MRI, 223–265, (2005).
2. A. Borisov and I. Chikalov, *Performance and Scalability Analysis of Tree-Based Models in Large-Scale Data-Mining Problems*, Intel Tech. J. 143–150, (2005).

Experimenting Grid Protocols to Improve Privacy Preservation in Efficient Distributed Image Processing

Antonella Galizia[1], **Federica Viti**[2], **Daniele D'Agostino**[1], **Ivan Merelli**[2], **Luciano Milanesi**[2], **and Andrea Clematis**[1]

[1] Institute for Applied Mathematics and Information Technologies,
National Research Council, Genova, Italy
E-mail: {*clematis, dago, antonella*}*@ge.imati.cnr.it*

[2] Institute for Biomedical Technologies,
National Research Council, Segrate (MI), Italy
E-mail: {*federica.viti, ivan.merelli, luciano.milanesi*}*@itb.cnr.it*

Abstract

In the image processing community, a topic of interest is represented by privacy preservation and secure processing of sensitive data, as biomedical images. This issue is stressed in distributed systems aimed to support data sharing and multi-institutional collaborations, in fact the transfer of patient clinical information and genomic data around the world must be completely protected. One of the possible architectures to achieve these goals is the Grid[1], that provides storage and computational resources under a solid security infrastructure.

In this paper we describe our approach to add privacy preservation to $\mathrm{PIMA(GE)^2}$ Lib, Parallel IMAGE processing GEnoa Library, in order to allow its use to process distributed medical images. We enable the interaction of $\mathrm{PIMA(GE)^2}$ Lib with computational grid, and in this way we are able to exploit grid security policies, and obtain computational and storage facilities. In particular we considered the EGEE, Enabling Grids for E-sciencE[2] middleware, that provides the Grid File Access Library (GFAL) API. Using GFAL, we developed a set of specialized I/O functions for secure data access; they have been added to the library. These functions avoid sensitive data transfer enabling their secure elaborations. This strategy does not compromise the user-friendliness of the library.

We tested this approach through the analysis of images obtained considering the Tissue MicroArray technique[3]. We evaluated also the performance aspects, comparing GFAL with other protocols for data acquisition. The experimentation of the Grid to analyse remote TMA images in the EGEE environment gives positive results.

1. I. Forster and C. Kesselman, *The grid: blueprint for a new computing infrastructure*, 2nd Edition. Morgan Kaufmann Publishers Inc, (2004).
2. EGEE Home Page, `http://www.eu-egee.org/`
3. J. Kononen, L. Bubendorf, A. Kallioniemi, M. Barlund, P. Schraml, S. Leighton, J. Torhorst, M. J. Mihatsch, G. Sauter, and O. P Kallioniemi, *Tissue microarrays for high-throughput molecular profiling of tumor specimens*, Nature Medicine **4**, 844–847, (1998).

Efficient Parallel Simulations in Support of Medical Device Design

Marek Behr, Mike Nicolai, and Markus Probst

Chair for Computational Analysis of Technical Systems (CATS)
Center for Computational Engineering Science (CCES)
RWTH Aachen University
52056 Aachen, Germany
E-mail: {*behr, nicolai, probst*}*@cats.rwth-aachen.de*

Abstract

Parallel computing is enabling computational engineering analyses of unprecedented complexity to be performed. We are reporting our experiences with parallel finite element flow simulations supporting the development of implantable ventricular assist devices in the form of continuous-flow axial pumps. These pumps offer simplicity and reliability needed in long-term clinical applications. Their design however poses continuing challenges.

The challenges can be itemized as high shear stress levels, flow stagnation and onset of clotting, and loss of pump efficiency. Elevated shear stress levels are particularly evident in mechanical biomedical devices. Biorheological studies point out a number of platelet responses to shear: adhesion, secretion, aggregation (i.e. activation), and finally, hemolysis (i.e. damage). These responses are dependent on both the level and duration of elevated shear at the platelet surface, not unlike the response to chemical factors. The primary response of the red blood cells is hemolysis, again dependent on dose and time.

The distribution of the shear stress levels in a complex flow field in a rotary blood pump chamber as well as the duration of the blood cells' exposure to these pathological conditions are largely unknown. Device designers are often compelled to make decisions about the details of pump configuration guided only by the global, time- and space-averaged, indicators of the shear stress inside the pump, such as the hemolysis observations made on the exiting blood stream. This challenge of detailed analysis and reduction of shear stress levels while maintaining pump efficiency as well as the need to pinpoint any persistent stagnation areas in the flow field motivates our current computational work.

We describe the flow simulation methodology and apply it to the problem of analysis of blood flow in an axial ventricular assist device – the MicroMed DeBakey LVAD. This pump consists of the flow straightener, a six-bladed impeller, and a six-bladed diffuser inside a cylindrical housing. The simulations must explore a range of impeller speed and various pressure conditions. The computations are performed on an IBM Blue Gene.

Using these simulations as an illustration, we will focus on the architecture of the MPI-based finite element code, and on steps taken to ensure reasonable parallel speed-up on 4096 CPUs, including performance analysis and bottleneck identification. In view of the need for design optimization, where unsteady flow fields as well as their sensitivities with respect to the design parameters must be computed repeatedly while seeking a minimum of an flow-dependent objective function, the use of thousands of CPUs is a critical factor that makes such optimization practical.

Session

“Fault Tolerance”

Thursday, September 6, 2007
14:00 to 16:00

Mitigating the Post-Recovery Overhead in Fault Tolerant Systems

Guna Santos, Angelo Duarte, Dolores Rexachs, and Emilio Luque

Computer Architecture and Operating Systems Department,
University Autonoma of Barcelona. Bellaterra, Barcelona 08193, España
E-mail: {*guna, angelo*}*@caos.uab.es,* {*dolores.rexachs, emilio.luque*}*@uab.es*

Abstract

The demand for computational power has been leading the improvement of the High Performance Computing (HPC) area. In this area, fault tolerance plays an important role in order to provide high availability isolating the application from the faults effects. Performance and availability form an undissociable binomial for some kind of applications. Therefore, the fault tolerant solutions must take into consideration these two constraints when it has been designed. Our previous work, called RADIC[1], implemented a basic level protection allowing to recover from faults just using the active cluster resources, changing the system configuration. Such approach is well suited for some kind of applications like the dynamic workload balanced. However, it may genenerate some performance degradation in other cases. In this paper, we present RADIC II, which incorporates a new protection level in RADIC using a dynamic redundancy functionality, allowing to mitigate or avoid the recovery side-effects. RADIC II allows dynamically inserting new spare nodes during the application execution in order to replace the requested ones. Moreover, RADIC II provides a transparent management of spare nodes, which is able to request and use them without need any administrator intervention and not maintaining any centralized information about these spare nodes. The results has shown that RADIC-II operates correctly and becomes itself as a good approach to provide high availability to the parallel applications without suffer a system degradation in post-recovery execution.We evaluate our solution performing several experiments comparing the effects of recovery having or not available spare nodes. These experiments observe two measures: the overall execution time, and the throughput of an application. We executed a matrix product algorithm, using a SPMD approach implementing a Cannon algorithm and we executed an N-Body particle simulation using a pipeline paradigm. The results are conclusive showing the benefits of use our solution in these scenarios[a]

1. A. Duarte, D. Rexachs, and E. Luque, *A distributed scheme for fault tolerance in large clusters of workstations*, in: Proc. Parallel Computing 2005 (ParCo2005), Malaga, G. Joubert *et al.*, (eds.), NIC Series **35**, (2005).

[a]This work is supported by MEyC-Espaa under contract TIN 2004-03388

Towards Fault Resilient Global Arrays

Vinod Tipparaju, Manoj Krishnan, Bruce Palmer, Fabrizio Petrini, and Jarek Nieplocha

Pacific Northwest National Laboratory
Richland, WA 99352, USA
E-mail: {vinod, manoj, bruce.palmer, fabrizio.petrini, jarek.nieplocha}@pnl.gov

Abstract

As the number of processors for high-end systems grows to tens or hundred of thousands, hardware failures are becoming frequent and must be handled in such manner that the capability of the machines is not severely degraded. Scientific applications for upcoming petascale systems are expected to be based on a variety of algorithms, methods, and programming models that impose differing requirements on the system resources (memory, disk, system area network, external connectivity) and may have widely differing resilience to faults or data corruption. Because of this variability it is not practical to rely on a single technique (e.g., system initiated checkpoint/restart) for addressing fault tolerance for all applications and programming models.

Under the US DoE FASTOS program funding, we have been developing fault tolerance solutions for global address space (GAS) models. This includes both automatic (user-transparent) as well as user-coordinated approach described in the current paper where we focus on fault tolerance for the Global Arrays (GA) model. GA implements a global address space programming model, is compatible with MPI, and offers bindings to multiple popular serial languages. These two technologies are complementary; however, they differ in several key respects such as portability, generality, and use of system resources. Our user-transparent approach relies on Xen virtualization and supports high-speed networks. With Xen, we can checkpoint and migrate the entire OS image including the application to another node. The user-coordinated approach uses a spare pool of processors to perform reconfiguration after the fault, process virtualization, incremental or full checkpoint scheme and restart capabilities. We demonstrate usefulness of fault resilient Global Arrays in context of a Self Consistent Field (SCF) chemistry application. On our experimental platform, the overhead introduced by checkpointing is less than 1% of the total execution time. A time to recover from a single fault increased the execution time by only 8%.

Using AOP to Automatically Provide Distribution, Fault Tolerance, and Load Balancing to the CORBA–$\mathcal{LC}$ Component Model

Diego Sevilla[1], **José M. García**[1], **and Antonio Gómez**[2]

[1] Department of Computer Engineering

[2] Department of Information and Communications Engineering
University of Murcia, Spain
E-mail: {dsevilla, jmgarcia}@ditec.um.es, skarmeta@dif.um.es

Abstract

Programming abstractions, libraries and frameworks are needed to better approach the design and implementation of distributed High Performance Computing (HPC) applications, as the scale and number of distributed resources is growing. Moreover, when Quality of Service (QoS) requirements such as load balancing, efficient resource usage and fault tolerance have to be met, the resulting code is harder to develop, maintain, and reuse, as the code for providing the QoS requirements gets normally mixed with the functionality code.

Component Technology, on the other hand, allows a better modularity and reusability of applications and even a better support for the development of distributed applications, as those applications can be partitioned in terms of components installed and running (deployed) in the different hosts participating in the system. Components also have requirements in forms of the aforementioned non-functional aspects. In our approach, the code for ensuring these aspects can be automatically generated based on the requirements stated by components and applications, thus leveraging the component implementer of having to deal with these non-functional aspects.

In this paper we present the characteristics and the convenience of the generated code for dealing with load balancing, distribution, and fault-tolerance aspects in the context of CORBA–$\mathcal{LC}$. CORBA–$\mathcal{LC}$ is a lightweight distributed reflective component model based on CORBA that imposes a peer network model in which the whole network acts as a repository for managing and assigning the whole set of resources: components, CPU cycles, memory, etc.

VirtuaLinux: Virtualised High-Density Clusters with no Single Point of Failure

Marco Aldinucci[1], Marco Danelutto[1], Massimo Torquati[1], Francesco Polzella[1], Gianmarco Spinatelli[1], Marco Vanneschi[1], Alessandro Gervaso[2], Manuel Cacitti[2], and Pierfrancesco Zuccato[2]

[1] Computer Science Department,
University of Pisa, Largo B. Pontecorvo 3, I-56127 Pisa, Italy
E-mail: {aldinuc, marcod, spinatel, polzella, torquati, vannesch}@di.unipi.it

[2] Eurotech S.p.A.,
Via Fratelli Solari 3/a, I-33020 Amaro (UD), Italy
E-mail: {m.cacitti, a.gervaso, p.zuccato}@exadron.com

Abstract

VirtuaLinux is a Linux meta-distribution that allows the creation, deployment and administration of both physical and virtualized clusters with no single point of failure. They are avoided by means of a combination of architectural, software and hardware strategies, including the transparent support for disk-less and master-less cluster configuration[1]. VirtuaLinux supports the creation and management of Virtual Clusters, each of them being a collection of Xen-based Virtual Machines that are running onto one or more physical nodes of a cluster, and that are wired by a virtual private network[2].

VirtuaLinux Virtual Cluster Manager enables the system administrator to seamlessly create, save, restore virtual clusters, and to map and dynamically re-map them onto the nodes of the physical cluster. These features enable the flexible management of a physical cluster, thus both the consolidation of several parallel applications (possibly running on different OSes) in single platform, and to dynamically share and partition a single expensive large cluster.

We introduce and discuss VirtuaLinux architecture, features, and tools. These rely on a novel disk abstraction layer, which enables the fast, space-efficient, dynamic creation of virtual clusters composed of fully independent complete virtual machines. VirtuaLinux is an open source software package under GPL available at `http://virtualinux.sourceforge.net/`.

1. M. Aldinucci, M. Torquati, M. Vanneschi, M. Cacitti, A. Gervaso, and P. Zuccato, *VirtuaLinux design principles*, Technical Report TR-07-13, Università di Pisa, Dipartimento di Informatica, Italy, (June 2007).
2. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, *Xen and the art of virtualization*, in: Proc. 9th ACM Symposium on Operating Systems Principles (SOSP'03), pp. 164–177, ACM Press, (2003).

Session

“Performance Analysis II”

Thursday, September 6, 2007
14:00 to 16:00

Comparative Study of Concurrency Control on Bulk-Synchronous Parallel Search Engines

Carolina Bonacic[1] **and Mauricio Marin**[2]

[1] ArTeCS, Complutense University of Madrid, Spain
CEQUA, University of Magallanes, Chile
E-mail: cbonacic@fis.ucm.es

[2] Yahoo! Research, Santiago, Chile
E-mail: mmarin@yahoo-inc.com

Abstract

In this paper we propose and evaluate the performance of concurrency control strategies for a parallel search engine that is able to cope efficiently with concurrent read/write operations. Read operations come in the usual form of queries submitted to the search engine and write operations come in the form of new documents added to the text collection in an on-line manner, namely the insertions are embedded into the main stream of user queries in an unpredictable arrival order but with query results respecting causality.

Gb Ethernet Protocols for Clusters: An OpenMPI, TIPC, GAMMA Case Study

Stylianos Bounanos and Martin Fleury

University of Essex, Electronic Systems Engineering Department,
Colchester, CO4 3SQ, United Kingdom
E-mail: {sbouna, fleum}@essex.ac.uk

Abstract

Gigabit Ethernet is a standard feature of cluster machines. Provision of fast network interconnects is negated if communication software cannot match the available throughput and latency. Transparent Inter Process Communication (TIPC)[1] has been proposed as an alternative to TCP in terms of reduced message latency and system time. This study compares through low-level tests and application benchmarks the relative performance of TIPC, OpenMPI, and also what improvement the GAMMA User-Level Network interface[2], can bring over both these. TIPC's system time usage is reduced compared to TCP, leading to computational gains, especially on loaded cluster nodes. GAMMA is shown to bring significant improvement in computational performance on an unloaded cluster but a more flexible alternative is OpenMPI[3] running over TIPC.

This paper's contribution is a port of TIPC to OpenMPI v. 1.0.2 to thoroughly benchmark the benefits of a cluster-optimized protocol compared to TCP/IP. The tests are conducted with and without background load. In turn, the paper also examines: the value of hardware off-loading of some compute-intensive TCP features onto the Network Interface Card (NIC); and whether the GAMMA Linux kernel module is a way of improving standard MPI with MPICH implementation. The communication software is evaluated by low-level metrics of performance and by a standardized set of NAS application The cluster under test, with Gb switching and high-performance AMD processors, scales up to thirty processors; it is of a moderate but accessible size.

The tests comparing TCP and TIPC reveal that TIPC has very real advantages over TCP, both within and outside an OMPI environment. This is the paper's strongest conclusion, as it is shown for low-level benchmarks and for NAS application kernels. Short message latency was reduced. Moreover, the efficiency of the TIPC stack is demonstrated for nodes with high background load. The GAMMA user-level network interface (with MPI) is also capable of much improved performance over a combination of TCP and MPI.

1. J. P. Malloy, *TIPC: Providing Communication for Linux Clusters*, Linux Symposium, **2**, pp. 347–356, (2004).
2. G. Ciaccio, M. Ehlert and B. Schnor, *Exploiting Gigabit Ethernet for Cluster Applications*, 27^{th} IEEE Conf. on Local Computer Networks, pp. 669–678, (2002).
3. E. Gabriel *et al.*, *Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation*, 11^{th} Eur. PVM/MPI Users' Group Meeting, pp. 97–104, (2004).

Performance Measurements and Analysis of the BlueGene/L MPI Implementation

Michael Hofmann[a] **and Gudula Rünger**

Department of Computer Science,
Chemnitz University of Technology
E-mail: {mhofma, ruenger}@informatik.tu-chemnitz.de

Abstract

The massively parallel architecture of the BlueGene/L[1] supercomputer poses a challenge to the efficiency of parallel applications in scientific computing. Three different communication networks are used for implementing MPI communication operations. The specifics of these networks and a number of optimized algorithms cause varying performance results for single communication operations.

This paper presents measurements of several MPI operations for a BlueGene/L system[b] using up to 4,096 nodes. The efficiency of the MPI operations is discussed in terms of the utilized communication networks. For point-to-point communication, the influences of the network topology and the different communication protocols are shown. The results demonstrate the efficient usage of collective broadcast and reduction operations and investigate how to benefit from the specialties of the communication hardware. The behavior of non-blocking operations is shown and performance results for different implementations of communication schemes like nearest neighbor communication are given. The properties of the BlueGene/L system have significant influences on the performance of MPI communication operations. The presented results give an overview of their efficient usage and can be used to optimize the performance of parallel applications.

1. G. L.-T. Chiu, M. Gupta, and A. K. Royyuru, (eds). *IBM J. of Research and Development: Blue Gene* , vol. 49, IBM, (2005).

[a]Supported by Deutsche Forschungsgemeinschaft (DFG)

[b]Measurements are performed on the BlueGene/L system at the John von Neumann Institute for Computing, Jülich, Germany. `http://www.fz-juelich.de/zam/ibm-bgl`

Potential Performance Improvement of Collective Operations in Current UPC Implementations

Rafik A. Salama and Ahmed Sameh

Department of Computer Science,
American University in Cairo
E-mail: {raamir, sameh}@aucegypt.edu

Abstract

Advances in high-performance architectures and networking have made it possible to build complex systems with several parallel and distributed interacting components. Unfortunately, the software needed to support such complex interactions has lagged behind. In specific we focused on the collective operations that involve more than one thread/process and act on multiple streams of data. Generally it is known that the parallel language's API should provide both algorithmic and run-time system support to optimize the performance of these operations. Some developers, however, choose to play clever and start from the language's primitive operations and write their own versions of the parallel operations. The question that always pops up: Are these developers wise? In this paper we have used a number of benchmarks to test performance improvement of the primitive operations over current Unified Parallel C (UPC) native collective implementations. Specifically, the current native collective implementation of the Berkley UPC was compared with the optimized Michigan UPC primitive implementation. The Michigan UPC implementation was tested using two provided techniques (PUSH & PULL) to compare the performance improvement using each of them. These two techniques have shown a notable performance effect which we have explained. Berkley UPC Implementation has shown worse performance in allexchange, allscatter, allbroadcast and better performance in allgather and allreduce. The allreduce primitive collective was then further optimized using a binary tree algorithm which showed better performance, the allgather and allexchange was approached for enhancement using a borrowed MPI algorithm but it was not implemented since it required asynchronous communication support from the runtime to provide better algorithm that provided no wait. So generally, we have found a lag in the Unified Parallel C collective operations which can be improved using algorithmic optimization (i.e. borrowing MPI Collective algorithms) and runtime optimizations (i.e. supporting asynchronous communication).

Session

"MHD and Turbulence Simulation"

Thursday, September 6, 2007
14:00 to 16:00

Massively Parallel Simulations of Solar Flares and Plasma Turbulence

**Lukas Arnold, Christoph Beetz, Jürgen Dreher,
Holger Homann, Christoph Schwarz, and Rainer Grauer**

Institute for Theoretical Physics I,
Ruhr-University Bochum, Germany
E-mail: grauer@tp1.rub.de

Abstract

Some of the outstanding problems in space- and astrophysical plasmasystems include solar flares and hydro- or magnetohydrodynamic turbulence (e.g. in the interstellar medium). Both fields demand for high resolution and thus numerical simulations need an efficient parallel implementation.

Numerical modelling of solar flares require the resolution of phenomena on scales varying from global scales down to scales comparable to the electron skin depth near reconnection events. In order to treat this enormous range of scales efficently, our numerical simulations make use of block-structured adaptive mesh refinement for the compressible magnetohydrodynamic (MHD) equations. The MHD equations are solved using a third order central weighted ENO scheme. Special care has to be taken to maintain the solenoidality of the magnetic field. This is realized either by the method of divergence cleaning or constraint transport. In our AMR framework *racoon* parallelization is obtained using space filling curves for the loadbalancing. First scaling results on BlueGene will be reported.

racoon also has the ability to advect tracer particles with the flow using the same parallelisation strategy as for the blocks. The main numerical work is spent in the interpolation routines from cell values to the actual particle positions.

Statistics of Lagrangian tracer particles in turbulent fluid and plasma flows is of great importance to understand important processes in astrophysics like star formation and in fusion plasmas where it is directy connected to anomalous transport and diffusion. The numerical simulations of incompressible fluid and plasma turbulence are performed using a pseudo-spectral code, which is implemented for two different computer architectures: A standard slice based FFT for simulations on platforms like the IBM regatta series and a column like decomposition for massive parallel simulations on BlueGene. The treatment of passive tracer particles is also done in a parallel way. This is necessary, because a large number of particles has to be integrated in order to sample the considered volume homogeneously and obtain reliable statistical results. We performed simulations with up to 10^7 particles on the IBM p690 machine. The crucial point is the interpolation scheme needed in order to obtain the velocity field at the particle positions from the numerical grid. The code uses a tri-cubic interpolation scheme which on the one hand provides a high degree of accuracy and on the other hand parallelizes efficiently.

Especially, the comparison between fluid and plasma flows helps to understand the different impact of dissipative structures on Eulerian and Lagrangian turbulence.

Object-Oriented Programming and Parallel Computing in Radiative Magnetohydrodynamics Simulations

Vladimir Gasilov, Sergei D'yachenko, Olga Olkhovskaya, Alexei Boldarev, Elena Kartasheva, and Sergei Boldyrev

Institute for Mathematical Modelling, Russian Academy of Sciences,
Miusskaya Sq. 4-A, 125047 Moscow, Russia
E-mail: {gasilov, boldar, bsn}@imamod.ru

Abstract

Modern problems in pulsed-power energetics issue a real challenge to the computer simulation theory and practice. High-performance computing is a promising technology for modeling complex multiscale nonlinear processes such as transient flows of strongly radiative multicharged plasmas. An essential part of such numerical investigations is devoted to computer simulation of pinches resulted from electric explosion of cold matter, e.g. gas-puff jets, foam strings, or metallic wire arrays. The goal of numerical research in pulsed-power is to study the evolution of very intensive transient electric discharges and to perform a multiparametric optimization of future experimental schemes.

Certain specificity of introducing parallelism into a program complex in our case relates to the object-oriented nature of MARPLE code, developed in IMM RAS, which essentially employs C++ language facilities, such as polymorphism, encapsulation, inheritance, and parametric programming. Special data structures based on the concept of topological complex have been elaborated to provide problem statement in an arbitrary domain, and for handling unstructured meshes, including dynamic mesh changes. Some of these structures have to be adapted to allow for parallel computations and data exchanges, taking into account the requirement of keeping interprocessor communication adequately small.

Highly accurate simulation of radiative energy transfer, including detailed reproducing of the radiation spectrum, is among the most important requirements to the developed code. Thereto, the entire spectrum is divided into a number of frequency ranges (from several tens to several hundreds). It is necessary to reconstruct the radiation field with respect to its angular distribution for each frequency range, that makes the radiative transport computation one of the most laborious steps in radiative MHD simulations. The frequency ranges model requires repeating large volume of uniform computation with different sets of opacity and emissivity values for each range. So we decided to carry out these computations concurrently by several processors, and then to collect the results by simple summation, using the fact that all wavebands produce a uniform and independent contribution to the whole radiative energy fluxes.

Parallel computing technology applied to the radiative energy transfer calculation helped us to reduce the total processing time by factor of about 2 to 4. This is already a significant achievement, since for the experiment scheme optimization a big series of numerical simulations is actually needed. The concurrent advantage is that the number of ranges in a spectrum can be greatly increased, that gives immediate effect on the accuracy and quality of numerical solutions.

Parallel Simulation of Turbulent Magneto-hydrodynamic Flows

Axelle Viré[2], Dmitry Krasnov[1], Bernard Knaepen[2], and Thomas Boeck[1]

[1] Fakultät für Maschinenbau, Technische Universität Ilmenau,
P.O. Box 100565, 98684 Ilmenau, Germany
E-mail: {thomas.boeck, dmitry.krasnov}@tu-ilmenau.de

[2] Université Libre de Bruxelles, Service de Physique Théorique et Mathématique,
Campus Plaine - CP231, Boulevard du Triomphe, 1050 Brussels, Belgium
E-mail: {bknaepen, avire}@ulb.ac.be

Abstract

Turbulent flows of electrically conducting liquids in the presence of external magnetic fields occur in a variety of metallurgical processes. Such magnetohydrodynamic (MHD) flows are affected by the Lorentz force arising from the induced electric currents in the liquid. Important examples are the electromagnetic braking of molten steel in continuous casting or in electromagnetic stirring of melts. Experimental investigation of MHD flows is complicated by the opacity and corrosiveness of liquid metals. For this reason, the accurate predicition of such flows by numerical simulations is of particular interest.

The magnetic field affects the properties of turbulence through additional Joule energy dissipation, and by the anisotropic action of the Lorentz force. The detailed investigation of these effects requires direct numerical simulations (DNS), in which all turbulent eddies are resolved by the numerical grid. However, DNS are restricted to low Reynolds numbers, i.e., they cannot be applied for flow parameters close to practical applications. For such problems, the small turbulent eddies cannot be resolved and have to be modelled. The so-called Large-Eddy simulation (LES), which is conceptually similar to DNS, is particularly appealing because LES retains the time-dependent large scale motions of the flow. The effect of the small scale motions is accounted for by subgrid models.

Many recent works have examined the performance of certain subgrid-stress models for different classes of flow problems. In this context, DNS are essential for the validation of the subgrid-stress models. We are interested in the performance of the dynamic Smagorinsky model for turbulent MHD flows. As recently shown by Knaepen & Moin (Phys. Fluids **16**, 1255-1261, 2004), this model can fairly accurately reproduce the features of decaying homogeneous MHD turbulence observed previously by DNS.

In the continuation of this work, we focus on the simplest wall-bounded flow, namely a channel flow subjected to a uniform wall-normal magnetic field. We compare the parallel performance of two different numerical approaches – a pseudospectral method and a finite-volume method – for both DNS and LES of such channel flows. The pseudospectral code is shown to be more accurate than the finite-volume one at the same grid resolution. In contrast to the finite-volume method, the implementation of the LES model leads to a significant additional computational cost in the pseudospectral method. For two test cases, the dynamic Smagorinsky model reproduces the DNS results with approx. 2% of error, using the pseudospectral method.

Pseudo-Spectral Modeling in Geodynamo

Maxim Reshetnyak and Bernhard Steffen

Central Institute for Applied Mathematics (ZAM),
Research Centre Jülich, 52425 Jülich, Germany
E-mail: {b.steffen, m.reshetnyak}@fz-juelich.de

Abstract

Many stars and planets have magnetic fields. The heat flux causes 3D convection of plasma or metal, which can generate a large-scale magnetic field like those observed. The small-scale behavior, demonstrating self-similarity in a wide range of the spatial and temporal scales, is a field of active research using modeling, as it is usually not observed.
Rapid rotation gives a geostrophic system, where convection degenerates in the direction of the axis of rotation and all variation along this axis is weak. These systems are somewhere in between the full 3D and 2D-systems. Their special properties show up in the physical and the spectral space simultaneously. Pseudo-spectral modeling solves the PDE in spectral space for easy calculations of integrals and derivatives. The nonlinear terms are calculated physical space, requiring many direct and inverse FFTs per time step. We apply this technique to the thermal convection problem with heating from below in a Cartesian box. Above a threshold of the kinetic energy the system generates the magnetic field.
The most time consuming part of our MPI code is FFT transforms. For efficiency, we selected a FFT library which makes use of the symmetry of the fields. The optimal number of processors is $\sim$ half the number of grid planes, with superlinear speedup. The single node performance is poor, each processor delivering only $\sim$ 5% of its peak rate.
We see cyclonic convection with a cyclone density of the $\sim E^{-1/3}$ (E Ekman number $\sim 10^{-15}$ for earth). This causes a high anisotropy of the convection even for high Reynolds numbers. Our simulations demonstrates the generation of the large-scale hydrodynamic helicity. Helicity is an integral of the Navier-Stokes equation, and it has close relation to the α-effect which generates the large scale magnetic field via the small-scale turbulence. This process has three stages:. At first, the magnetic field grows exponentially from a small seed. When the magnetic and kinetic energies are comparable the growth slows down, and finally equilibrium is reached. The magnetic field again quenches the helicity, damping primarily the toroidal part of velocity field. It slows down the rotation of the cyclones (anti-cyclones). The helicity causes a divergence (convergence) of the cyclones near the upper (lower) boundaries ($z = 0,\ 1$). It is generated at the boundaries and transported to center of the box. It changes sign at the middle of the box.
Convection and dynamo systems are dissipative, so the equilibrium of the system in sense of the statistical mechanics is not reached. The kinetic energy is injected into the system at the medium scale of cyclons, one sink of energy is at the small viscous scale, another at the large (magnetic field) scale. For some (small) scales the cascade of the energy is direct (like it is in the Kolmogorov's like turbulence), for others (larger than cyclones) it is inverse, like it is observed in 2D turbulence. At the small scales there is a constant energy flux, as is plausible as from theorie and from semi-empirical models.

Session

“Parallel Tools and Middleware”

Friday, September 7, 2007
11:00 to 12:30

Design and Implementation of a General-Purpose API of Progress and Performance Indicators

Ivan Rodero, Francesc Guim, Julita Corbalan, and Jesus Labarta

Barcelona Supercomputing Center,
Technical University of Catalonia,
Jordi Girona 29, 08034 Barcelona, Spain
E-mail: {ivan.rodero, francesc.guim, julita.corbalan, jesus.labarta}@bsc.es

Abstract

In HPC centers, queuing systems are used by the users to access the HPC resources. They provide interfaces that allow users to submit jobs, track the jobs during their execution and carry out actions on the jobs (i.e. cancel or resume). For example, in LoadLeveler the llsubmit command is used to submit an LL script to the system. Once the job is queued, and the scheduler decides to start it, it is mapped to the resources by the corresponding resource manager.

After job submission, users lose control of the job and they only dispose of a very restricted set of interfaces for accessing data concerning the performance, or progress or job events. In this situation, the queuing system only provides a list of the submitted jobs and some information about them, such as the job status or the running time. Although this is the scenario in almost all the HPC centers, there is information about the jobs that have been submitted that is missing but required by the users. For example, they want to know when their application will start running, once started, when it will finish, how much time remains, and if their application is performing well enough.

If experienced users could obtain such information during the job run time, they would be able to take decisions that would improve system behavior. For example, if an application is not achieving the expected performance, the user can decide to adjust some parameters on run time, or even resubmit this application with new parameters. In the worst case, if an application achieves low performance during its execution, it is cancelled by the system because of a wall clock limit timeout and thus consumes resources unnecessarily.

In this paper we present a general-purpose API which can implement progress and provide performance indicators of individual applications. The API is generic and it is designed to be used at different levels, from the operating system to a grid portal. The design can also be extended, and the development of new services on top the API are easy to develop. The implementation is done through a lightweight library to avoid important overheads and starvation with the running application. We also present two additional components built on top of the API and explain how they ares in the HPC-Europa portal, which is a production testbed composed of HPC centers. The API and the additional tools can be used in both sequential and parallel applications (MPI, OpenMP and mixed MPI+OpenMP). Furthermore, we discuss how to use the proposed API and tools in the eNANOS project to implement scheduling policies based on dynamic load balancing techniques and self tuning in run time, to improve the behavior of the applications and optimize the use of resources.

Efficient Object Placement including Node Selection in a Distributed Virtual Machine

Jose M. Velasco[1], **David Atienza**[2], **Katzalin Olcoz**[2], **and Francisco Tirado**[2]

[1] DACYA/UCM, Avda. Complutense s/n, 28040 Madrid, Spain
E-mail: mvelascc@fis.ucm.es

[2] *E-mail: {datienza, katzalin, ptirado}@dacya.ucm.es*

Abstract

A cluster-aware Java Virtual Machine (JVM) can transparently execute java applications in a distributed fashion on the nodes of a cluster while providing the programmer with the single system image of a classical JVM. This way multi-threaded server applications can take advantage of cluster resources without increasing programming complexity.

When a JVM is ported into a distributed environment, one of the most challenging tasks is the development of an efficient, scalable and fault-tolerant automatic dynamic memory manager. The automatic recycling of the memory blocks no longer used is one of the most attractive characteristics of Java for software engineers, as they do not need to worry about designing a correct dynamic memory management. This automatic process, very well-known as Garbage Collection, makes much easier the development of complex parallel applications that include different modules and algorithms that need to be taken care of from the software engineering point of view. However, since the GC is an additional module with intensive processing demands that runs concurrently with the application itself, it always accounts for a critical portion of the total execution time spent inside the virtual machine in uniprocessor systems. As Plainfosse outlined, distributed GC is even harder because of the difficult job to keep updated the changing references between address spaces of the different nodes.

Additionaly to this problem, the node choice policy for object emplacement is an additional task that can facilitate or difficult an efficient memory management. The inter-node message production increases proportionally to the distribution of objects that share dependencies. These dependencies can be seen as a connectivity graph, where objects are situated in the vertices and edges represent references.

In prior work, it have been proposed a global object space design based on object access behaviour and object connectivity. In this approach it is needed to profile extensively the object behaviour. The main weakness of this solution is that knowledge of the connectivity graph during the allocation phase requieres a lot of profile code and extra metadata with the consequent cost in both performance and space.

In our proposal the object placement is based on object connectivity as well, but it is managed by the GC and it takes place during its reclaiming phase. As a result, we have eliminated the profiling phase and the extra needed code. We have chosen the tracing garbage collector family as optimal candidate. Our results show a significative reduction in both number of inter-node messages and global execution time.

Memory Debugging of MPI-Parallel Applications in Open MPI

Rainer Keller, Shiqing Fan, and Michael Resch

High-Performance Computing Center, University of Stuttgart,
E-mail: {keller, fan, resch}@hlrs.de

Abstract

In this paper we describe the implementation of memory checking functionality based on instrumentation using `valgrind`[1]. The `valgrind`-suite allows checking for memory-related bugs, such as buffer-overflows, usage of uninitialized data, double-frees or wrong parameters for known functions such as `strcpy`. However, `valgrind` does not have any knowledge of the semantics of MPI-calls. In this paper, we present the integration of `valgrind`-instrumentation within Open MPI in the so-called `memchecker`-framework.

The combination of valgrind based checking functions within the MPI-implementation offers superior debugging functionality, for errors that otherwise are not possible to detect with comparable MPI-debugging tools. The tight control of the user's memory passed to Open MPI, allows not only to find application errors, but also helps track bugs within Open MPI itself.

We describe the actual checks done within Open MPI, error-classes being detectable in user applications, how memory buffers internally are being handled, and show the performance implications of this instrumentation. The functionality has been used to test codes such as MPI testsuite developped at HLRS[2] and other well-known codes such as PETsc[3], ScalaPACK and CPMD[4].

1. J. Seward and N. Nethercote. *Using Valgrind to detect undefined value errors with bit-precision*, in: Proc. USENIX'05 Annual Technical Conference, Anaheim, CA, (April 2005).
2. R. Keller and M. Resch, *Testing the correctness of MPI implementations*, in: Proc. 5th Int. Symp. on Parallel and Distributed Computing, Timisoara, Romania, pp. 291–295, (2006).
3. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.
4. CPMD Homepage. Internet, (2007). `http://www.cpmd.org`.

Session

“Image Processing and Visualization”

Friday, September 7, 2007
11:00 to 12:30

Lessons Learned Using a Camera Cluster to Detect and Locate Objects

Daniel Stødle, Phuong Hoai Ha, John Markus Bjørndalen, and Otto J. Anshus

Dept. of Computer Science,
Faculty of Science,
N-9037 University of Tromsø, Norway
E-mail: {*daniels, phuong, jmb, otto*}*@cs.uit.no*

Abstract

A typical commodity camera rarely supports selecting a region of interest to reduce bandwidth, and depending on the extent of image processing, a single CPU may not be sufficient to process data from the camera. Further, such cameras often lack support for synchronized inter-camera image capture, making it more difficult to accurately relate images from different cameras. Such limitations induce challenges in designing large-scale human-computer interaction systems.

We have developed a scalable, dedicated parallel camera system for detecting objects in front of a wall-sized, high-resolution, tiled display. The system is used to support multi-user touch-free[a] interaction with applications running on a 220-inch 7x4 tiles, 7168x3072 pixels resolution display wall. This requires that the system can accurately and with low latency determine the positions of fingers, hands, arms and other objects in front of the wall. To achieve this, a consistent and synchronized set of position data from each camera is needed to enable triangulation of object positions.

Since a single camera can saturate either the bus or CPU, depending on the camera characteristics and the image processing complexity, the system is designed to support configuring the number of cameras per computer according to bandwidth and processing needs. To minimize image processing latency, the system focuses only on detecting where objects are, rather than what they are, reducing the complexity of the problem. To overcome the lack of synchronized cameras, short periods of waiting are used. Our experimental study using 16 commodity cameras has shown that our system achieves a latency of 115 ms, which is acceptable for playing games like Quake 3 Arena and Homeworld on the display wall. The majority of the latency is due to camera capture, while the next biggest contributor is processing detected object positions.

The main contributions of this paper are the lessons learned from building and using the system, including: (i) The flexibility of the system architecture allows configuring available camera and processing resources in order to accommodate the end-application's needs, (ii) by reducing the complexity of image processing from identifying *what* objects are to identifying *where* objects are, processing is reduced, and (iii) despite the lack of cameras with support for inter-camera synchronization, useful results may still be obtained by introducing short periods of waiting.

[a]We refer to the interface as "touch-free," as users must be able to interact with the display wall without direct physical contact with the display wall's non-rigid canvas.

Parallel Morphological Neural Networks for Hyperspectral Image Classification on Fully Heterogeneous and Homogeneous Networks of Workstations

Javier Plaza, Antonio Plaza, Rosa Pérez, and Pablo Martínez

Department of Technology of Computers and Communications,
University of Extremadura, Avda. de la Universidad s/n, E-10071 Cáceres, Spain
E-mail: {jplaza, aplaza, rosapere, pablomar}@unex.es

Abstract

Hyperspectral imaging is a new technique in remote sensing which allows an airborne/satellite sensor to collect hundreds of images (at different wavelength channels) for the same area on the surface of the Earth. Most hyperspectral imaging applications require that a response is provided quickly enough for practical use. In this paper, we develop a new parallel morphological/neural algorithm for thematic classification of hyperspectral images which has been specifically designed to be efficiently executed on fully heterogeneous computing platforms. The algorithm integrates spatial and spectral information by making use of a special kind of parallel morphological perceptrons specifically developed for this purpose. The performance of the different implementation strategies adopted for the two main modules (morphological and neural) is tested in this work by using a collection of hyperspectral image data sets obtained from real, application-oriented remote sensing missions. Performance tests are conducted in various homogeneous/heterogeneous computing platforms, including two networks of workstations at University of Maryland and a Beowulf cluster at NASA's Goddard Space Flight Center.

1. A. F. H. Goetz, G. Vane, J. E. Solomon and B. N. Rock, *Imaging spectrometry for Earth remote sensing*, Science **228**, 1147–1153, (1985).
2. A. Plaza, J. Plaza and D. Valencia, *Impact of platform heterogeneity on the design of parallel algorithms for morphological processing of high-dimensional image data*, Journal of Supercomputing **40**, 81–107, (2007).
3. S. Suresh, S. N. Omkar, and V. Mani, *Parallel implementation of back-propagation algorithm in networks of workstations*, IEEE Transactions on Parallel and Distributed Systems **16**, 24–34, (2005).
4. A. Lastovetsky, *Parallel computing on heterogeneous networks*, Wiley-Interscience: Hoboken, NJ (2003).

Hybrid Parallelization for Interactive Exploration in Virtual Environments

Marc Wolter, Marc Schirski, and Torsten Kuhlen

Virtual Reality Group, RWTH Aachen University
E-mail: {wolter, schirski, kuhlen}@rz.rwth-aachen.de

Abstract

With the growing size of scientific simulation output, efficient algorithms for the presentation of the resulting data to a human user have become an important topic. Virtual Reality (VR) is a useful instrument for displaying and interacting with these visualizations, as time-varying 3D structures are perceived in a natural way. However, the application of VR introduces an interactivity criterion ($<$ 100 ms) to the systems response time, which is not easily met even with modern high performance computers

To deal with this problem, we propose a task distribution optimized for the user's interaction behavior. Tasks with frequent parameter changes are computed locally inside a resampled user-defined region of interest (ROI), while tasks for which parameters change less frequently or predictably are computed using hybrid parallelization on a remote HPC machine. The simplification of the ROI in space and complexity together with optimized local visualization algorithms (e.g., using graphics processing units (GPUs) or thread-level parallelization) allows for interactive response times for a large class of algorithms.

The less frequent resampling of a time-varying ROI out of the whole data set should be fast, but here the interactivity criterion is relaxed somewhat, because this event is of significantly less frequent occurrence. In previous work the resampling, in particular the cell search in the original unstructured grid, was the most time-consuming part of the ROI's recomputation. Therefore, this approach was only feasible with structured grids. To eliminate this problem, we propose an optimized resampling algorithm based on a kd-tree cell search, which is easily parallelizable with OpenMP. As resampling an unstructured grid into a Cartesian grid introduces interpolation errors, but is necessary for fast local visualization, we provide the user with different error metrics describing the resampling quality. On the basis of the global or local error, the user can validate his findings and choose a finer or coarser resampling resolution or even return to the original grid structure.

Resampling is integrated in a hybrid parallelization system. Independent time steps of the time-varying data set are distributed on nodes using MPI. Single time steps are processed via thread-based pipelining (loading, computing, sending) as well as OpenMP parallelization of the computation task. We show a significant improvement in runtime and good scalability for our resampling algorithm. However, two new bottlenecks emerge, file access and ROI transmission to the visualization system. Nonetheless, extraction of time-varying, Cartesian regions of interest out of unstructured grids with high resolutions is made possible within short waiting times, as required for the execution in virtual environments.

Session

“Fluid Dynamics Simulation”

Friday, September 7, 2007
11:00 to 12:30

Parallelisation of a Geothermal Simulation Package: A Case Study on Four Multicore Architectures

Andreas Wolf[1], **Volker Rath**[2], **and H. Martin Bücker**[1]

[1] Institute for Scientific Computing,
RWTH Aachen University, D–52056 Aachen, Germany
E-mail: {wolf, buecker}@sc.rwth-aachen.de

[2] Institute for Applied Geophysics,
RWTH Aachen University, D–52056 Aachen, Germany
E-mail: v.rath@geophysik.rwth-aachen.de

Abstract

In this case study, we assess the performance of an OpenMP-based approach to parallelise a new geothermal simulation package that is currently developed at the Institute for Applied Geophysics, RWTH Aachen University, and that is used by several academic institutions and different small enterprises in geotechnical engineering. It is capable of solving the coupled transient equations for groundwater flow and heat transport.

By enabling an incremental approach to shared-memory programming, we use the OpenMP[1,2] programming paradigm to offer a smooth transition from serial to multicore architectures. An overall parallelisation strategy is described which consists of parallelising the two most time- and memory-intensive tasks: the assembly of large, sparse coefficient matrices and the solution of the resulting systems of linear equations. For the iterative solution of the linear systems, a parallel version of the biconjugate gradient method[3] for nonsymmetric systems is used. A parallel implementation of ILU(0) is taken for preconditioning.

Two sets of numerical experiments involving a conductive model of a synthetic sedimentary basin are configured to show a comparison between a small cache-intensive problem and a larger computational model that tries to address hardware limitations of the main memory. We compare their parallel performance on four recent multicore architectures (Clovertown, Woodcrest, and two Opteron-based platforms). The OpenMP parallelisation of this simulation package using up to 8 threads demonstrates that it is possible to obtain moderate to good speedups for different discretised models with only modest human effort.

1. OpenMP Architecture Review Board, *OpenMP Application Program Interface, Version 2.5*, (2005).
2. R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon, *Parallel Programming in OpenMP*, Morgan Kaufmann Publishers, San Francisco, CA, (2001).
3. H. A. van der Vorst, *BI-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing **13**, 631–644, (1992).

A Lattice Gas Cellular Automata Simulator with Cell Broadband Engine^{TM}

Yusuke Arai, Ryo Sawai, Yoshiki Yamaguchi, Tsutomu Maruyama, and Moritoshi Yasunaga

Graduate School of Systems and Information Engineering,
University of Tsukuba Ten-ou-dai 1-1-1 Tsukuba, Ibaraki, 305-8573, Japan
E-mail: {arai, yoshiki}@islab.cs.tsukuba.ac.jp

Abstract

In this paper, we have a large increase in speed of simulating FHP-III lattice-gas automata (LGA) using one Cell Broadband Engine produced by TOSHIBA Corporation Semiconductor Company. Experiments on the LGA simulation that includes more than 11 million lattice points showed good results, whose speedup by our approach is about 3.2 times compared with Core2 Duo 2.4GHz.

A considerable number of studies have been conducted on Cellular Automata (CA) that J. Von Neumann and Stan Ulam proposed[1]. Many kinds of phenomena of hydrodynamics and reaction-diffusion system have been an important modeling problem. Then, the Lattice Gas Automaton (LGA) has been the central model for simulating them[2]. The Cell Broadband Engine (CBE) is widely accepted as one of novel architectures for the next generation. developed by Sony, TOSHIBA, and IBM.

To achieve high performance with one CBE, we adopt SIMD instruction set and iteration algorithm. In 2-dimensional FHP model, a lattice point is expressed 7bit because it includes seven particles. The six particles and the other have the unit velocity and no velocity, respectively. Sixteen (=128/8) lattice points can be computed in parallel, and however particle state on a point generally differs from the other neighbors. Given this factor, the collision rules and propagation are written by only bit operation without using any branching instructions. The problem that we have to consider next is region-splitting method. Our target simulation has more than 11 million sites. In other words, it requires 10MB storage capacity. Therefore, we have to divide the simulation space to a region that can be stored in LS. The result indicated that speedup rate of CBE (3.2GHz) to Core2 Duo at 2.4GHz and a dedicated system with FPGA[3] was about 3.2 times and 5 times respectively. Future task is to analyze the tradeoff between data bandwidth of EIB and wasted computation demanded for reducing frequent memory access. The EIB is implemented as a set of four concentric rings that is a 128 bit wide interconnect. For the improvement in computational speed, we need to examine the many parameters of LGA.

1. J. V. Neumann, *The Theory of Self-Reproducing Automata*, A. W. Burks (ed), Univ. of Illinois Press, Urbana and London, (1966).
2. U. Frish, *et al.*, *Lattice gas hydrodynamics in two and three dimensions*, Complex Systems, **1**, pp. 649–707, (1987).
3. T. Kobori, T. Maruyama, and T. Hoshino, *A Cellular Automata System with FPGA*, IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 120–129, (2001).

Session

“Hyperscalable Applications”

Friday, September 7, 2007
11:00 to 12:30

Massively Parallel All Atom Protein Folding in a Single Day

Abhinav Verma[1], **Srinivasa M. Gopal**[2], **Alexander Schug**[2],
Jung S. Oh[3], **Konstantin V. Klenin**[2], **Kyu H. Lee**[3], **and Wolfgang Wenzel**[2]

[1] Institute for Scientific Computing,
Research Centre Karlsruhe, D-76344, Karlsruhe, Germany
E-mail: verma@int.fzk.de

[2] Institute for Nanotechnology,
Research Centre Karlsruhe, D-76344, Karlsruhe, Germany
E-mail: {gopal, klenin, schug, wenzel}@int.fzk.de

[3] Supercomputational Materials Lab,
Korean Institute for Science and Technology, Seoul, Korea
E-mail: {soo5, khlee}@kist.re.kr

Abstract

The search for efficient methods for all-atom protein folding remains an important grand-computational challenge. Folding even small proteins can literally consume thousands of CPU years. We have developed models and algorithms which permit reproducible and predictive folding of small proteins from random initial conformations using free-energy forcefields. According to Anfinsen's thermodynamic hypothesis many proteins are in thermodynamic equilibrium with their environment under physiological conditions. Their unique three-dimensional native conformation then corresponds to the global optimum of a suitable free-energy model. The free-energy model captures the internal energy of a given backbone conformation with the associated solvent and side-chain entropy via an implicit solvent model.

Here we describe predictive all atom folding simulations of proteins with up to sixty amino acids using an evolutionary stochastic optimization technique[1,2]. We have implemented a master-client model of this algorithm on an IBM BlueGene, where the algorithm scales near perfectly from 64 to 4096 nodes. Using a PC cluster we have folded the sixty-amino acid bacterial ribosomal protein L20 to near-native experimental conformations. Starting from a completely extended conformation with 2048 nodes of the IBM BlueGene we predictively fold the fourty amino acid HIV accessory protein in less than 24 hours.

1. A. Schug, and W. Wenzel, *Predictive in-silico all-atom folding of a four helix protein with a free-energy model*, J. Am. Chem. Soc. **126**, 16736–16737, (2004).
2. A. Schug, and W. Wenzel, *An evolutionary strategy for all-atom folding of the sixty amino acid bacterial ribosomal proein L20*, Biophysical Journal **90**, 4273–4280, (2006).

Simulations of QCD in the Era of Sustained Tflop/s Computing

Thomas Streuer[1] **and Hinnerk Stüben**[2]

[1] Department of Physics and Astronomy,
University of Kentucky, Lexington, KY, USA
E-mail: thomas.streuer@desy.de

[2] Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB),
Takustr. 7, 14195 Berlin, Germany
E-mail: stueben@zib.de

Abstract

The first computer delivering a performance of more than 1 Tflop/s peak as well as in the Linpack benchmark appeared on the Top500 list in June 1997. For German QCD researchers it has taken until the installation of the current generation of supercomputers at national centres until a sustained Tflop/s was available in everyday runs of their simulation programmes.

In this paper we report on how the sustained performance was obtained on these supercomputers, the IBM BlueGene/L at NIC/ZAM Jülich and the SGI Altix 4700 at Garching/Munich. Both started user operation in 2006. The BlueGene/L has 16.384 CPUs (cores) and offers a peak performance of 45 Tflop/s. The Altix 4700 originally had 4096 cores delivering 26 Tflop/s peak. It was upgraded in 2007 to 9726 cores delivering 62 Tflop/s peak. The performance figures we present were measured on the upgraded system.

We compare the performance of Fortran/MPI code with assembler code. The latter allows to exploit concurrency at more levels, in particular in overlapping communication and computation as well as prefetching data from main memory. Using lower level programming techniques improves the performance of QCD programmes significantly. The speed-up that can be achieved in comparison to programming in Fortran is a factor of 1.3–2.0.

We found that our code scales up to the whole Blue Gene/L in Jülich. The highest performance measured was 8.05 Tflop/s on the whole machine. In production typically one rack (2048 cores) is used on which a performance 1.1 Tflop/s or 19 % of peak is sustained. The high performance and scaling could be obtained by using *double hummer* instructions and techniques to overlap communication and computation.

On the Altix 4700 the large L3 data cache helps a lot to boost performance. Due to the hierarchical nature of the communication network performance measurement depend to some degree on the placement of programmes. In production an average sustained performance of 1.485 Tflop/s or 23 % of peak is achieved when using 1000 cores.

Optimizing Lattice QCD Simulations on BlueGene/L

Stefan Krieg

Theoretische Physik, Bergische Universität Wuppertal &
Zentralinstitut für Angewandte Mathematik, Forschungszentrum Jülich GmbH
E-mail: s.krieg@fz-juelich.de

Abstract

Lattice QCD (LQCD) is the only known approach to Quantum Chromo Dynamics (QCD), the theory of the strong nuclear force, which allows the calculation of important low energy properties of the Standard Model of elementary particle physics such as hadron masses or the mechanisms of confinement and chiral symmetry breaking among others. Unfortunately LQCD Simulations require an enormous amount of computational resources which has been a mayor driving force behind the development of supercomputers so far. The IBM BlueGene/L (BGL) itself stems from a family of dedicated LQCD computers (QCDOC, QCDSP). While it was originally intended to study bimolecular phenomena such as protein folding, through its ancestry it is still a particularly well suited architecture for LQCD simulations.

Making good news of the BGL is unfortunately no trivial task. To use the CPU efficiently SIMD have to be used. As the auto-simdization of the compiler often fails to generate efficient code, typical C-code will not reach its theoretical peak performance. Fortunately, the performance of LQCD applications is dominated by a kernel operator which is not too complicated to be tuned by hand. I describe how one can use the so called intrinsics of the IBM XLC compiler to generate efficient code and prefetch data into L1 ahead of time. I will show how special runtime system calls that provide areas of memory with special caching strategies can significantly improve performance.

I will show how one can reach good performance with MPI but also describe how a special low level API can be used to more efficiently use the BGL's networks improving performance for LQCD. The resulting code shows almost prefect scaling properties and scales up to thousands of CPUs. This is in deed necessary to make good use of the BGL, since with the individual CPUs having a rather low clock frequency, good performance can only be achieved with a software with good scaling properties.

Although the FPU of the BGL CPU has no hardware for single precision arithmetic, it is still possible to perform single precision calculations. This is done by using special load and store operations, that e.g. load a single precision number from memory and convert it on the fly to double precision and store it in one of the double precision registers. Since LQCD is bandwidth limited, even for the BGL where double and single precision floating point operations have the same peak value it makes sense to use single precision arithmetic. Another advantage is the smaller memory footprint of the single precision numbers, which improves the scaling of the application with the number of CPUs. I will show how the single precision kernel compares to the double precision implementation and describe how one can use the single precision version to get double precision results with special solver algorithms and show the performance gain by this approach.

Mini-Symposium

"The Future of OpenMP in the Multi-Core Era"

Tuesday, September 4, 2007

OpenMP 3.0

J. Mark Bull

EPCC, University of Edinburgh,
King's Buildings, Mayfield Road,
Edinburgh, EH9 3JZ, UK.
E-mail: m.bull@epcc.ed.ac.uk

Abstract

Version 3.0 of the OpenMP language specification is in the final stages of preparation. We will describe the new features which are likely to be included in the new specification, including:

- Tasking constructs
- Multiple internal control variables
- Loop collapsing
- Stack size control
- Thread wait policy control
- Improved C++ compatibility
- Additional loop scheduling features
- Revisions to the memory model
- Additional routines to support nested parallelism
- Improved support for allocatable arrays and Fortran pointers

We will discuss the rationale for these new features, and present some examples of how they can be used.

OpenMP for Clusters

Larry Meadows

Intel Corporation,
2111 NE 25th Avenue, Hillsboro, OR, USA 97124
E-mail: lawrence.f.meadows@intel.com

Abstract

OpenMP[1] is designed for shared memory multiprocessors; however, shared memory multiprocessors with large processor counts are expensive, even in today's multi-core era. Clusters of modest- sized SMPs with fast interconnects are fairly cheap. Most users of these clusters program them using MPI.

Intel's Cluster OpenMP product is an alternative model. Using DVSM technology (descended from the Treadmarks software developed at Rice University), Cluster OpenMP provides the user with a full implementation of OpenMP that runs on any cluster of 64 bit Intel Architecture nodes (Intel64 or Itanium). Cluster OpenMP is included with the Intel compiler suite for C, C++, and Fortran.

This talk will introduce Cluster OpenMP, give a short review of the implementation and some technical details, discuss the kinds of applications that are appropriate for Cluster OpenMP, give an overview of the tools that Intel provides for porting and tuning Cluster OpenMP applications, and show performance evaluations for several Cluster OpenMP applications.

1. *OpenMP Application Program Interface 2.5*, OpenMP Architecture Review Board (2005).

Getting OpenMP Up to Speed

Ruud van der Pas

Sun Microsystems
Menlo Park, CA 94025
USA
E-mail: ruud.vanderpas@sun.com

Abstract

OpenMP provides for a very powerful and flexible programming model, but unfortunately there is a persistent misconception that it does not perform well. Surely one may have performance issues with an OpenMP program, but that does not mean these can not be addressed. In this talk, we first cover the basic rules how to get good performance out of an OpenMP program. This is followed by a detailed coverage of false sharing, a potentially nasty inhibitor of scalable performance. The talk concludes with several case studies, illustrating several of the points made.

PerfOMP: A Runtime Performance Monitoring API for OpenMP

Van Bui[1], Oscar Hernandez[1], Barbara Chapman[1], Rick Kufrin[2], Danesh Tafti[3], and Pradeep Gopalkrishnan[3]

[1] Department of Computer Science,
University of Houston
Houston, TX 77089
E-mail: {vtbui, oscar, chapman}@cs.uh.edu

[2] University of Illinois
Urbana, IL, 61801
E-mail: rkufrin@ncsa.uiuc.edu

[3] Department of Mechanical Engineering,
Virginia Tech
Blacksburg, VA, 24061
E-mail: {dtafti, pradeepg}@vt.edu

Abstract

Parallel programming languages/libraries including OpenMP, MPI, and UPC are either in the process of defining or have already established standard performance profiling interfaces. The OpenMP Architecture Review Board (ARB) recently sanctioned an interface specification for profiling/tracing tools that defines a protocol for two-way communications and control between the OpenMP runtime library and performance tools[1]. To evaluate this approach to performance measurement, the PerfOMP interface has been designed to operate as an intermediary software layer to support unidirectional communications from the OpenMP runtime library to performance tools. PerfOMP can support the implementation of the OpenMP ARB sanctioned profiling interface by providing the underlying infrastructure for tracking OpenMP events/states inside the OpenMP runtime and satisfying specific queries made by the performance tool to the OpenMP runtime. Alternatively, PerfOMP can be implemented to directly profile or trace the performance of an OpenMP application through the OpenMP runtime library. PerfOMP has been integrated into an existing open source compilation and performance tool environment and successfully tested with benchmark kernels and a production quality parallel computational fluid dynamics application. Design, implementation details and case study results are presented that highlight the benefits and potential insights that this approach can offer to compiler and tool developers, performance analysts, and end-user application scientists.

1. M. Itzkowitz, O. Mazurov, N. Copty, and Y. Lin, *White Paper: An OpenMP runtime API for profiling*, Sun Microsystems, Inc, (2006).

Affinity Matters!
OpenMP on Multicore and ccNUMA Architectures

Dieter an Mey and Christian Terboven

Center for Computing and Communication,
RWTH Aachen University, Germany
E-mail: {anmey, terboven}@rz.rwth-aachen.de

Abstract

OpenMP is an Application Programming Interface (API) for a portable, scalable programming model for developing shared-memory parallel applications in Fortran, C, and C++.

So far OpenMP was predominantly employed on large shared memory machines. With the growing number of cores on all kinds of processor chips and with additional OpenMP implementations e.g. the GNU and Visual Studio compilers, OpenMP is available for use by a rapidly growing, broad community. Upcoming multicore architectures make the playground for OpenMP programs even more diverse. The memory hierarchy will grow, with more caches on the processor chips.

Whereas applying OpenMP to Fortran and C programs on machines with a flat memory (UMA architecture) is straight forward in many cases, there are quite some pitfalls when using OpenMP for the parallelization of C++ codes on one hand and on ccNUMA architectures on the other hand. The increasing diversity of multicore processor architectures further introduces more aspects to be considered for obtaining good scalability.

Approaches to improve the support for memory and thread affinity within the upcoming OpenMP specification are still under discussion. So far operating system and compiler dependent calls to pin threads to processor cores and to control page allocation have to be employed to improve the scalability of OpenMP applications on ccNUMA and multicore architectures.

1. C. Terboven, D. an Mey, *OpenMP and C++*, IWOMP 2006, Reims, France, (2006).
2. S. Johnson, C. Ierotheou, A. Spiegel, D. an Mey, I. Hörschler, *Nested Parallelization of the Flow Solver TFS using the ParaWise Parallelization Environment*, IWOMP 2006, Reims, France, (2006).
3. Ch. Terboven, D. an Mey and S. Sarholz, *OpenMP on Multicore Architectures*, IWOMP 2007, Beijing, China, (2007).

Mini-Symposium

"Scaling Science Applications on Blue Gene"

Tuesday, September 4, 2007

Turbulence in Laterally Extended Systems

Jörg Schumacher[1] **and Matthias Pütz**[2]

[1] Department of Mechanical Engineering,
Technische Universität Ilmenau,
98684 Ilmenau, Germany
E-mail: joerg.schumacher@tu-ilmenau.de

[2] Deep Computing – Strategic Growth Business,
IBM Deutschland GmbH,
55131 Mainz, Germany
E-mail: mpuetz@de.ibm.com

Abstract

Turbulent flows appear frequently in lateral extensions that exceed the vertical ones by orders of magnitude. Examples can be found in planetary and stellar physics or in atmospheric and oceanographic science. Many turbulence studies in this context are then conducted in two dimensions from beginning. However, the reduction of the space dimension from three to two alters the turbulence physics in a fundamental way, e.g. as a reversal of the direction of the cascade of kinetic energy through the hierarchy of vortex structures. The crossover from three- to (quasi-)two-dimensional turbulence is the physical question which we want to study by numerical simulations. In order to analyse a possible formation of large lateral vortices – as observed in two-dimensional turbulence – we successively increase the aspect ratio of the cell while keeping the grid resolution the same. Frequently, such layers are in a state of convective turbulence that is caused by the temperature dependence of the fluid density. In addition Coriolis forces can appear due to rotation of the layer. This is exactly the system which we want to study, a flat convective layer with rotation. In the following, we describe preparational steps for the implementation and first tests of the simulation program on the Blue Gene/L system.

The Boussinesq equations are solved by a pseudospectral method for the three-dimensional case. Lateral boundary conditions are periodic and vertical ones free-slip. One of the main building blocks of the present numerical method is the fast Fourier transform (FFT). Due to the rather small memory size per core, the Blue Gene/L requires a volumetric FFT which decomposes the three-dimensional volume into so-called pencils and hence allows a parallelization degree of N^2. We compared therefore three different packages with respect to their strong scaling on up to 1024 CPUs.

Strong scaling tests of the whole code on up to one rack with 2048 CPUs are also reported. It turned out that runs in the virtual node mode are faster than those in the coprocessor mode. The effect of different processor mappings on the performance is analysed as well. For the largest grids on up to 16384 CPUs, we observe that the differences between the virtual node and coprocessor modes become small. This results from the fact that large local grid fractions do no longer fit into the L3 cache of a Blue Gene/L node and have to be streamed from memory. If the second core is used for computation as being the case in the virtual node mode the two cores are actually competing for the memory bandwidth of the node and the gain in computing time remains small. Furthermore, the communication time in the virtual node mode is then slightly bigger than in the coprocessor mode.

Large Simulations of Shear Flow in Mixtures via the Lattice Boltzmann Equation

Kevin Stratford[1] **and Jean Christophe Desplat**[2]

[1] Edinburgh Parallel Computing Centre, The University of Edinburgh, Edinburgh, Scotland
E-mail: kevin@epcc.ed.ac.uk

[2] Irish Centre for High-End Computing, Dublin, Ireland
E-mail: j-c.desplat@ichec.ie

Abstract

Fluid dynamics presents many computational challenges, particularly in the area of complex fluids, where microscopic/mesoscopic details of the fluid components are important in addition to the bulk properties such as the viscosity. One useful method for studying such systems is based on the lattice Boltzmann equation (LBE) for the incompressible Navier-Stokes equations. The LBE provides a natural way for the microscopic details — e.g., composition, liquid crystal ordering, and so on — to be coupled to the fluid flow. In addition, by relaxing the constraint of exact incompressibility the LBE allows the fluid pressure to be computed locally, and thus is extremely well suited to parallel computation. We give a brief overview of the method and its related performance and scalability issues.

One application where the LBE is useful is flow involving a mixture of fluids, where the fluid-fluid interface can evolve on the lattice in a natural way without the need for explicit interface tracking. Here, we consider the problem of spinodal decomposition involving two symmetric liquids. If at high temperature the liquids are miscible, a drop in temperature can cause spinodal decomposition, where the liquids separate and form domains which grow continuously in time. The growth in domain size occurs in a well-understood fashion and is ultimately limited by the container size in experiment, or the system size in simulation. However, experiments on sheared systems report saturation in the length scales after a period of (anisotropic) domain growth. The extreme elongation of the domains in the flow direction means that finite-size effects cannot be excluded as the reason for saturation even in experiments. Evidence for steady-states from simulations large enough to be uncontaminated by finite-size effects is therefore of great interest.

In our LBE calculations, a shear flow is driven by block-wise introduction of Lees-Edwards sliding periodic boundary conditions (first used in molecular dynamics). The different blocks of the system translate (conceptually) relative to each other as the simulation proceeds. This gives rise to the need for a refined version of the halo exchange between parallel sub-domains in one of the coordinate directions. We have two implementations of this approach (1) using point-to-point communication, and (2) using MPI-2 single-sided communication. We report on the relative merits of the two approaches.

In two dimensions, our recent results indicate unambiguously, for the first time, that non-equilibrium steady states do exist in sheared binary mixtures. In three dimensions, the situation is slightly more complex as the domains can grow in the vorticity direction. We illustrate the results of these large three-dimensional calculations and their performance.

Simulating Materials with Strong Correlations on BlueGene

Erik Koch

Institut für Festkörperforschung,
Forschungszentrum Jülich, 52425 Jülich, Germany
E-mail: e.koch@fz-juelich.de

Abstract

Understanding the physics of strongly correlated materials is one of the grand-challenges in condensed-matter physics. Simple approximations such as the local density approximation fail, due to the importance of the Coulomb repulsion between localized electrons. Instead we have to resort to non-perturbative many-body techniques. Such calculations are, however, only feasible for quite small model systems. This means that the full Hamiltonian of a real material has to be approximated by a lattice Hamiltonian comprising only the most important electronic degrees of freedom, while the effect of all other electrons can merely be included in an average way. Realistic calculations of strongly correlated materials need to include as many of the electronic degrees of freedom as possible.

Two important non-perturbative many-body solvers are quantum Monte Carlo and exact diagonalization by the Lanczos method. Being a sampling technique it is not too surprising that quantum Monte Carlo can readily take advantage of a machine like BlueGene, since communication while taking statistics is quite limited. It is more surprising that also the Lanczos method can benefit tremendously form the new architecture.

In the Lanczos method we have to handle the full many-body state of the correlated system. The method is thus limited by the available main memory. The principal problem for a distributed-memory implementation is that the central routine of the code, the application of the Hamiltonian to the many-body state, leads, due to the kinetic energy term, to very non-local memory access. Thus, a naive implementation, using one-sided communication to access the required vector elements, gives extremely poor performance, even a speed-down. We can, however, create an efficient MPI implementation by using a simple but important observation: in the kinetic term of the Hamiltonian the electron-spin is not changed. Thus, writing the many-body vector as a matrix $v(i_\uparrow, i_\downarrow)$, where the indices label spin-configurations, we find that the hopping term only connects vector elements that differ in one index. Hence, storing entire slices $v(i_\uparrow, :)$ on one node, the kinetic term for the spin-down electrons is local to that thread. After transposing v, the same is true for the hopping of the spin-up electrons. With an efficient matrix transpose, implemented via `MPI_Alltoall`, we thus obtain a highly scalable version of the Lanczos method.

We can use a simplified version of this approach to simulate quantum spins and decoherence.

1. A. Dolfen, *Massively parallel exact diagonlaization of strongly correlated systems*, Diploma Thesis, RWTH Aachen, (October 2006).

DL_POLY_3: Parallel Performance and Large Scale Simulations

Ilian T. Todorov

Computational Science & Engineering Department, STFC Daresbury Laboratory
Daresbury, Warrington WA4 1EP, Cheshire, United Kingdom
E-mail: i.t.todorov@dl.ac.uk

Abstract

DL_POLY_3[1] is a new generation software package for molecular dynamics (MD) simulations developed at Daresbury Laboratory. Dedicated to support academic research by a large number of groups worldwide, it is specially designed to address the efficient utilization of multi-processor power. DL_POLY_3 is employed for a wide range of applications and runs on many platforms; from single processor workstations to multi-processor computers. DL_POLY_3 parallelism, load balancing and memory distribution rely on an intricate blend of modern techniques such as domain decomposition[2] (DD), linked cells[3] (LC) and an adaptation of the Smoothed Particle Mesh Ewald Method (SPME)[4] for calculating long range Coulombic forces. This adaptation incorporates a novel 3D Fast Fourier Transform[5] (DaFT), that respects the data organisation and memory distribution dictated by DD and LC, which makes it possible to simulate systems of order of a hundred million particles and beyond.

This presentation shall start with an overview of the DL_POLY_3 package. We present data for DL_POLY_3 weak scaling and discuss its dependence on force field complexity[5]. We present benchmark data of DL_POLY_3 performance on modern cutting edge parallel architectures such as BG/L and Cray XT3. We shall finish with a review of the challenges and successes in applying DL_POLY_3 in two different by nature kinds of simulation studies: (i) radiation damage cascades in minerals and oxides[6], where the problem size (lengthscale) is of importance and (ii) biochemical simulations, where long timescales simulations are required[7].

1. I. T. Todorov, W. Smith, K. Trachenko & M. T. Dove, J. Mater. Chem. **16**, 1611–1618, (2006).
2. D. Rapaport, Comp. Phys. Comm. **62**, 217, (1991).
3. M. R. S. Pinches, D. Tildesley, W. Smith, Mol. Sim. **6**, 51, (1991).
4. U. Essmann, L. Perera, M. L. Berkowtz, T. Darden, H. Lee, L. G. Pedersen, J. Chem. Phys. **103**, 8577, (1995).
5. I. J. Bush, I. T. Todorov & W. Smith, Comp. Phys. Comm. **175**, 323–329, (2006).
6. I. T. Todorov, N. L. Allan, J. A. Purton, M. T. Dove & W. Smith, J. Mater. Sc. **42** 1920–1930, (2007).
7. C. W. Yong, W. Smith, R. W. Strange & S. S. Hasnain, Mol. Sim. **32**, 963, (2006).

Massively Parallel Simulation of Cardiac Electrical Wave Propagation on Blue Gene

Jeffrey J. Fox[1], **Gregery T. Buzzard**[2], **Robert Miller**[1], **and Fernando Siso-Nadal**[1]

[1] Gene Network Sciences,
53 Brown Rd, Ithaca, NY, USA
E-mail: {jeff, robert, siso}@gnsbiotech.com

[2] Department of Mathematics,
Purdue University,
W. Lafayette, Indiana, USA
E-mail: buzzard@math.purdue.edu

Abstract

Heart rhythm disorders are a leading contributor to morbidity and mortality in the industrialized world. Treatment and prevention of cardiac rhythm disorders remains difficult because the electrical signal that controls the hearts rhythm is determined by complex, multi-scale biological processes. Data-driven computer simulation is a promising tool for facilitating a better understanding of cardiac electrical properties. Conducting detailed, large-scale simulations of the cardiac electrical activity presents several challenges: the heart has a complicated 3D geometry, conduction of the electrical wave is anisotropic, and cardiac tissue is made up of cells with heterogeneous electrical properties.

Our group has developed a software platform for conducting massively parallel simulations of wave propagation in cardiac tissue. The code is being used in an on-going study to connect drug-induced modifications of molecular properties of heart cells to changes in tissue properties that might lead to a rhythm disorder. The platform uses a finite difference method for modeling the propagation of electrical waves in cardiac tissue using the cable equation with homogeneous Neumann boundary conditions. We use a novel algorithm which is based on the phase field method for handling the boundary conditions in complicated geometries. To map grid points within the spatial domain to compute nodes, an optimization process is used to balance the trade-offs between load balancing and increased overhead for communication. The performance of the code has been studied by simulating wave propagation in an anatomically realistic model of the left and right ventricles of a rabbit heart on Blue Gene partitions of up to 4,096 processors. Based on these results, the Blue Gene architecture seems particularly suited for cardiac simulation, and offers a promising platform for rapidly exploring cardiac electrical wave dynamics in large spatial domains.

Petascale Atmospheric General Circulation Models for CCSM

Henry M. Tufo

NCAR
1850 Table Mesa Dr.
Boulder, CO 80305
United States of America
E-mail: tufo@ucar.edu

Abstract

The High-Order Method Modeling Environment (HOMME), developed by the Computational and Information Systems Laboratory at NCAR in collaboration with the Computational Science Center at the University of Colorado, is a vehicle to investigate using high-order element based methods to build conservative and accurate dynamical cores. Currently, HOMME employs the discontinuous Galerkin and spectral element methods on a cubed-sphere tiled with quadrilateral elements, is capable of solving the shallow water equations and the dry/moist primitive equations, and has been shown to scale to 32,768 processors of an IBM BlueGene/L system.

The ultimate goal for HOMME is to provide the atmospheric science community a framework upon which to build a new generation of atmospheric general circulation models for CCSM based on high-order numerical methods that efficiently scale to hundreds-of-thousands of processors, achieve scientifically useful integration rates, provide monotonic and mass conserving transport of multiple species, and can easily couple to community physics packages.

Blue Gene/P: The Next Generation Enabling Breakthrough Simulation Based Engineering and Science

Kirk E. Jordan

Deep Computing
IBM Systems and Technology Group
1 Rogers Street
Cambridge, MA 02142
USA
E-mail: kjordan@us.ibm.com

Abstract

For the last several years, IBM's Blue Gene/L machine at Lawrence Livermore National Laboratory has been listed by the TOP500 organization as the fastest machine in the world. The TOP500 list requires the running of the LINPACK benchmark to obtain the performance numbers. However, the real interest in developing ultrascale computers like Blue Gene/L and the next generation machine, Blue Gene/P, is in the enablement of breakthrough simulation based engineering and science on problems that are of great interest to the scientific community with impact for society. Blue Gene/L is now demonstrating that ultrascale computing is of value in many areas such as protein folding, material sciences, fluid dynamics, climate modeling and geosciences to list a few. Blue Gene/P will further this enablement. In this talk, I will set the stage for the audience by quickly reviewing the hardware architecture and the software philosophy of Blue Gene/L. Using this as a basis, a comparison of new features added to Blue Gene/P will be discussed. In order, to give the audience some idea of how to exploit these new features, some application programs results will be presented. In conclusion, some discussion not only on the most obvious way to use Blue Gene/P will be given but also some thoughts on how one might use Blue Gene/P to tackle previously intractable problems. Hopefully, the latter will generate further discussion with the attendees and get them thinking of new ways to tackle to challenging problems.

Mini-Symposium

"Scalability and Usability of HPC Programming Tools"

Wednesday, September 5, 2007

Benchmarking the Stack Trace Analysis Tool for BlueGene/L

Gregory L. Lee[1], Dong H. Ahn[1], Dorian C. Arnold[2], Bronis R. de Supinski[1], Barton P. Miller[2], and Martin Schulz[1]

[1] Computation Directorate,
Lawrence Livermore National Laboratory, Livermore, California, U.S.A.
E-mail: {lee218, ahn1, bronis, schulzm}@llnl.gov

[2] Computer Sciences Department,
University of Wisconsin, Madison, Wisconsin, U.S.A.
E-mail: {darnold, bart}@cs.wisc.edu

Abstract

We present STATBench, an emulator of a scalable, lightweight, and effective tool to help debug extreme-scale parallel applications, the Stack Trace Analysis Tool (STAT). STAT periodically samples stack traces from application processes and organizes the samples into a call graph prefix tree that depicts process equivalence classes based on trace similarities. We have developed STATBench which only requires limited resources and yet allows us to evaluate the feasibility of and identify potential roadblocks to deploying STAT on entire large scale systems like the 131,072 processor BlueGene/L (BG/L) at Lawrence Livermore National Laboratory.

In this paper, we describe the implementation of STATBench and show how our design strategy is generally useful for emulating tool scaling behavior. We validate STATBench's emulation of STAT by comparing execution results from STATBench with previously collected data from STAT on the same platform. We then use STATBench to emulate STAT on configurations up to the full BG/L system size – at this scale, STATBench predicts latencies below three seconds.

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48 (UCRL-ABS-233113).

Scalable, Automated Performance Analysis with TAU and PerfExplorer

Kevin A. Huck and Allen D. Malony

Performance Research Laboratory
Computer and Information Science Department
University of Oregon, Eugene, OR, USA
E-mail: {khuck, malony}@cs.uoregon.edu

Abstract

Scalable performance analysis is a challenge for parallel development tools. The potential size of data sets and the need to compare results from multiple experiments presents a challenge to manage and process the information, and to characterize the performance of parallel applications running on potentially hundreds of thousands of processor cores. In addition, many exploratory analysis processes represent potentially repeatable processes which can and should be automated. In this paper, we will discuss the current version of PerfExplorer, a performance analysis framework which provides dimension reduction, clustering and correlation analysis of individual trails of large dimensions, and can perform relative performance analysis between multiple application executions. PerfExplorer analysis processes can be captured in the form of Python scripts, automating what would otherwise be time-consuming tasks. We will give examples of large-scale analysis results, and discuss the future development of the framework, including the encoding and processing of expert performance rules, and the increasing use of performance metadata.

Developing Scalable Applications with Vampir

Matthias S. Müller, Holger Brunst, Matthias Jurenz, Andreas Knüpfer, and Wolfgang E. Nagel

ZIH (Center for Information Services and HPC,
Technische Universität Dresden, 01162 Dresden
E-mail: {matthias.mueller, holger.brunst, matthias.jurenz}@tu-dresden.de
E-mail: {andreas.knuepfer, wolfgang.nagel}@tu-dresden.de

Abstract

With petaflop systems consisting of hundreds of thousands of processors at the high end and multi-core CPUs entering the market at the low end application developers face the challenge to exploit this vast range of parallelism by writing scalable applications. Any applied tool has to provide the same scalability. Since many performance problems and limitations are only reveilled at high processors counts this is especially true for performance analysis tools. At ZIH the tool Vampir for the analysis of large trace file was developed[1,2]. We perform some scalability studies with large trace files containing events from many thousand processors on one hand. The usability for real applications is analyzed with data collected with real applications, e.g. the thirteen applications contained in the SPEC MPI benchmark suite[3].

The analysis covers all phases of performance analysis: instrumenting the application, collecting the performance data, and finally viewing and analyzing the data. Examined aspects include instrumenting effort, monitoring overhead, trace file sizes, load time and response time during analysis.

1. H. Brunst and W. E. Nagel, *Scalable performance analysis of parallel systems: Concepts and experiences*, in: Parallel Computing: Software, Algorithms, Architectures Applications, G. R. Joubert, W. E. Nagel, F. J. Peters, and W. V. Walter, (eds.), pp. 737–744. Elsevier, (2003).
2. H. Brunst, D. Kranzlmüller, and W. E. Nagel, *Tools for scalable parallel program analysis - VAMPIR NG and DEWIZ*, in: Distributed and Parallel Systems, Cluster and Grid Computing, International Series in Engineering and Computer Science **777**, pp. 93–102, Kluwer, (2005).
3. M. S. Müller, M. van Waveren, R. Liebermann, B. Whitney, H. Saito, K. Kalyan, J. Baron, B. Brantley, Ch. Parrott, T. Elken, H. Feng, and C. Ponder, *SPEC MPI2007 - an application benchmark for clusters and hpc systems*, in: ISC2007, (2007).

Scalable Collation and Presentation of Call-Path Profile Data with CUBE

Markus Geimer[1], **Björn Kuhlmann**[2], **Farzona Pulatova**[3], **Felix Wolf**[1,4], **and Brian Wylie**[1]

[1] John von Neumann Institute for Computing, Forschungszentrum Jülich, 52425 Jülich, Germany
E-mail: {m.geimer, f.wolf, b.wylie}@fz-juelich.de

[2] SAP Deutschland AG & Co. KG, 69190 Walldorf, Germany
E-mail: bjoern.kuhlmann@sap.com

[3] National Instruments Corporation, Austin, TX 78759-3504, USA
E-mail: farzona@gmail.com

[4] RWTH Aachen University, Department of Computer Science, 52056 Aachen, Germany

Abstract

Developing performance-analysis tools for applications running on thousands of processors is extremely challenging due to the vast amount of performance data being generated. One aspect where this is particularly obvious is the visual presentation of analysis results. For example, interactive response times may become unacceptably long, the amount of data may exceed the available memory, or insufficient display size may prevent a meaningful presentation. Already writing the files to store the data for later presentation can consume a substantial amount of time on modern large-scale systems. In this talk, we describe how CUBE, a presentation component for call-path profiles, that is primarily used to display runtime summaries and trace-analysis results in the SCALASCA toolkit, has been modified to more efficiently handle data sets from thousands of processes. The modifications target both the scalable collation of input data files suitable for CUBE as well as the interactive display of the corresponding data.

Challenges addressed to increase scalability of the collation step include avoiding to write large numbers of files as well as memory limitations of individual nodes. Instead of writing one file per process, the process-local data is now centrally collected in small portions via MPI gather operations, which allow utilizing special network hardware, such as the global tree network of Blue Gene/L. The file itself is written incrementally by a single master node as the portions sent by other nodes arrive, minimizing the amount of data held in memory at a time.

The capability of the display to show large data sets is extended by reducing the memory footprint of the data and increasing the available memory. The reduction of the memory footprint is achieved through an optimization of the internal data structures used to hold the data. The amount of available memory is increased by using a remote server with a more generous memory configuration in combination with a graphical client running on the local desktop to store only the data currently on display.

Coupling DDT and Marmot for Debugging of MPI Applications

Bettina Krammer[1], Valentin Himmler[1], and David Lecomber[2]

[1] HLRS - High Performance Computing Center Stuttgart,
Nobelstrasse 19, 70569 Stuttgart, Germany
E-mail: {krammer, himmler}@hlrs.de

[2] Allinea Software,
The Innovation Centre, Warwick Technology Park, Gallows Hill,
Warwick, CV34 6UW, UK
E-mail: david@allinea.com

Abstract

Parallel programming is a complex, and since the multi-core era has dawned, also a more and more common task that can be alleviated considerably by tools supporting the application development and porting process. Therefore, we plan to couple existing tools, namely the MPI (Message Passing Interface) correctness checker Marmot[1], and the parallel debugger DDT, to provide MPI application developers with a powerful and user-friendly environment. So far, both tools have been used on a wide range of platforms as stand-alone tools to cover different aspects of correctness debugging. While (parallel) debuggers are great help in examining code at source level, e.g. by monitoring the execution, tracking values of variables, displaying the stack, finding memory leaks, etc., they give little insight into *why* a program actually gives wrong results or crashes when the failure is due to incorrect usage of the MPI API. To unravel such kinds of errors, the MARMOT library has been developed. The tool checks at run-time for errors frequently made in MPI applications, e.g. deadlocks, the correct construction and destruction of resources, etc., and also issues warnings in case of non-portable constructs.

In the final paper we will describe these two tools in more detail and report first experiences and results with their integration.

1. B. Krammer, M. S. Mueller, and M. M. Resch, *Runtime checking of MPI applications with MARMOT*, in: ParCo 2005, Malaga, Spain, September, NIC Series **35**, G. Joubert *et al.* (eds.) (2005).

Compiler Support for Efficient Profiling and Tracing

Oscar Hernandez and Barbara Chapman

Computer Science Department, University of Houston
501 Phillip G. Hoffman
4800 Calhoun, Houston, TX 77204-3010
E-mail: {oscar, chapman}@cs.uh.edu

Abstract

We are developing an integrated environment[2] for application tuning that combines robust, existing, open source software - the OpenUH compiler[1], Dragon program analysis tool and three performance tools, TAU, KOJAK and PerfSuite. As a result, we are able to accomplish a scalable strategy for performance analysis, which is essential if performance tuning tools are to address the needs of emerging very large scale systems. The performance tools provide different levels of detail of performance information but at given cost; being tracing the most accurate but expensive one.

We have discovered that one of the benefits of working with compiler technology is that it can direct the performance tools to decide which regions of code they should measure selectively combining both coarse grain (parallel region level, call path/procedure level) and fine grain regions (control flow level) of the code. Using the internal cost models in the compiler inter procedural analyzer, we can estimate the importance of a region by estimating cost vectors which includes its size and how often gets invoked. Using this analysis we can set different thresholds that a region must meet in order to be instrumented or not. This approach has shown to significantly reduce overheads to acceptable levels for both profiling and tracing. In this paper we present how the compiler helped to select the important regions of the code to measure in the NAS parallel benchmarks and in a weather code, significantly reducing its overhead by approximately 10 times, to acceptable levels within 5% of overhead. The goal of the system is to provide an automated, scalable performance measurement and optimization to increase user productivity by reducing the manual effort of existing approaches.

1. C. Liao, O. Hernandez, B. Chapman, W. Chen, and W. Zheng, *OpenUH: An Optimizing, Portable OpenMP Compiler*, Concurrency and Computation: Practice and Experience, (2007).
2. O. Hernandez, F. Song, B. Chapman, J. Dongarra, B. Mohr, S. Moore, and F. Wolf, *Performance Instrumentation and Compiler Optimizations for MPI/OpenMP Applications*, Second International Workshop on OpenMP, (2006).

Comparing Intel Thread Checker and Sun Thread Analyzer

Christian Terboven

Center for Computing and Communication,
RWTH Aachen University, 52074 Aachen, Germany
E-mail: terboven@rz.rwth-aachen.de

Abstract

Multiprocessor compute servers have been available for many years now. It is expected that the number of cores per processor chip will increase in the future and at least some multicore architectures will even support multiple threads running simultaneously. Hence, parallel programming will become more wide-spread and land on almost any programmer's desk. Both multicore systems and also larger SMP or ccNUMA systems can be programmed employing shared-memory parallelization paradigms.

Posix-Threads[1] and OpenMP[2] are the most wide-spread programming paradigms for shared-memory parallelization. At the first sight, programming for Posix-Threads or OpenMP may seem to be easily understandable. But for non-trivial applications, reasoning about the correctness of a parallel program is much harder[3] than for sequential control flow. The typical programming errors of shared-memory parallelization are Data Races, where the result of a computation is non-deterministic and dependent on the timing of other events, or Deadlocks, where two or more threads are waiting for each other. Finding those errors with traditional debuggers is hard, if not impossible.

This talk will compare the two software tools Intel Thread Checker[4] and Sun Thread Analyzer[5], that help the programmer in finding errors like Data Races and Deadlocks in multi-threaded programs. Experiences using these tools on OpenMP and Posix-Threads applications will be presented together with findings on the strenghts and limitations of each individual product. Recommendations for embedding such tools into the software development process will be given.

1. *IEEE: Portable Operating System Interface (POSIX), Part 1: System Application Program Interface (API)*, IEEE Std 1003, (1990).
2. *OpenMP Application Program Interface 2.5*, OpenMP Architecture Review Board, (2005).
3. H. Sutter, *The Free Lunch Is Over: A Fundamental Turn Toward Concurrency In Software*, Dr. Dobb's Journal, **30**, (March 2005).
4. *Intel Threading Analysis Tools: Intel Thread Checker 3.1*, Intel Corp., URL: `http://www3.intel.com/cd/software/products/asmo-na/eng/threading/index.htm`
5. *Sun Studio 12: Sun Thread Analyzer*, Sun Microsystems Inc., URL: `http://developers.sun.com/sunstudio/downloads/ssx/tha/`

Continuous Runtime Profiling of OpenMP Applications

Karl Fürlinger and Shirley Moore

Innovative Computing Laboratory,
EECS Department,
University of Tennessee
Knoxville, Tennessee, USA
E-mail: {karl, shirley}@eecs.utk.edu

Abstract

Profiling and tracing are the two common techniques for performance analysis of parallel applications. Profiling is often preferred over tracing because it gives smaller amounts of data, making a manual interpretation easier. Tracing, on the other hand, allows the full temporal behavior of the application to be reconstructed at the expense of larger amounts of performance data and an often more intrusive collection process.

In this paper we investigate the possibility of combing the advantages of tracing and profiling with the goal of limiting the data volume and enabling manual interpretation while retaining some temporal information about the program execution. Our starting point is a profiling tool for OpenMP applications called `ompP`[1]. Instead of capturing profiles only at the end of program execution ("one-shot" profiling), in the new approach profiles are captured at several points of time while the application executes. We call our technique *incremental* or *continuous* profiling and demonstrate its usefulness on a number of benchmark applications.

We discuss in general the dimensions of performance data and which new kind of performance displays can be derived by adding a temporal dimension to profiling-type data. Among the most useful new displays are *overheads over time* which allows the location of when overheads such as synchronization arise in the target application and *performance counter heatmaps*, that show performance counters for each thread over time.

1. K. Fuerlinger and M. Gerndt, *ompP: A profiling tool for OpenMP*, in: Proc. 1st International Workshop on OpenMP (IWOMP 2005), Eugene, Oregon, USA, (May 2005).

Understanding Memory Access Bottlenecks on Multi-core

Josef Weidendorfer

Institut für Informatik, Technische Universität München,
D-85747 Garching bei München, Germany
E-mail: josef.weidendorfer@cs.tum.edu

Abstract

This talk focuses on scalability and usability issues of analyzing the memory access behavior of multi-threaded applications on multi-core chips. The main objective is to help in the development of optimization strategies for application controlled prefetching agents running on dedicated cores, ensuring optimal exploitation of the limited connection to the main memory.

To reach this goal, the multi-core simulation collects metrics such as read/write bandwidth requirements and working set size of the threads as well as working set overlapping. The data is associated to the execution stream of the threads in an aggregated way, in order to pinpoint code regions where cache optimization is required, and where prefetch requests are useful to be handled by the prefetching agent.

Although the tool scenario does not target parallel systems with thousands of processors, the issues which needs to be solved regarding the amount of collected information, as well as regarding methods for easy to understand visualization, is quite related. For both, the amount of measurement data has to kept at a manageable size by using techniques for online aggregation. To allow quick browsing in the visualization tool, fast data structures have to be used with persistent indexing, as well as aggregation views with support for interactive selection and filtering of data.

The tool is being developed in the scope of the Munich Multicore Initiative[a] as an extension of the suite consisting of Callgrind, based on Valgrind[b], and KCachegrind[c], a visualization tool for profiling data. As it is work in progress, we focus on existing parts, active development issues and design alternatives.

1. J. Weidendorfer and C. Trinitis, *Block Prefetching for Numerical Codes*, Proceedings of 19th Symposium on Simulation Techniques (ASIM 2006), Hannover, Germany, September 12–14, (2006).
2. J. Weidendorfer, M. Kowarschik, and C. Trinitis, *A Tool Suite for Simulation Based Analysis of Memory Access Behavior*, Special Session of ICCS2004: Tools for Program Development and Analysis in Computational Science, Cracow, Poland, June 7–9, (2004).

[a] `mmi.cs.tum.edu`
[b] `www.valgrind.org`
[c] `kcachegrind.sf.net`

Mini-Symposium

"DEISA: Extreme Computing in an Advanced Supercomputing Environment"

Wednesday, September 5, 2007

DEISA: Enabling Cooperative Extreme Computing in Europe

Victor Alessandrini

CNRS/IDRIS, Bâtiment 506, BP 167
91403 Orsay cedex, France
E-mail: va@idris.fr

Abstract

The DEISA European Research Infrastructure was initially designed to act as a vector of integration of High Performance Computing (HPC) resources at the continental scale. Its services have been tailored to enable seamless access to, and high performance cooperative operation of, a distributed park of leading supercomputing platforms in Europe.

The DEISA services are deployed on top of a dedicated high speed network infrastructure connecting computing platforms, using selected middleware. Their primordial objective is enabling capability computing across remote computing platforms and data repositories. Workflows across different platforms, transparent remote I/O, large file transfers, are starting to operate without inducing performance bottlenecks that would invalidate high performance computing. These services will bloom as the number of scientific users accessing different computing platforms in Europe will grow.

After quickly reviewing the existing services and the service provisioning mode of the DEISA research Infrastructure based today on the DEISA Extreme Computing Initiative, we will discuss how DEISA has been paving the way to the deployment of a coherent HPC environment in Europe, and why their persistency is mandatory to support and cooperate with new initiatives like PACE in the area of HPC. Some comments will be advanced about the relevance of the DEISA environment for the efficient operation of future European supercomputers, and the current vision about the overall role of DEISA in the new emerging European HPC ecosystem will be discussed.

Effective Methods for Accessing Resources in a Distributed HPC Production System

Andrea Vanni

System and Technologies Department,
CINECA, 40033 Casalecchio di Reno BO, Italy
E-mail: a.vanni@cineca.it

Abstract

DEISA is a consortium of leading national supercomputing centres in Europe that are jointly building and operating a distributed terascale supercomputing facility. This objective is being attained by a deep integration using modern Grid technologies - of existing national high performance computing infrastructures A fundamental objective of the DEISA Consortium is to deploy a production quality, persistent, Grid enabled pan-European supercomputing environment that will act as the integrating infrastructure for High Performance Computing in Europe. DEISAs technology choices, based on the concept that integration of resources and services, add value to existing infrastructures.

eDEISA project aims to improve the DEISA infrastructure extending the network connectivity infrastructure, the middleware services, user support and application services.

Leading scientists across Europe may exploit the bundled supercomputing power and the related global data management infrastructures in a coherent and comfortable way. A special focus is set on grand challenge applications from scientific key areas like material sciences,climate research, astrophysics, life sciences, fusion oriented energy research.

DEISA provides multiple means for managing user jobs and data in a efficient and secure way and software facilities for Grids interoperabilty. Challenging applications can take advantage of global parallel filesystem installed on a 10Gbit network, LoadLevel Multicluster Batch Sytstem and a Teraflops infrastructure. Users may exploit graphical user interface for creating their job with UNICORE or with the DEISA life science web portal. Unicore provides support for complex user jobs workflow as well. Again, DEISA supplies commandline instruments for job submission and data federation.

Last but not least, DEISA is a open Grid infrastructure that permits users to collaborate and interoperate with world Grids. This achievement has been achieved by tuning, closely, Globus toolkit in the DEISA network.

1. DEISA consortium, *DEISA technical annex*, December 2006.
2. DEISA consortium, *eDEISA technical annex*, March 2006.

GPFS - A Cluster File System

Klaus Gottschalk

IBM Germany
Pascalstr. 100
70569 Stuttgart
Deutschland
E-mail: gottschalk@de.ibm.com

Abstract

The IBM General Parallel File System (GPFS) is a reliable and proven parallel high throughput file system for parallel computers. Recent added functions for information lifecycle management and multi-cluster data access did evolve GPFS into a data management solution for data centric computing between clusters of clusters within a data center or even between remote WAN connected data centers.

With these functions GPFS is the global file system that enables global data access in the DEISA consortium. This talk explains the concepts of GPFS and highlights functions that will be added in the upcoming GPFS V3.2 release.

Development Strategies for Modern Predictive Simulation Codes

Alice Koniges, Robert Anderson, Aaron Fisher, Brian Gunney, Nathan Masters

Lawrence Livermore National Laboratory,
Livermore, CA 94550,
U.S.A.
E-mail: koniges@llnl.gov

Abstract

Modern simulation codes often use a combination of languages and libraries for a variety of reasons including reducing time to solution, automatic parallelism when possible, portability, and modularity. We describe the process of designing a new multiscale simulation code, which takes advantage of these principles. One application of our code is high-powered laser systems, where hundreds of laser beams are concentrated on dime-sized targets to enable the demonstration of controlled fusion and exciting new experiments in astrophysics and high-energy-density science. Each target must be carefully analyzed so that debris and shrapnel from the target will be acceptable to optics and diagnostics, creating new simulation regimes. These simulations rely on a predictive capability for determining the debris and shrapnel effects. Our new three-dimensional parallel code uses adaptive mesh refinement (AMR) combined with more standard methods based on Arbitrary Lagrangian Eulerian (ALE) hydrodynamics to perform advanced modeling of each different target design. The AMR method provides a true multiscale simulation that allows for different physical models on different scales.

We discuss our code development strategies. The code is built on top of the SAMRAI library (structured adaptive mesh refinement application interface) that provides scalable automatic parallelism. During the development phase of this code we have instituted testing procedures, code writing styles, and team coordination applicable to a rapidly changing source code, several dependent libraries, and a relatively small team including university collaborators with their own source and platforms. We use modern coding techniques and open source tools when possible for code management and testing including CppUnit, Subversion (and previously GNU ARCH), TiddlyWikki and group chat rooms. Additionally, we are conducting experiments aimed at providing a data set for validation of the fragmentation models. We describe our tools and our efforts in the area of code verification and validation.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. UCRL-ABS-233551

Submission Scripts for Scientific Simulations on DEISA

Gavin J. Pringle, Terry M. Sloan, Elena Breitmoser, and Odysseas Bournas and Arthur S. Trew

EPCC,
University of Edinburgh
E-mail: {g.pringle, t.sloan, e.breitmoser, o.bournas, a.trew}@epcc.ed.ac.uk

Abstract

The DEISA Services for the Heterogeneous management Layer, better known as the DESHL, allows users and their applications to manage batch jobs and data across a computing Grid in a uniform manner regardless of the underlying hardware and software on the various nodes. The DESHL employs emerging Grid standards so that a user and their applications are not affected by the differences or changes in the hardware and software configuration on the Grid nodes.

The DESHL provides a means for coordinating and integrating services for distributed resource management in a heterogeneous supercomputing environment. It provides users and their applications with a command line tool and application programming interfaces for job and data management in DEISA's UNICORE-based grid.

The DESHL employs the SAGA (Simple API for Grid Applications) and JSDL (Job Submission Description Language) emerging grid standards as well as the Open Group Batch Environment Services specification. Reliance on Grid standards will allow interoperability with other Grid environments.

The job management and file transfer capabilities of the DESHL command line tool include: determining the sites to which a user can submit a batch job to; submitting batch jobs; terminating batch jobs; viewing the status of batch jobs; uploading/downloading files; deleting files; determining if files exist; renaming files; and, copying/moving files between sites.

The DESHL supports the command line tool operations through a SAGA-based API, the DESHL Client library. This SAGA API interfaces with the more powerful and complex Grid Access library (also known as Roctupus) that in turn interacts with the low-level UNICORE specific library, ARCON. Any of these APIs can be used by Java applications developers as appropriate.

Three DESHL-based bash scripts have been created which demonstrate how DEISA can be exploited directly from the user's workstation. These scripts are for Task Farms, Workflows, and for Code Migration/Resubmission.

Task Farms are currently not possible with UNICORE, however, users can employ the DEISA infrastructure as a vast Task Farm environment via our DESHL script, which uses their workstation as the Task Farm Master, and exploits the DEISA platforms as a Task Farm workforce.

Whilst UNICORE can create Workflows, some users no not wish to, or find it difficult to, manipulate a workstation via using a mouse. Further, some users are simply more familiar with accessing HPC resources via the command line, thus we have developed a DESHL-based workflow script template.

Lastly, we have developed Code Migration/Resubmission scripts. Here we define Code Migration as a very large simulation which starts running on one particular DEISA platform and finishes on another, different platform of a potentially different vendor. Code Resubmission is where only one platform is employed. Allowing simulations to either migrate or to resubmit itself is key for large simulations which require more time to run than any single batch job permits.

Application Enabling in DEISA: Hyperscaling of Turbulence Codes Supporting ITER

Hermann Lederer[1], **Reinhard Tisma**[1], **Roman Hatzky**[1],
Alberto Bottino[2], **Frank Jenko**[2]

[1] Rechenzentrum Garching der Max-Planck-Gesellschaft,
Boltzmannstr. 2, D-85748 Garching
E-mail: {lederer, tisma, hatzky}@rzg.mpg.de

[2] Max-Planck-Institut für Plasmaphysik,
Boltzmannstr. 2, D-85748 Garching
E-mail: {jenko, bottino}@ipp.mpg.de

Abstract

The ITER experiment (International Thermonuclear Experimental Reactor) will have to be accompanied by challenging numerical plasma turbulence simulations. Due to the high demands for compute power and memory consumption, simulations must be capable of using tens of thousands of processor-cores simultaneously. Highly scalable applications will be mandatory.

The EU DEISA project (Distributed European Infrastructure for Supercomputing Applications) has developed and put into operation a grid of the most powerful supercomputing platforms in Europe. But DEISA also provides advanced application enabling support through a team of European experts, the Applications Task Force, and through Joint Research Activities.

Through a joint effort of DEISA specialists and scientists engaged in the theory support for ITER, two important European simulation codes for core turbulence, ORB5 and GENE, could be adapted for portable usage within the heterogeneous DEISA infrastructure. Moreover, the codes were deeply analyzed, bottlenecks were identified and removed, and, most important, the scalability of the codes could be extraordinarily enhanced.

Through application of the domain cloning concept, the PIC code ORB5 was enabled for high scalability. Efficient usage of ORB5 code could be demonstrated up to 8 k processors, both on a Cray XT3 and on an IBM BlueGene/L system.

GENE was parallelized through domain decomposition of the five-dimensional problem grid to such a high degree that close to efficiency loss-free usage up to 32 k processors of an IBM BlueGene/L machine was achieved. Results combined from both strong and weak scaling measurements indicate an even higher scalability potential of GENE. Extrapolations suggest an efficient usage on up to the order of 1 M processor-cores of a similarly scalable future HPC architecture, representing a milestone on the way towards realistic core turbulence simulations of future fusion devices like ITER.

First Principles Simulations of Plasma Turbulence within DEISA

Frank Jenko, Alberto Bottino, Tobias Görler, and Emanuele Poli

Max-Planck-Institut für Plasmaphysik
EURATOM Association
85748 Garching
Germany
E-mail: fsj@ipp.mpg.de

Abstract

In recent years, the world has become increasingly aware of the fact that we are in urgent need of energy resources which are free of CO_2 emission. Magnetic confinement fusion aims at contributing to the solution of this problem. However, the success of the international fusion experiment ITER (currently under construction in Southern France) will depend to a large degree on the value of the so-called energy confinement time. Two of the most advanced (first principles based) tools in the world describing the underlying physical processes are the plasma turbulence codes `GENE` and `ORB5`. Some of the outstanding issues in fusion research addressed with these codes in the framework of the DECI project GYROKINETICS will be described.

Heavy Particle Transport in Turbulent Flows

Alessandra S. Lanotte[1], Luca Biferale[2], Jérémie Bec[3], Massimo Cencini[4], Stefano Musacchio[5], and Federico Toschi[6]

[1] CNR-ISAC and INFN, Lecce, Italy
E-mail: a.lanotte@isac.cnr.it

[2] Dept. Physics, Univ. Roma II and INFN, Rome, Italy
[3] CNRS UMR6202, OCA, Nice, France
[4] SMC-INFM and CNR-ISC, Rome, Italy
[5] The Weizmann Institute of Science, Rehovot, Israel
[6] CNR-IAC, Rome and INFN, Ferrara, Italy

Abstract

Dust, droplets and other finite-size impurities with a large mass density suspended in incompressible flows are commonly encountered in many natural phenomena, such as the growth of raindrops in sub-tropical clouds, and industrial processes. One of the most salient feature of such suspensions is the presence of strong inhomogeneities in the spatial distribution of particles. This phenomenon, known as *preferential concentration* [1] can strongly modify the probability to find particles close to each other and thus have influence on their possibility to interact (chimically, biologically, ..) or collide. Its statistical description largely remains an open question.

We present the results from Direct Numerical Simulations (DNS) of high-resolution turbulent flows, seeded with millions of particles much heavier than the carrier fluid. Flow and particles dynamics are fully parallel; a differential time dumping of particles motion is applied to have both highly resolved trajectories, and a sufficiently large statistical dataset.

We characterize some aspects of the particles dynamics and statistics at varying both the flow turbulence and the particle inertia. An important observation is that, beside clusters, the particle distribution presents large voids where the mass is orders of magnitude below its average. Such regions are typically correlated with the vortical structures of the flow, meaning that eddies act as small centrifuges and eject heavy particles leading to their concentration in the flow regions dominated by strain. Clusters and voids of different typical sizes have a signature on the coarse-grained particles mass distribution, which can be partly explained in terms of a simple model[2].

1. J. K. Eaton and J. R. Fessler, *Preferential concentration of particles by turbulence*, Int. J. Multiph. Flow **20**, 169–209, (1994).
2. J. Bec, L. Biferale, M. Cencini, A. Lanotte, S. Musacchio, and F. Toschi, *Heavy Particle Concentration in Turbulence at Dissipative and Inertial Scales*, Phys. Rev. Lett. **98**, 084502, (2007).

Membranes Under Tension: Atomistic Modeling of the Membrane-Embedded Synaptic Fusion Complex

Marc Baaden

Laboratoire de Biochimie Théorique
Institut de Biologie Physico-Chimique,
CNRS UPR 9080, 13, rue Pierre et Marie Curie
F-75005 Paris, France
E-mail: baaden@smplinux.de

Abstract

The SNARE protein complex is central to membrane fusion, a ubiquitous process in biology. Modeling this system in order to better understand its guiding principles is a challenging task. This is mainly due to the complexity of the environment: two adjacent membranes and a central bundle of four helices made up by vesicular and plasma membrane proteins as shown in Fig. 1. Not only the size of the actual system but also the computing time required to equilibrate it render this a demanding task requiring exceptional computing resources. Within the DEISA Extreme Computing Initiative, we have performed 40 ns of atomistic molecular dynamics simulations with an average performance of 81.5 GFlops on 96 processors using 218 000 CPU hours. These simulations establish a realistic microscopic view of the membrane-embedded synaptic fusion complex.

1. `http://www.baaden.ibpc.fr/projects/snaredeci/`

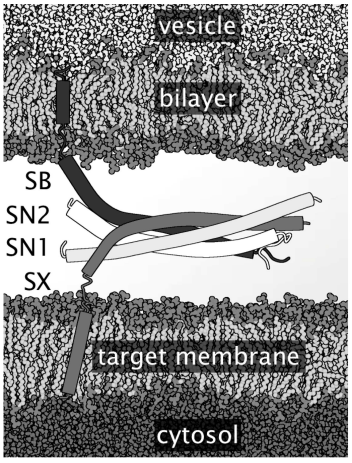


Figure 1. Illustration of the simulation system. The atomistic model comprises 339 792 atoms consisting of 4 proteins (346 amino acids), two lipid bilayers (1008 lipids), 296 Na^+ ions, 166 Cl^- ions and 92 217 water molecules. The simulations were carried out with the Gromacs software (http://www.gromacs.org).

Mini-Symposium

“Parallel Computing with FPGAs”

Thursday, September 6, 2007

Parallel Computing with Low-Cost FPGAs: A Framework for COPACOBANA

Tim Güneysu[1], Christof Paar[1], Jan Pelzl[3], Gerd Pfeiffer[2], Manfred Schimmler[2], and Christian Schleiffer[3]

[1] Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
E-mail: {gueneysu, cpaar}@crypto.rub.de

[2] Institute of Computer Science and Applied Mathematics, Faculty of Engineering, Christian-Albrechts-University of Kiel, Germany
E-mail: {gp, masch}@informatik.uni-kiel.de

[3] escrypt GmbH - Embedded Security, Bochum, Germany
E-mail: {jpelzl, cschleiffer}@escrypt.com

Abstract

In many disciplines such as applied sciences or computer science, computationally challenging problems demand for extraordinary computing power, mostly provided by super computers or clusters of conventional desktop CPUs. During the last decades, several flavors of super computers have evolved, most of which are suitable for a specific type of problem. In general, dedicated clusters and super computers suffer from their extremely high cost per computation and are, due to the lack of cheap alternatives, currently the only possible solution to computational hard problems. More recently, emerging low-cost FPGAs tend to be a very cost-effective alternative to conventional CPUs for solving at least some of the computational hard problems such as those appearing in cryptanalysis and bio-informatics.

In cryptanalysis, breaking symmetric or asymmetric ciphers is computationally extremely demanding. Since the security parameters (in particular the key length) of almost all practical crypto algorithms are chosen such that attacks with conventional computers are computationally infeasible, the only promising way to tackle existing ciphers (assuming no mathematical breakthrough) is to build special-purpose hardware. Dedicating those machines to the task of cryptanalysis holds the promise of a dramatically improved cost-performance ratio so that breaking of commercial ciphers comes within reach.

This contribution presents the realization of a very generic framework for the COPACOBANA (Cost-Optimized Parallel Code Breaker) machine. COPACOBANA consists of up to 120 low-cost FPGAs and can be realized for US$ 10,000 while being able to outperform conventional computers by several orders in magnitude, depending on the application. The presented framework allows for a simple distribution of parallelizable tasks and corresponding control mechanisms. With the framework at hand, the overhead and, as a consequence, the time to deploy custom made applications on the COPACOBANA platform is minimized. Exemplarily, we show how cryptanalytical applications can be based on top of the framework, resulting in a very low cost per computation.

Accelerating the Cube Cut Problem with an FPGA-Augmented Compute Cluster

Tobias Schumacher, Enno Lübbers, Paul Kaufmann, and Marco Platzner

Paderborn Center for Parallel Computing (PC^2)
University of Paderborn
E-mail: {tobe, luebbers, paulk, platzner}@uni-paderborn.de

Abstract

The cube cut problem refers to determining the minimum number $C(d)$ of hyperplanes that slice all edges of the d-dimensional hypercube and has numerous applications in the domain of optimization theory, e.g. in integer linear programming (ILP). While it is known that for $d \leq 5$ exactly d slices are needed, the exact numbers for most $d > 5$ are unknown. An analytical solution for the problem of determining $C(d)$ appears to be extremely difficult to find. Up to now, $C(d)$ has only be proven for rather small values of d despite the substantial effort that went into that problem.

A different and computational approach uses *exhaustive search* and evaluates all possible slices of the d-dimensional hypercube to find the smallest set that slices all edges. The corresponding algorithm consists of three phases, where the most time-consuming phase requires a huge number of comparisons between mostly independent bitstrings. This is a class of computations that general-purpose microprocessors are not particularly well-suited for. The cube cut algorithm lends itself to a parallel implementation on fine-grained reconfigurable logic, such as FPGAs. Additionally, the algorithm reveals sufficient data parallelism which can be exploited by distributing the computations onto a modern compute cluster with FPGA-equipped nodes.

While the design of a fine-grained computing unit for bitstring comparisons is straight-forward, determining an overall CPU-FPGA accelerator node for efficient dataflow-driven execution is challenging. Depending on the application's behavior and the target's architecture and technology, a proper dimensioning of the different data channels (host bus, FPGA bus interfaces, FIFOs) and parallel computation units (in terms of width and depth of the execution pipeline) is required for maximum performance. We support this task by a model-based approach. In this paper we propose such an approach based on the LDA (latency of data access) model. The original LDA model has been designed to effectively describe not only the temporal behavior of an algorithm, but also to take memory accesses across various hierarchies (e.g. multi-level caches) into account. We extend LDA to fine-grained parallel architectures as commonly applied in FPGA designs. In contrast to an ad-hoc design of an FPGA accelerator, such a model-based approach facilitates a structured design-space exploration and allows for a quick adaptation of the accelerator architecture to different technologies. We outline the underlying ideas of the model-based approach and its iterative application to the design of an FPGA accelerator for the cube cut problem. The resulting implementation on one hybrid CPU/FPGA node (Xeon 3.2GHz/4GB, Virtex-IIPro 2VP70) achieves a speedup of 27 over a CPU-only node; a 4-node hybrid cluster accelerates the problem by a factor of 105 over a single CPU.

A Run-Time Reconfigurable Cache Subsystem

Fabian Nowak, Rainer Buchty, and Wolfgang Karl

Universität Karlsruhe (TH), Institut für Technische Informatik (ITEC)
76128 Karlsruhe, Germany
E-mail: {nowak, buchty, karl}@ira.uka.de

Abstract

In traditional hardware architectures, hardware parameters are defined with respect to given application scenarios. For general purpose architectures, this results in a compromise-based architecture serving best for that very application set. This is especially true for the cache subsystem and may hamper high performance execution of high-performance computing applications which reportedly show distinct phases of certain cache access behavior[1]. Such applications would greatly benefit from a reconfigurable cache infrastructure, which can be reconfigured on-the-fly to match the current requirements. Also energy efficiency can be improved using dedicated cache architectures[2,3].

Aim of the presented work is to create a versatile hardware infrastructure for cache performance analysis. Unlike off-line analysis such an infrastructure will enable real-time monitoring of running applications, instead of off-line analysis of a more or less reduced trace. Reconfiguration of the cache infrastructure introduces certain problems which we will address in this paper. We will outline the problems of a potentially reconfigurable cache subsystem, and further prove the feasibility of such a system and how to overcome the individual problems. Finally, we will present a hardware prototype implementation of our approach.

The proposed reconfigurable cache architecture (RCA) enables run-time changes of several cache parameters. Due to hardware implications, the maximum value of certain design parameters – such as e.g. overall cache size or the maximum amount of associativity – is already set before synthesis. We were able to validate our hardware design and to successfully demonstrate changing cache hardware parameters such as cache associativity, write allocation, as well as write-back and replacement strategy.

1. J. Tao and W. Karl, *Optimization-oriented visualization of cache access behavior*, in: Proc. International Conference on Computational Behavior, Lecture Notes in Computer Science **3515**, pp. 174–181, Springer, (2005).
2. A. Gordon-Ross, C. Zhang, F. Vahid, and N. Dutt, *Tuning caches to applications for low-energy embedded systems*, in: Ultra Low-Power Electronics and Design, E. Macii, (ed.), Kluwer Academic Publishing, (2004).
3. A. Gordon-Ross and F. Vahid, *Dynamic optimization of highly configurable caches for reduced energy consumption*, Riverside ECE Faculty Candidate Colloquium, (March 2007).

Novel Brain-Derived Algorithms Scale Linearly with Number of Processing Elements

Jeff Furlong[1], Andrew Felch[2], Jayram Moorkanikara Nageswaran[1], Nikil Dutt[1], Alex Nicolau[1], Alex Veidenbaum[1], Ashok Chandrashekar[2], and Richard Granger[2]

[1] University of California, Irvine,
Irvine, CA 92697, USA
E-mail: {jfurlong, jmoorkan, dutt, nicolau, alexv}@ics.uci.edu

[2] Dartmouth College
Hanover, NH 03755, USA
E-mail: {andrew.felch, ashok.chandrashekar, richard.granger}@dartmouth.edu

Abstract

Algorithms are often sought whose speed increases as processors are added, yet attempts at such parallelization typically result in little speedup, due to serial dependencies intrinsic to many algorithms. We here introduce a novel class of algorithms that exhibit intrinsic parallelism, increasing their speed with no diminishing returns as processors are added. The algorithms are derived from the brain circuitry of visual processing. Given the brain's ability to outperform engineering approaches on a range of visual and auditory perceptual tasks, these algorithms have been studied in attempts to imitate the successes of brain circuits on these tasks. These algorithms are slow on serial processors, but as might be expected of algorithms derived from highly parallel brain architectures, their lack of internal serial dependencies makes them highly suitable for efficient implementation across multiple processors. Here we describe these methods, show their efficacy on a set of visual benchmarks of known difficulty, and demonstrate the characteristics of their parallel implementation on both single and multiple FPGAs. Implementation on a single Xilinx Virtex 4 FPGA system gave rise to a speedup of 62x in performance, and 2500x improvement in performance per watt of power used. We show results indicating that multiple parallel FPGAs can achieve significantly higher performance improvements, and in particular that these improvements exhibit desirable scaling properties, increasing linearly with the number of processors added. Since linear scaling renders these solutions applicable to arbitrarily large applications, the findings may provide a new class of novel approaches for many domains, such as embedded computing and robotics, that require compact, low-power, fast processors.

Programmable Architectures for Realtime Music Decompression

Martin Botteck[1], Holger Blume[2], Jörg von Livonius[2], Martin Neuenhahn[2], and Tobias G. Noll[2]

[1] Nokia Research Center,
Nokia GmbH, Meesmannstraße 103, 44807 Bochum
E-mail: martin.botteck@nokia.com

[2] Chair for Electrical Engineering and Computer Systems
RWTH Aachen University, Schinkelstrasse 2, 52062 Aachen
E-mail: {blume, livonius, neuenhahn, tgn}@eecs.rwth-aachen.de

Abstract

Music playback has become a distinguishing and indispensable feature in modern mobile communication devices. A previous study[1] has modeled and analysed embedded general purpose processors such as the ARM 940T processor core which is frequently used in embedded applications and modeled their power consumption for several algorithms incl. MP3 and AAC decoding. Although such embedded processors yield attractive low power consumption the absolute values are still not low enough in order to support radical improvements for future mobile devices. Therefore, further implementation alternatives have to be studied retaining the flexibility of such platforms in order to be able to e.g. adapt to changes in coding standards etc.

While programmable processor architectures provide less power- and area-efficiency than other architecture blocks they have a high flexibility. The best power- and area-efficiency is achieved by physically optimised macros. However, they provide no flexibility.

FPGAs present an attractive compromise between these two extremes as they allow for highly parallelised implementations while preserving in-system reconfigurability at moderate implementation cost. This paper analyses implementation efficiency for MP3 decoding on a FPGA. The results of an exemplary FPGA implementation of a complete MP3 decoder featuring arithmetic datapaths as well as control overhead are compared concerning performance and power consumption to further implementation alternatives.

1. H. Blume, D. Becker, M. Botteck, J. Brakensiek, T. G. Noll, *Hybrid Functional and Instruction Level Power Modeling for Embedded Processor Architectures*, in: Proc. Samos 2006 Workshop, Samos, Greece, LCNS **4017**, pp. 216–226, Springer (2006).

The HARWEST High Level Synthesis Flow to Design a Special-Purpose Architecture to Simulate the 3D Ising Model

Alessandro Marongiu[1 2] **and Paolo Palazzari**[1 2]

[1] Ylichron Srl,
Via Anguillarese, 301,
00123 S. Maria di Galeria (Rome), Italy
E-mail: {*a.marongiu, p.palazzari*}*@ylichron.it*

[2] ENEA - Computing and Modelling Unit,
Via Anguillarese, 301,
00123 S. Maria di Galeria (Rome), Italy

Abstract

The 3D-Ising model is a mathematical model in statistical mechanics, used to model basic phenomena as the paramagnetic-ferromagnetic transition. While the 2D Ising model admits an analytical solution, the more interesting 3D case can be solved only with computationally intensive numerical methods. Due to its numerical intractability, many Monte Carlo methods have been proposed to solve the 3D Ising model, as the Demon and the Heat-Bath algorithms. In this work we refer to the Heat-Bath algorithm and we show how it can be efficiently implemented on an FPGA-based dedicated architecture through the adoption of the HARWEST High Level Synthesis design flow, developed by Ylichron Srl.
The HARWEST design flow, which allows to extract the parallelism from an ANSI C program and to generate the syntesizable VHDL of a parallel architecture which implements the original algorithm, is described in all its phases (from the translation of the C program into the Control Data FLow Graphs (CDFG) and System of Affine Recurrence Equations (SARE) computing models up to the generation of the synthesizable VHDL).
We report the results achieved implementing, with the HARWEST design flow, the 3D Ising Model into an FPGA. Such a dedicated implementation shows a speedup factor ranging from 1 to 3 order of magnitudo with respect to some optimized implementations on current high-end processors.

1. S. G. Brush, *History of the Lenz-Ising Model*, Reviews of Modern Physics **39**, 883–893, (1967).

Towards an FPGA Solver for the PageRank Eigenvector Problem

Séamas McGettrick, Dermot Geraghty, and Ciarán McElroy

Dept of Mechanical and Manufacturing Engineering,
Trinity College Dublin, Ireland
E-mail: {*mcgettrs, tgerghty, ciaran.mcelroy*}*@tcd.ie*

Abstract

Calculating the Google PageRank eigenvector is a massive computational problem dominated by Sparse Matrix by Vector Multiplication (SMVM) where the matrix is very sparse, unsymmetrical and unstructured. The computation presents a serious challenge to general-purpose processors (GPP) and the result is a very lengthy computation. Other scientific applications have performance problems with SMVM and FPGA solutions have been proposed. However, the performance of SMVM architectures is highly dependent on the matrix structure. Internet link matrices contain on average 10 non-zero elements per row[1]. Architectures for SMVM designed for other scientific areas may be unsuitable for use with these very sparse Internet link matrices. An FPGA architecture can bring advantages such as custom numeric units, increased memory bandwidth and parallel processing to bear on a problem and can be deployed as a coprocessor to a GPP.

In this paper, we investigate the SMVM performance of a modern Intel processor for Internet link matrices. We show that it only achieves a fraction of its peak performance. We evaluate the performance of two FPGA based SMVM architectures originally designed for finite element problems. The architectures are implemented on a Virtex II FPGA. We present these results and extrapolate them to show expected performance on a Virtex 5 FPGA. We demonstrate the effect of matrix reordering on these matrices using Reverse Cuthill McKey (RCM) reordering. We show that RCM increases performance in some cases. This highlights the need for a more in depth investigation into reordering in Internet link matrices. Finally, we investigate the possibility of outperforming the GPP using parallelization of processing units. This shows that FPGA based SMVM can perform better than SMVM on the GPP. It is projected that one of the FPGA based solvers could achieve 1450MFLOPS when implemented on a Virtex 5 FPGA. This should be further increased by adapting the design specifically for Internet link matrices.

1. A. N. Langville, C.D Meyer, *Google's PageRank and Beyond, The Science of Search Engine Rankings*, Princeton University Press, (2006).

Author Index

A

Ahn, Dong H. 155
Aldinucci, Marco 58, 100
Alessandrini, Victor 167
Aliaga, José I. 65
Alvarez, José Antonio 31
an Mey, Dieter 141
Anderson, Robert 170
Anshus, Otto J. 121
Arai, Yusuke 128
Arnold, Dorian C. 155
Arnold, Guido 20
Arnold, Lukas 109
Atienza, David 116

B

Bücker, H. Martin 76, 127
Baaden, Marc 175
Badía, José M. 51
Bader, Michael 35
Badia, Rosa M. 14
Barker, Kevin J. 24
Barrault, Maxime 9
Bec, Jérémie 174
Beetz, Christoph 109
Behr, Marek 5, 93
Belletti, Francesco 47
Bencteux, Guy 9
Benner, Peter 51
Bernreuther, Martin 19
Berrendorf, Rudolf 69
Berthold, Jost 13
Biferale, Luca 174
Birkeland, Tore 70
Bischof, Christian H. 76
Bjørndalen, John Markus 121
Blanco, Vicente 25
Blume, Holger 183
Boeck, Thomas 111
Boldarev, Alexei 110
Boldyrev, Sergei 110
Bollhöfer, Matthias 65
Bonacic, Carolina 103
Botteck, Martin 183
Bottino, Alberto 172, 173
Boullón, Marcos 25
Bounanos, Stylianos 104
Bournas, Odysseas 171
Breitmoser, Elena 171
Brunst, Holger 157
Buchholz, Martin 19
Buchty, Rainer 181
Bui, Van 140
Bull, J. Mark 137
Bungartz, Hans-Joachim 19
Buzzard, Gregery T. 149

C

Cabaleiro, José Carlos 25
Cacitti, Manuel 100
Cancès, Eric 9
Casas, Marc 14
Castillo, Maribel 51
Cencini, Massimo 174
Chandrashekar, Ashok 182
Chapman, Barbara 4, 140, 160
Clematis, Andrea 29, 71, 92
Coleman, Thomas F. 75
Corbalan, Julita 115
Cotallo, Maria 47
Crngarov, Ace 69
Cruz, Andres 47

D

D'Agostino, Daniele 29, 71, 92
D'yachenko, Sergei 110
Danelutto, Marco 58, 100
Davis, Kei 24
de Supinski, Bronis R. 155
Desplat, Jean Christophe 146
Drakenberg, N. Peter 82
Dreher, Jürgen 109
Duarte, Angelo 97
Dutt, Nikil 182
Dümmler, Jörg 81

E

Eberhard, Peter 17
Eicker, Norbert 87

F

Falcou, Joel 59
Fan, Shiqing 117
Faßbender, Heike 51

Felch, Andrew 182
Fernández, José Jesús 31
Fernández, Luis Antonio 47
Fisher, Aaron 170
Fleissner, Florian 17
Fleury, Martin 104
Fox, Jeffrey J. 149
Furlong, Jeff 182
Fürlinger, Karl 162

G

Görler, Tobias 173
Güneysu, Tim 179
Gómez, Antonio 99
Galizia, Antonella 71, 92
García, Carlos 64
García, José M. 99
Gasilov, Vladimir 110
Geimer, Markus 158
Geraghty, Dermot 185
Gerndt, Michael 23
Gervaso, Alessandro 100
González, Alberto 52
Gopal, Srinivasa M. 131
Gopalkrishnan, Pradeep 140
Gordillo, Antonio 47
Gottschalk, Klaus 169
Granger, Richard 182
Grauer, Rainer 109
Guim, Francesco 115
Gunney, Brian 170

H

Ha, Phuong Hoai 121
Hager, William 9
Hanigk, Sebastian 35
Hatzky, Roman 172
Hermanns, Marc-André 69
Hernandez, Oscar 140, 160
Himmler, Valentin 159
Hofmann, Michael 105
Homann, Holger 109
Honecker, Andreas 63
Huck, Kevin A. 156
Huckle, Thomas 35
Hülsemann, Frank 53

J

Janda, Rick 86
Janoschek, Florian 18
Jenko, Frank 172, 173
Jordan, Kirk E. 151
Jurenz, Matthias 157

K

Karl, Wolfgang 181
Kartasheva, Elena 110
Kaufmann, Paul 180
Keller, Rainer 117
Kerbyson, Darren J. 24
Kessler, Christoph W. 57
Kilpatrick, Peter 58
Klenin, Konstantin V. 131
Knüpfer, Andreas 157
Knaepen, Bernard 111
Knafla, Bjoern 43
Koch, Erik 147
Koniges, Alice 170
Krammer, Bettina 159
Krasnov, Dmitry 111
Krieg, Stefan 133
Krishnan, Manoj 98
Krusche, Peter 37
Kufrin, Rick 140
Kuhlen, Torsten 123
Kuhlmann, Björn 158
Kunis, Raphael 81

L

Löwe, Welf 57
Lübbers, Enno 180
Labarta, Jesús 14, 115
Lanotte, Alessandra S. 174
Le Bris, Claude 9
Lecomber, David 159
Lederer, Hermann 172
Lee, Gregory L. 155
Lee, Kyu H. 131
Leopold, Claudia 41–43
Lippert, Thomas 20, 87
Loogen, Rita 13
Luque, Emilio 97

M

Müller, Matthias S. 86, 157
Maiorano, Andrea 47
Malony, Allen D. 156
Mantovani, Filippo 47
Marin, Mauricio 103
Marinari, Enzo 47
Marongiu, Alessandro 184
Martín-Mayor, Victor 47

Martínez, Diego R. 25
Martínez, Pablo 122
Martínez-Zaldívar, Francisco-Jose 52
Martín, Alberto F. 65
Maruyama, Tsutomu 128
Masters, Nathan 170
Matsuoka, Satoshi 6
Mayo, Rafael 51
McElroy, Ciarán 185
McGettrick, Séamas 185
Meadows, Larry 138
Merelli, Ivan 29, 92
Milanesi, Luciano 92
Miller, Barton P. 155
Miller, Robert 149
Moore, Shirley 162
Muñoz-Sudupe, Antonio 47
Musacchio, Stefano 174

N

Nagel, Wolfgang E. 85, 86, 157
Nageswaran, Jayram Moorkanikara 182
Naumann, Uwe 77
Navarro, Denis 47
Neuenhahn, Martin 183
Nicolai, Mike 93
Nicolau, Alex 182
Nieplocha, Jarek 98
Noll, Tobias G. 183
Nowak, Fabian 181
Numrich, Robert W. 26

O

Oh, Jung S. 131
Olcoz, Katzalin 116
Olkhovskaya, Olga 110

P

Pütz, Matthias 145
Pérez, Rosa 122
Pérez-Gaviro, Sergio 47
Paar, Christof 179
Palazzari, Paolo 184
Palmer, Bruce 98
Pelzl, Jan 179
Pena, Tomás F. 25
Petrini, Fabrizio 98
Pfeiffer, Gerd 179
Pflüger, Stefan 85
Piera, Javier Roca 31
Platzner, Marco 180
Plaza, Antonio 122
Plaza, Javier 122
Poli, Emanuele 173
Polzella, Francesco 100
Popova, Elena 91
Prasanna, Viktor K. 36, 48
Prieto, Manuel 64
Pringle, Gavin J. 171
Probst, Markus 93
Pulatova, Farzona 158

Q

Quintana-Ortí, Enrique S. 51, 65
Quintana-Ortí, Gregorio 51

R

Ramalho-Natario, Maria 3
Rasch, Arno 76
Rath, Volker 127
Resch, Michael 117
Reshetnyak, Maxim 112
Rexachs, Dolores 97
Richter, Marcus 20
Rodero, Ivan 115
Rossi, Mauro 47
Ruiz-Lorenzo, Juan Jesus 47
Rünger, Gudula 81, 105

S

Sérot, Jocelyn 59
Salama, Rafik A. 106
Sameh, Ahmed 106
Santos, Guna 97
Sawai, Ryo 128
Schöne, Robert 85
Schüle, Josef 63
Schifano, Sebastiano Fabio 47
Schimmler, Manfred 179
Schirski, Marc 123
Schleiffer, Christian 179
Schug, Alexander 131
Schulz, Martin 155
Schumacher, Jörg 145
Schumacher, Tobias 180
Schwarz, Christoph 109
Sciretti, Daniele 47
Seidel, Jan 69
Sevilla, Diego 99
Shah, N. Jon 30
Siso-Nadal, Fernando 149
Sloan, Terry M. 171

Sosonkina, Masha 10
Spinatelli, Gianmarco 100
Stöcker, Tony 30
Stüben, Hinnerk 132
Steffen, Bernhard 112
Stratford, Kevin 146
Streuer, Thomas 132
Strohhäcker, Sebastian 23
Stødle, Daniel 121
Suess, Michael 41, 42
Sutmann, Godehard 18
Sørevik, Tor 70

T

Tafti, Danesh 140
Tarancón, Alfonso 47
Terboven, Christian 141, 161
Tipparaju, Vinod 98
Tirado, Francisco 64, 116
Tiskin, Alexander 37
Tisma, Reinhard 172
Todorov, Ilian T. 148
Torquati, Massimo 100
Toschi, Federico 174
Trautmann, Sven 82
Trenkler, Bernd 86
Trew, Arthur S. 171
Trieu, Binh 20
Tripiccione, Raffaele 47
Tufo, Henry M. 150

V

Vahedipour, Kaveh 30
van der Pas, Ruud 139
Vanneschi, Marco 100
Vanni, Andrea 168
Varnik, Ebadollah 77
Veidenbaum, Alex 182
Velasco, Jose Luis 47
Velasco, Jose M. 116
Verma, Abhinav 131
Vidal-Maciá, Antonio. M. 52
Viré, Axelle 111
Viti, Federica 92
von Livonius, Jörg 183

W

Weidendorfer, Josef 163
Wenzel, Wolfgang 131
Wolf, Andreas 127
Wolf, Felix 158
Wolter, Marc 123
Wylie, Brian 158

X

Xia, Yinglong 36
Xu, Wei 75

Y

Yamaguchi, Yoshiki 128
Yasunaga, Moritoshi 128

Z

Zhuo, Ling 48
Zuccato, Pierfrancesco 100

Already published:

Modern Methods and Algorithms of Quantum Chemistry - Proceedings
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 1
ISBN 3-00-005618-1, February 2000, 562 pages
out of print

Modern Methods and Algorithms of Quantum Chemistry - Poster Presentations
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 2
ISBN 3-00-005746-3, February 2000, 77 pages
out of print

Modern Methods and Algorithms of Quantum Chemistry - Proceedings, Second Edition
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 3
ISBN 3-00-005834-6, December 2000, 638 pages
out of print

Nichtlineare Analyse raum-zeitlicher Aspekte der hirnelektrischen Aktivität von Epilepsiepatienten
Jochen Arnold
NIC Series Volume 4
ISBN 3-00-006221-1, September 2000, 120 pages

Elektron-Elektron-Wechselwirkung in Halbleitern: Von hochkorrelierten kohärenten Anfangszuständen zu inkohärentem Transport
Reinhold Lövenich
NIC Series Volume 5
ISBN 3-00-006329-3, August 2000, 146 pages

Erkennung von Nichtlinearitäten und wechselseitigen Abhängigkeiten in Zeitreihen
Andreas Schmitz
NIC Series Volume 6
ISBN 3-00-007871-1, May 2001, 142 pages

Multiparadigm Programming with Object-Oriented Languages - Proceedings
Kei Davis, Yannis Smaragdakis, Jörg Striegnitz (Editors)
Workshop MPOOL, 18 May 2001, Budapest
NIC Series Volume 7
ISBN 3-00-007968-8, June 2001, 160 pages

Europhysics Conference on Computational Physics - Book of Abstracts
Friedel Hossfeld, Kurt Binder (Editors)
Conference, 5 - 8 September 2001, Aachen
NIC Series Volume 8
ISBN 3-00-008236-0, September 2001, 500 pages

NIC Symposium 2001 - Proceedings
Horst Rollnik, Dietrich Wolf (Editors)
Symposium, 5 - 6 December 2001, Forschungszentrum Jülich
NIC Series Volume 9
ISBN 3-00-009055-X, May 2002, 514 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms - Lecture Notes
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 10
ISBN 3-00-009057-6, February 2002, 548 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms- Poster Presentations
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 11
ISBN 3-00-009058-4, February 2002, 194 pages

Strongly Disordered Quantum Spin Systems in Low Dimensions: Numerical Study of Spin Chains, Spin Ladders and Two-Dimensional Systems
Yu-cheng Lin
NIC Series Volume 12
ISBN 3-00-009056-8, May 2002, 146 pages

Multiparadigm Programming with Object-Oriented Languages - Proceedings
Jörg Striegnitz, Kei Davis, Yannis Smaragdakis (Editors)
Workshop MPOOL 2002, 11 June 2002, Malaga
NIC Series Volume 13
ISBN 3-00-009099-1, June 2002, 132 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms - Audio-Visual Lecture Notes
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 14
ISBN 3-00-010000-8, November 2002, DVD

Numerical Methods for Limit and Shakedown Analysis
Manfred Staat, Michael Heitzer (Eds.)
NIC Series Volume 15
ISBN 3-00-010001-6, February 2003, 306 pages

Design and Evaluation of a Bandwidth Broker that Provides Network Quality of Service for Grid Applications
Volker Sander
NIC Series Volume 16
ISBN 3-00-010002-4, February 2003, 208 pages

Automatic Performance Analysis on Parallel Computers with SMP Nodes
Felix Wolf
NIC Series Volume 17
ISBN 3-00-010003-2, February 2003, 168 pages

Haptisches Rendern zum Einpassen von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen
Stefan Birmanns
NIC Series Volume 18
ISBN 3-00-010004-0, September 2003, 178 pages

Auswirkungen der Virtualisierung auf den IT-Betrieb
Wolfgang Gürich (Editor)
GI Conference, 4 - 5 November 2003, Forschungszentrum Jülich
NIC Series Volume 19
ISBN 3-00-009100-9, October 2003, 126 pages

NIC Symposium 2004
Dietrich Wolf, Gernot Münster, Manfred Kremer (Editors)
Symposium, 17 - 18 February 2004, Forschungszentrum Jülich
NIC Series Volume 20
ISBN 3-00-012372-5, February 2004, 482 pages

Measuring Synchronization in Model Systems and Electroencephalographic Time Series from Epilepsy Patients
Thomas Kreutz
NIC Series Volume 21
ISBN 3-00-012373-3, February 2004, 138 pages

Computational Soft Matter: From Synthetic Polymers to Proteins - Poster Abstracts
Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Editors)
Winter School, 29 February - 6 March 2004, Gustav-Stresemann-Institut Bonn
NIC Series Volume 22
ISBN 3-00-012374-1, February 2004, 120 pages

Computational Soft Matter: From Synthetic Polymers to Proteins - Lecture Notes
Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Editors)
Winter School, 29 February - 6 March 2004, Gustav-Stresemann-Institut Bonn
NIC Series Volume 23
ISBN 3-00-012641-4, February 2004, 440 pages

Synchronization and Interdependence Measures and their Applications to the Electroencephalogram of Epilepsy Patients and Clustering of Data
Alexander Kraskov
NIC Series Volume 24
ISBN 3-00-013619-3, May 2004, 106 pages

High Performance Computing in Chemistry
Johannes Grotendorst (Editor)
Report of the Joint Research Project:
High Performance Computing in Chemistry - HPC-Chem
NIC Series Volume 25
ISBN 3-00-013618-5, December 2004, 160 pages

Zerlegung von Signalen in unabhängige Komponenten: Ein informationstheoretischer Zugang
Harald Stögbauer
NIC Series Volume 26
ISBN 3-00-013620-7, April 2005, 110 pages

Multiparadigm Programming 2003
Joint Proceedings of the
3rd International Workshop on Multiparadigm Programming with Object-Oriented Languages (MPOOL'03)
and the
1st International Workshop on Declarative Programming in the Context of Object-Oriented Languages (PD-COOL'03)
Jörg Striegnitz, Kei Davis (Editors)
NIC Series Volume 27
ISBN 3-00-016005-1, July 2005, 300 pages

Integration von Programmiersprachen durch strukturelle Typanalyse und partielle Auswertung
Jörg Striegnitz
NIC Series Volume 28
ISBN 3-00-016006-X, May 2005, 306 pages

OpenMolGRID - Open Computing Grid for Molecular Science and Engineering
Final Report
Mathilde Romberg (Editor)
NIC Series Volume 29
ISBN 3-00-016007-8, July 2005, 86 pages

GALA Grünenthal Applied Life Science Analysis
Achim Kless and Johannes Grotendorst (Editors)
NIC Series Volume 30
ISBN 3-00-017349-8, November 2006, 204 pages

Computational Nanoscience: Do It Yourself!
Lecture Notes
Johannes Grotendorst, Stefan Blügel, Dominik Marx (Editors)
Winter School, 14. - 22 February 2006, Forschungszentrum Jülich
NIC Series Volume 31
ISBN 3-00-017350-1, February 2006, 528 pages

NIC Symposium 2006 - Proceedings
G. Münster, D. Wolf, M. Kremer (Editors)
Symposium, 1 - 2 March 2006, Forschungszentrum Jülich
NIC Series Volume 32
ISBN 3-00-017351-X, February 2006, 384 pages

Parallel Computing: Current & Future Issues of High-End Computing
Proceedings of the International Conference ParCo 2005
G.R. Joubert, W.E. Nagel, F.J. Peters,
O. Plata, P. Tirado, E. Zapata (Editors)
NIC Series Volume 33
ISBN 3-00-017352-8, October 2006, 930 pages

From Computational Biophysics to Systems Biology 2006 Proceedings
U.H.E. Hansmann, J. Meinke, S. Mohanty, O. Zimmermann (Editors)
NIC Series Volume 34
ISBN-10 3-9810843-0-6, ISBN-13 978-3-9810843-0-6,
September 2006, 224 pages

Dreistufig parallele Software zur Parameteroptimierung von Support-Vektor-Maschinen mit kostensensitiven Gütemaßen
Tatjana Eitrich
NIC Series Volume 35
ISBN 978-3-9810843-1-3, March 2007, 262 pages

All volumes are available online at

http:// www.fz-juelich.de/nic-series/.