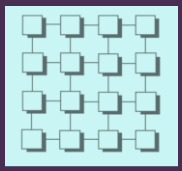
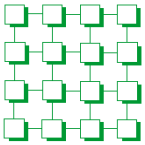


*ParCo 2011*

# PARALLEL COMPUTING 2011

## BOOK OF ABSTRACTS

**Ghent University**
**Het Pand**
**30 August – 2 September 2011**



*ParCo 2011*

# PARALLEL COMPUTING 2011

BOOK OF ABSTRACTS

**Ghent University**
**Het Pand**
**30 August – 2 September 2011**

ISBN 978 90 382 1835 9

# CONTENTS

## CONFERENCE COMMITTEE

- **Gerhard Joubert, conference chair**
- **Frans Peters, finance chair**
- **Koen De Bosschere, organising chair**
- **Erik D'Hollander, program chair**
- **David Padua, program co-chair**
- **Mark Sawyer, industrial chair**

## PROGRAM COMMITTEE

George Almási
Rosa Badia
Henri Bal
Gianfranco Bilardi
Christian Bischof
Arndt Bode
Jan Broeckhove
Mark Bull
Andrea Clematis
Luisa D'Amore
Michel Dayde
Bjorn De Sutter
Frank Dehne
Frédéric Desprez
Anne Elster
Dick Epema
Thomas Fahringer
Paul Feautrier
Basilio Fraguela
Franz Franchetti
Efstratios Gallopoulos
William Gropp
Lei Huang
Chris Jesshope
Hai Jin
David R. Kaeli
Paul Kelly
Christoph Kessler
Bettina Krammer
Dieter Kranzlmüller
Herbert Kuchen
Alexey Lastovetsky
Jin-Fu Li
Thomas Ludwig
Bernd Mohr
Wolfgang Nagel
Viktor Pankratius
Nicolai Petkov
Oscar Plata
Sabri Pllana
Thierry Priol
Christian Pérez
Markus Püschel
Dirk Roose
Gudula Rünger
Martin Schulz
Tor Sørevik
Domenico Talia
Guangming Tan
Paco Tirado
Denis Trystram
Marco Vanneschi
Wim Vanroose
Robert van de Geijn
Christoph von Praun

## SESSION OVERVIEW

## DATE: TUESDAY, 30/AUG/2011

| | | | |
|---|---|---|---|
| **8:30 - 9:30** | **R: Registration** | | |
| **9:30 - 9:45** | **O: Opening**<br>**Gerhard Joubert,** Conference chair<br>**Luc Moens,** Vice-Principal, University of Ghent | | |
| **9:45 - 10:45** | **K1: Keynotes**<br>Chair: **Gerhard Joubert**<br>**Bernhard Fabianek,** The Future of High Performance Computing in Europe<br>**Thomas Lippert,** Europe's Supercomputing Research Infrastructure PRACE | | |
| **10:45 - 11:00** | **A: Announcements** | | |
| **11:30 - 12:30** | **A1: Algorithms I**<br>Chair: **Thomas Lippert** | **B1: Languages I**<br>Chair: **Koen De Bosschere** | **C1: High performance applications I**<br>Chair: **Christoph F. Eick** |
| **12:30 - 13:45** | **L1: Lunch** | | |
| **14:00 - 15:00** | **A2: Numerical algorithms I**<br>Chair: **Frans Peters** | **B2: Load balancing I**<br>Chair: **David Padua** | **C2: GPU applications I**<br>Chair: **Tsutomu Maruyama** |
| **15:30 - 17:00** | **A3: High performance applications II**<br>Chair: **Koen De Bosschere** | **B3: Thread management**<br>Chair: **Welf Löwe** | **C3: Performance modeling and analysis I**<br>Chair: **Erik D'Hollander** |

| | |
|---|---|
| **17:30 - 19:30** | **S1: Reception**<br>Reception at the City Hall of Ghent |

## DATE: WEDNESDAY, 31/AUG/2011

| | | | |
|---|---|---|---|
| **9:00 - 10:00** | **K2: Keynote**<br>Chair: **Koen De Bosschere**<br>**Ignacio Martín Llorente,** Challenges in Hybrid and Federated Cloud Computing | | |
| **10:00 - 11:00** | **A4: Algorithms II**<br>Chair: **Frans Peters** | **B4: Multicores I**<br>Chair: **Arun Chauhan** | **C4: ParaFPGA Mini-symposium I**<br>Chair: **Dirk Stroobandt** |
| **11:30 - 12:30** | **A5: Massive parallelism I**<br>Chair: **Jack B. Dennis** | **B5: GPU applications II**<br>Chair: **Arun Chauhan** | **C5: ParaFPGA Mini-symposium II**<br>Chair: **Erik D'Hollander** |
| **12:30 - 13:45** | **L2: Lunch** | | |
| **14:00 - 15:00** | **A6: Numerical algorithms II**<br>Chair: **David Padua** | **B6: Heterogeneous computing**<br>Chair: **Joerg Keller** | **C6: ParaFPGA Mini-symposium III**<br>Chair: **Tsutomu Maruyama** |
| **15:30 - 18:00** | **I: Industrial session**<br>Chair: **Christopher Mark Sawyer** | | |

## DATE: THURSDAY, 01/SEP/2011

| | | | |
|---|---|---|---|
| **9:00 - 10:00** | **K3: Keynote**<br>Chair: **David Padua**<br>**William D. Gropp,** Performance Modeling as the Key to Extreme Scale Performance | | |
| **10:00 - 11:00** | **A7: Cloud computing I**<br>Chair: **Amey Karkare** | **B7: Languages II**<br>Chair: **Jack B. Dennis** | **C7: Exascale Mini-symposium I**<br>Chair: **Enrique S. Quintana-Orti** |
| **11:30 - 12:30** | **A8: Numerical algorithms III**<br>Chair: **Jack B. Dennis** | **B8: Multicores II**<br>Chair: **Sabri Pllana** | **C8: Exascale Mini-symposium II**<br>Chair: **Jesus Labarta** |
| **12:30 - 13:45** | **L3: Lunch**<br>Coffee, tea, refreshments and sandwiches | | |
| **14:00 - 15:00** | **K4: Keynote**<br>Chair: **Erik D'Hollander**<br>**Jack B. Dennis,** The Fresh Breeze Project | | |
| **15:30 - 16:30** | **A9: Automatic parallelization**<br>Chair: **David Padua** | **B9: Performance modeling and analysis II**<br>Chair: **Sabri Pllana** | **C9: Exascale Mini-symposium III**<br>Chair: **Jesus Labarta** |
| **17:30 - 22:00** | **S2: Conference dinner**<br>Walk in historical Ghent followed by the banquet. | | |

## DATE: FRIDAY, 02/SEP/2011

| | | | |
|---|---|---|---|
| **9:00 - 10:00** | **K5: Keynote**<br>Chair: **Frans Peters**<br>**Andy Adamatsky**, Physarum Machines | | |
| **10:00 - 11:00** | **A10: Massive parallelism II**<br>Chair: **Marco Danelutto** | **B10: Parallel I/O**<br>Chair: **Andy Adamatzky** | **C10: GPU applications III**<br>Chair: **Przemyslaw Stpiczynski** |
| **11:30 - 13:00** | **A11: Cloud computing II**<br>Chair: **Andy Adamatzky** | **B11: Skeleton programming**<br>Chair: **Marco Danelutto** | **C11: GPU applications IV**<br>Chair: **Erik D'Hollander** |
| **13:00 - 13:15** | **C: Closing session** | | |
| **13:15 - 14:00** | **L4: Lunch**<br>Coffee, tea, refreshments and sandwiches | | |

# KEYNOTES

# THE FUTURE OF HIGH PERFORMANCE COMPUTING IN EUROPE

*Bernhard Fabianek*

European Commission, Belgium

**Abstract**

This talk will highlight the pan-European relevance high-performance computing has gained. It will outline the setup of an HPC eco-system and the challenges associated with this. Furthermore a strategic agenda for high-performance computing will be presented including the underlying fundamental data and the envisaged actions.

# EUROPE'S SUPERCOMPUTING RESEARCH INFRASTRUCTURE PRACE

*Thomas Lippert*

Forschungszentrum Juelich GmbH, Germany

**Abstract**

Within the last three years a consortium of 20 European countries has prepared the legal and technical prerequisites for the establishment of a leadership-class supercomputing infrastructure in Europe. The consortium named "Partnership for Advanced Computing in Europe" has carried out a preparatory phase project supported by the European Commission. The research infrastructure was inaugurated in June 2010. Four members have committed to provide compute cycles worth EURO 100 Million each in the 5 years period until 2015. Six sites from the four hosting countries in succession will install machines of the highest performance class (Tier-0). Access to the infrastructure is provided on the basis of scientific quality through a pan-European peer review scheme under the guidance of the scientific steering committee (SSC) of PRACE. Proposals can be submitted in form of projects or as programs by communities. The provision of computer time through PRACE started in August 2010 on the supercomputer JUGENE of Research Centre Juelich. At this time PRACE is further developping its infrastructure in the first implementation project soon to be followed by the second implementation project, both again funded by the European Commission. As important steps forward, PRACE's Tier-0 supercomputing infrastructure will be complemented by national centres (Tier-1) of the PRACE partners. Furthermore, PRACE aims at establishing an industrial user-vendor platform with the goal to contribute to a European Technology Platform for HPC.

# CHALLENGES IN HYBRID AND FEDERATED CLOUD COMPUTING

*Ignacio Martín Llorente*

Universidad Complutense de Madrid, Spain

**Abstract**

Federated and hybrid clouds will play a significant role in IT strategies and e-infrastructures in the coming years. The keynote describes the different hybrid cloud computing scenarios, ranging from the combination of local private infrastructure with commercial cloud providers that offer no real support for federation to one built on data centers of the same organization where the sites are completely dedicated to supporting all aspects of federation. The level of federation is defined based on the amount of information disclosed and how much control over the resources is provided across sites. The talk presents the existing challenges for interoperability in federated and hybrid cloud computing scenarios, and ends with examples of multi-cloud environments running OpenNebula, such as the hybrid cloud computing approach in the StratusLab project.

# PERFORMANCE MODELING AS THE KEY TO EXTREME SCALE PERFORMANCE

*William Gropp*

University of Illinois at Urbana-Champaign, United States of America

**Abstract**

Parallel computing is primarily about achieving greater performance than is possible without using parallelism. Especially for the high-end, where systems cost tens to hundreds of millions of dollars, making the best use of these valuable and scarce systems is important. Yet few applications really understand how well they are performing with respect to the achievable performance on the system. The Blue Waters system, currently being installed at the University of Illinois, will offer sustained performance in excess of 1 PetaFLOPS for many applications. However, achieving this level of performance requires careful attention to many details, as this system has many features that must be used to get the best performance. To address this problem, the Blue Waters project is exploring the use of performance models that provide enough information to guide the development and tuning of applications, ranging from improving the performance of small loops to identifying the need for new algorithms. Using Blue Waters as an example of an extreme scale system, this talk will describe some of the challenges faced by applications at this scale, the role that performance modeling can play in preparing applications for extreme scale, and some ways in which performance modeling has guided performance enhancements for those applications.

# THE FRESH BREEZE PROJECT

*Jack B. Dennis*

MIT Computer Science and Atrificial Intelligence Laboratory, United States of America

**Abstract**

The development of multicore chips is producing a sea change in the world of computer system architecture. One result is the conventional program execution model (PXM), that has been the mainstay of commercial software development for more than thirty years, has been rendered obsolete. The Fresh Breeze PXM is proposed as a new basis for the design of massively parallel computer systems that can achieve high performance with sound support for program development including composability of parallel programs. The presentation will review the history of Program Execution Models (PXMs) and how their evolution has led to the Fresh Breeze project.

## PHYSARUM MACHINES

*Andy Adamatzky*

UWE, Bristol, UK, United Kingdom

**Abstract**

A Physarum machine is a programmable amorphous biological computer experimentally implemented in a plasmodium of Physarum polycephalum. Physarum machines are programmed by configurations of repelling and attracting gradients, and localized reflectors. Physarum is a a biological prototype of all storage modification machines and modern computer architectures. In this talk it will be shown how a plasmodium of Physarum polycephlum can solve geometrical and graph-theoretic problems, implement logical computations and intelligence actuation.

# A1: ALGORITHMS I

# PARALLEL REMESHING IN TREE CODES FOR VORTEX PARTICLE METHODS

*Robert Speck, Rolf Krause, Paul Gibbon*

University of Lugano, Switzerland

**Abstract**

Parallel vortex methods are an efficient and widely used technique for massively parallel simulations of turbulent fluid flows. Based on the vorticity-velocity formulation of the Navier-Stokes equations, the vorticity field is discretized with regularized particles. One of their big advantages is that vortex particles posses an intrinsic adaptivity, since computational elements exist only where the corresponding fields are non-zero. However, this method leads to an N-body problem, since the interaction between the regularized fluid particles is dominated by long-range forces. Moreover, the convergence condition of regularized vortex particle methods requires a permanent particle overlap. In this paper, we demonstrate that both problems can be solved using a mesh-free parallel Barnes-Hut tree code. Our code reduces the number of interactions from O(N^2) to O(N log N) and it automatically guarantees particle overlap by incorporating the remeshing concept.

# A CASE STUDY OF THE TASK-BASED PARALLEL WAVEFRONT PATTERN

*Antonio J. Dios, Angeles Navarro, Rafael Asenjo, Francisco Corbera, Emilio L. Zapata*

University of Malaga, Spain

**Abstract**

This paper analyzes the applicability of the task programming model in the parallelization of the wavefront pattern. Computations on this type of problem are characterized by a data dependency pattern across a data space, which can produce a variable number of independent tasks through the traversal of such a space. We explore several implementations of this pattern, based on the current state-of-the-art threading frameworks that support tasks. For each implementation, we discuss the specific issues from a programmer´s point of view, highlighting the advantageous features in each case. We conduct several experiments to identify the factors that can limit the performance in each implementation. Through our study, we have found that TBB provides some distinguishing features that allow the more efficient implementations, specially for the fine grain case, which is frequently found in real wavefront problems and that is more challenging. Features such as the atomic capture, and the task recycling mechanism, coupled with prioritization of task to exploit data locality, have demonstrated to lead to important performance improvements in our experiments, specially when the granularity of the task is fine. These optimizations could be wrapped in a higher level template to facilitate the coding of wavefront codes to less experienced users.

# B1: LANGUAGES I

# CORENTO - SIMD PARALLELISM FROM PORTABLE HIGH-LEVEL CODE

*Juhana Helovuo, Jarkko Niittylahti, Heikki Berg*

Atostek Ltd, Finland

**Abstract**

This article introduces the Corento programming language, which is an implementation tool for signal processing kernels on parallel processors. First, we discuss the current industry practice of programming such processors using the C language, and find that the situation is unsatisfactory from both technical and business viewpoints. Corento is an alternative tool to improve the situation. It improves the performance and portability of signal processing kernels, while keeping the programming task reasonably simple. As an example, we show how Corento can be used to implement the Fast Fourier Transform, and give an overview of the Corento compilation process. Finally, we show experimental performance results of our Corento implementation on multiple SIMD parallel processors from single source code.

# A PARALLEL BENCHMARK SUITE FOR FORTRAN COARRAYS

*David Henty*

The University of Edinburgh, United Kingdom

**Abstract**

Coarrays are a feature of the new Fortran 2008 standard that enable parallelism using a small number of additional language elements. A new array declaration syntax allows for remotely accessible variables, with data allocated across multiple images. The execution model is that of a Partitioned Global Address Space (PGAS) language. Since Fortran coarrays are in their infancy, and full compiler support has only recently emerged, it is important to understand the performance characteristics of any parallel operations. Although benchmark suites exist for well-established models such as MPI, OpenMP and UPC, none are currently available for coarrays. The results of such benchmarks are important as they guide both the applications programmer and the compiler or library developer. In this paper we describe a low- level benchmark suite for Fortran coarrays that measures the performance of a se- lection of basic parallel operations. We present initial performance results on Cray architectures and a general-purpose Intel cluster. We hope this suite will help in the development and uptake of these new parallel features of the Fortran language.

# C1: HIGH PERFORMANCE APPLICATIONS I

# TRAJECTORY-SEARCH ON SCALEMP'S VSMP ARCHITECTURE

*Nicolas Berr, Dirk Schmidl, Jens Henrik Göbbert, Stefan Lankes, Dieter an Mey, Thomas Bemmerl, Christian Bischof*

RWTH Aachen University, Germany

**Abstract**

ScaleMP's vSMP software turns commodity Infiniband clusters with Intel's x86 processors into large shared memory machines providing a single system image at low cost. However, codes need to be tuned to deliver good performance on these machines. TrajSearch, developed at the Institute for Combustion Technology at RWTH Aachen University, is a post-processing code for analyzing trajectories in three dimensional turbulent flow fields. The initial OpenMP version of TrajSearch has been tuned to run very efficiently on the vSMP cluster which exhibits a pronounced NUMA behavior. As a side effect, the code also performs nicely on a more expensive large SGI Altix UV shared memory machine.

# TOWARDS AN APPLICATION OF HIGH-PERFORMANCE COMPUTER SYSTEMS TO 3D SIMULATIONS OF HIGH ENERGY DENSITY PLASMAS IN Z-PINCHES

*Vladimir Gasilov, Alexey Boldarev, Sergey Dyachenko, Olga Olkhovskaya, Elena Kartasheva, Gennadiy Bagdasarov, Sergey Boldyrev, Irina Gasilova, Valeriy Shmyrov, Svetlana Tkachenko, Julien Grunenwald, Thierry Maillard*

Keldysh Institute of Applied Mathematics, Russian Federation

**Abstract**

The paper presents the new versatile code designed by our team for applications of supercomputer systems to large-scale 3D simulations of problems appearing in experimental Pulsed Power Energetic (PPE) and High Energy Density Plasmas (HEDP). Because of the diversity of physical processes which should be taken into account and because of high degrees of spatial and temporal scale non-uniformities, these problems on the whole are so intricate and computationally expensive, that may be comprehensively studied by use of distributed high-performance (multi-Tflops) computing only, especially in 3D case. The development of parallel codes for HEDP-"multiphysics" simulations which would be able to facilitate a design of future experiments and to predict final parameters of the pinch plasmas is a real challenge for specialists in applied mathematics. The developed code (referred to as MARPLE - Magnetically Accelerated Radiative Plasma Explorer, 3D-version) is based on magnetohydrodynamic model in accordance with modern knowledge about multicharged pinch plasmas. Essential elements of this model are wide-range semi-empirical equations of state, electron-ion energy relaxation, energy dissipation, and radiative transfer. This is an object-oriented, parallel code designed for scientific simulations at systems performing distributed computations.

# A2: NUMERICAL ALGORITHMS I

# SOLVING THE GENERALIZED SYMMETRIC EIGENVALUE PROBLEM USING TILE ALGORITHMS ON MULTICORE ARCHITECTURES

*Hatem Ltaief, Piotr Luszczek, Azzam Haidar, Jack Dongarra*

KAUST, Saudi Arabia

**Abstract**

This paper proposes an efficient implementation of the generalized symmetric eigenvalue problem on multicore architecture. Based on a four-stage approach, the original problem is first transformed into a standard symmetric eigenvalue problem by computing the Cholesky factorization of the right hand side symmetric definite positive matrix (first stage), and applying the inverse of the freshly computed triangular Cholesky factors to the original dense symmetric matrix of the problem (second stage). The computation proceeds by reducing the updated dense symmetric matrix to symmetric band form (third stage). The band structure is further reduced by applying a bulge chasing procedure, which annihilates the extra off-diagonal entries (fourth stage). The eigenvalues are calculated from the tridiagonal form using the LAPACK QR algorithm (i.e., DTSEQR routine). The various tasks can concurrently run in an out-of-order fashion using the dynamic scheduler QUARK, which ensures the dependencies are not violated. The obtained tile four-stage generalized symmetric eigenvalue solver significantly outperforms the state-of-the-art numerical libraries (up to 21-fold speed up against LAPACK with optimized multithreaded MKL BLAS and up to 4-fold speed up against the corresponding routine from the commercial numerical software Intel MKL) on four sockets twelve cores AMD system with a 24000x24000 matrix size.

# IMPROVING PERFORMANCE OF TRIANGULAR MATRIX-VECTOR BLAS ROUTINES ON GPUS

*Przemyslaw Stpiczynski, Marek Karwacki*

Maria Curie-Sklodowska University, Poland

**Abstract**

CUBLAS is a widely used implementation of BLAS (Basic Linear Algebra Subprograms) for NVIDIA CUDA Graphical Processing Units (GPUs). The aim of this paper is to show that the performance of the selected Level 2 BLAS routines for working with triangular matrices can be improved using some optimization techniques suitable for GPUs like using shared memory, coalesced memory access and load balancing. We present new implementations of the subroutines TRMV and {TRSV. The results of experiments carried out on two GPU architectures: Tesla C2050 and GeForce GTX 260 show that these new implementations are up to 500% faster than corresponding routines from CUBLAS Library.

# B2: LOAD BALANCING I

# BALANCING CPU LOAD FOR IRREGULAR MPI APPLICATIONS

*Joerg Keller, Mudassar Majeed, Christoph W. Kessler*

FernUniversitaet in Hagen, Germany

**Abstract**

MPI applications typically are designed to be run on a parallel machine with one process per core. If processes exhibit different computational load, either the code must be rewritten for load balancing, with negative side-effects on readability and maintainability, or the one-process-per-core philosophy leads to a low utilization of many processor cores. If several processes are mapped per core to increase CPU utilization, the load might still be unevenly distributed among the cores if the mapping is unaware of the process characteristics. Therefore, similarly to the MPI_Graph_create() function where the program gives hints on communication patterns so that MPI processes can be placed favorably, we propose a MPI_Load_create() function where the program supplies information on the relative loads of the MPI processes, such that processes can be favorably grouped and mapped onto processor cores. In order to account for scalability and restricted knowledge of individual MPI processes, we also propose an extension MPI_Dist_load_create() similar to MPI_Dist_graph_create(), where each individual MPI process only knows the loads of a subset of the MPI processes. We detail how to implement both variants on top of MPI, and provide experimental performance results both for synthetic and numeric example applications. The results indicate that load balancing is favorable in both cases.

# REACTIVE REBALANCING FOR SCIENTIFIC SIMULATIONS RUNNING ON EXASCALE HIGH PERFORMANCE COMPUTERS

*Roel Wuyts, Karl Meerbergen, Pascal Costanza*

imec, Belgium

**Abstract**

Exascale computers, the next generation of high performance computers, are expected to process 1 exaflops around 2018. However the processor cores used in these systems are very likely to suffer from unpredictable high variability in performance. We built a prototype general-purpose reactive work rebalancer that handles such performance variability with low overhead. We did an experimental validation by developing a reactive rebalancer library in UPC, and using it in a 5-point stencil (heat) simulation. The experiments show that our approach has very limited overhead that compensates for runtime processor speed variations, with or without simulated processor slowdowns.

# C2: GPU APPLICATIONS I

# MONTE CARLO OPTION PRICING WITH GRAPHICS PROCESSING UNITS

*Fredrik Nordh, Erwin Laure*

Royal Institute of Technology, Sweden

**Abstract**

Monte Carlo is a common method in financial engineering for a wide variety of problems, one being option pricing. Growing volumes and complexity of work that needs to be performed as well as strict requirements for fast responses makes for a pressing demand for high performance computing. In this paper we study the applicability of GPUs for pricing exotic options (basket, Asian) using Monte Carlo techniques within the environment of a large financial institution. Contrary to many experimental results achieved by analyzing the computational kernel we focus on the performance of the end-to-end system that is now in production use at Handelsbanken, one of the major Swedish financial institutes. We show that graphics cards can outperform CPUs given certain conditions and for reasonable problem sizes we find a 12x improvement over sequential code when pricing options in a production system.

# SPEEDING-UP THE DISCRETE WAVELET TRANSFORM COMPUTATION WITH MULTICORE AND GPU-BASED ALGORITHMS

*Vicente Galiano, Otoniel Lopez, Manuel P. Malumbres, Hector F. Migallon*

Universidad Miguel Hernandez, Spain

**Abstract**

In this work we propose several parallel algorithms to compute the two dimensional discrete wavelet transform (2D-DWT), exploiting the available hardware resources. In particular, we will explore OpenMP optimized versions of 2DDWT over a multicore platform and also we will develop CUDA-based 2D-DWT algorithms which are able to run on GPUs (Graphics Processing Unit). The proposed algorithms are based on several 2D-DWT computation approaches as (1) filter-bank convolution, (2) lifting transform and (3) matrix convolution, so we can determine which of them better adapts to our parallel versions.

# A3: HIGH PERFORMANCE APPLICATIONS II

# ON-THE-FLY SINGULAR VALUE DECOMPOSITION FOR AITKEN'S ACCELERATION OF THE SCHWARZ DOMAIN DECOMPOSITION METHOD

*Laurent Berenguer, Thomas Dufaud, Toan Pham, Damien Tromeur-Dervout*

Université Lyon 1, France

**Abstract**

We present a parallel implementation of the alternating Schwarz method improved by the Aitken's acceleration of the convergence. The overlapping alternating Schwarz procedure leads to solve only subdomains problems and to perform local communications between the neighboring subdomains to exchange boundary conditions at interfaces. The singular value decomposition of the solution on interfaces at successive iterations allows us to create a low-rank approximation of the error operator of the Schwarz method. The main idea is to consider successive solutions on interfaces as a set of snapshots. The truncated SVD of these snapshots gives a basis of an optimal low-dimensional space for representing these snapshots. The Aitken's acceleration is performed in this low-dimensional space. The main difficulty is to estimate how many iterations, so how many snapshots, are needed to perform efficient accelerations. We propose a new Aitken's acceleration algorithm, using on-the-fly SVD to compute the SVD simultaneously with the Schwarz process in order to avoid overestimating the number of needed iterations. Numerical results for groundwater flow problems, including scalability test, are discussed.

# A SOFTWARE CONCEPT FOR CACHE-EFFICIENT SIMULATION ON DYNAMICALLY ADAPTIVE STRUCTURED TRIANGULAR GRIDS

*Oliver Meister, Michael Bader*

Universität Stuttgart, Germany

**Abstract**

We present a software concept for element-oriented numerical solvers for PDEs on dynamically adaptive grids. The respective layer concept provides volume kernels to implement layer operators, which work on a traversal layer that implements cache-efficient traversals on structured adaptive triangular grids (provided by a grid layer underneath). We present a simple heat propagation problem and a coupled solver for flows in porous media as examples, and study the achieved single-core and memory performance. First results are also given for current work on parallelization of the approach.

# PERFORMANCE ANALYSIS OF AN ULTRASOUND RECONSTRUCTION ALGORITHM FOR NON-DESTRUCTIVE TESTING

*Antoine Pedron, Lionel Lacassagne, Victor Barbillon, Franck Bimbard, Gilles Rougeron, Stéphane Le Berre*

CEA, France

**Abstract**

The CIVA software platform developed by CEA-LIST offers various simulation and data processing modules dedicated to non-destructive testing (NDT). In particular, ultrasonic imaging and reconstruction tools are proposed in the purpose of localizing echoes and identifying and sizing the detected defects. Because of the complexity of data processed, computation time is now a limitation for the optimal use of available information. In this article, we present performance results on parallelization of one computationally heavy algorithm on general purpose processors (GPP) and graphic processing units (GPU). Although GPU implementation makes an intensive use of atomic intrinsics, it gave the highest performances. Only a dual 6-core GPP catches up its performances. Both architectures have been evaluated with different APIs: OpenMP, CUDA and OpenCL.

# B3: THREAD MANAGEMENT

# EXPERIENCE USING LAZY TASK CREATION IN OPENMP TASK FOR THE UTS BENCHMARK

*Adnan adnan, Mitsuhisa Sato*

University of Tsukuba, Japan

**Abstract**

In parallel programming, the problems of overhead and load imbalance need to be considered. To solve the overhead problem, specifically the task creation overhead, lazy task creation is known to be the most efficient technique. In this paper, the experience applying lazy task creation to the OpenMP task which uses the StackThreads/MP fine grain thread library is presented. StackThreads/MP, enhanced with lazy task creation and two extended work-stealing strategies, is used to implement an OpenMP task. The enhanced StackThreads/MP was evaluated using two classes of work-load distributions of the UTS benchmark. Comparing the results of evaluation using StackThreads/MP, Cilk, GCC, and the Intel OpenMP compiler, it was shown that the proposed enhancements improve the original StackThreads/MP so that the StackThreads/MP performance is more scalable compared with previous versions.

# FOLDING APPLICATIONS INTO HIGH DIMENSIONAL TORUS NETWORKS

*Lukas Arnold*

Forschungszentrum Juelich, Germany

**Abstract**

In high performance computing many scientific applications have next-neighbour dominated communication patterns. In this case each process communicates only to its neighbours with respect to the logical application mapping. Such applications strongly benefit from torus networks, as these networks provide direct connections for all neighbouring nodes. However, this assumes that neighbours in the application mapping are also neighbours in the torus network. This contribution aims to help application developer and user to efficiently utilize torus networks. The main idea to get a perfect mapping is to fold the additional network dimensions in an appropriate way. Here mainly the combination of additional dimensions and space filling curves are used. Synthetic and scientific application (PEPC, PSC, RACOON) benchmarks are used to demonstrate the benefits of a good mapping.

# COMPOSABLE PARALLELISM FOUNDATIONS IN THE INTEL® THREADING BUILDING BLOCKS TASK SCHEDULER

*Andrey Marochko, Alexey Kukanov*

Intel Corporation, Russian Federation

**Abstract**

Since its introduction, Intel® Threading Building Blocks (Intel® TBB) has evolved from a compact library implementing basic patterns of computational parallelism to one of the industry's leading parallel frameworks. Many of its customers were interested in seamless integration of parallelized code fragments and components into the logic of complex tiered and component-based applications, sometimes in preference to raw performance and scalability. The growing application field has revealed a number of limitations and suboptimal tradeoffs in various parts of the library, including its task scheduler. While the basic design of the Intel® TBB task scheduler has already been described in publications, evolution significantly affected its properties. This article provides a comprehensive overview of new usage models supported by the scheduler, with an accent on the features facilitating construction of highly composable parallel solutions. We also discuss design decisions and implementation details that influence the performance and behavior of Intel® TBB based applications. This article may be of value both to application developers, by explaining the semantics of the new APIs in Intel® TBB, and to researchers and engineers working on other parallelization solutions, by examining factors that influence design decisions in this field.

# C3: PERFORMANCE MODELING AND ANALYSIS I

# JUBE-BASED AUTOMATIC TESTING AND PERFORMANCE MEASUREMENT SYSTEM FOR FUSION CODES

*Andreas Galonska, Wolfgang Frings, Paul Gibbon, Dmitriy Borodin, Andreas Kirschner*

Forschungszentrum Jülich GmbH, Germany

**Abstract**

Numerical HPC modelling is an important technique to understand and predict various aspects of nuclear fusion experiments. According code development involves time consuming benchmarking of various versions. In addition, optimization of code performance is necessary for HPC applications. The present paper introduces the Automatic Testing and Performance Measurement System (ATPMS) - based on JuBE (Jülich Benchmarking Envirnoment) - which provides a solution for both tasks: Automatic Testing of new code versions and Time Benchmarking. The functionality of the system will be demonstrated using the example of the ERO code, however, it can also be applied to other fusion codes.

# VISUALIZATION OF MPI(-IO) DATATYPES

*Julian Martin Kunkel, Thomas Ludwig*

University of Hamburg, Germany

**Abstract**

To permit easy and efficient access to non-contiguous regions in memory for communication and I/O the message passing interface offers nested datatypes. Since nested datatypes can be very complicated, the understanding of non-contiguous access patterns and the debugging of wrongly accessed memory regions is hard for the developer. HDTrace is an environment which allows to trace the behavior of MPI programs and to simulate them for arbitrary virtual cluster configuration. It is designed to record all MPI parameters including MPI datatypes. In this paper we present the capabilities to visualize usage of derived datatypes for communication and I/O accesses -- a simple hierarchical view is introduced which presents them in a compact form and allows to dig into the nested datatypes. File regions accessed in non-contiguous I/O calls can be visualized in terms of the original datatype. The presented feature assists developers in understanding the datatype layout and spatial I/O access patterns of their application.

# OPEN TRACE FORMAT 2 – THE NEXT GENERATION OF SCALABLE TRACE FORMATS AND SUPPORT LIBRARIES

*Michael Wagner, Dominic Eschweiler, Markus Geimer, Andreas Knüpfer, Wolfgang E. Nagel, Felix Wolf*

ZIH, TU Dresden, Germany

**Abstract**

A well designed event trace data format is the basis of all trace-based analysis methods. In this paper, we introduce the Open Trace Format Version 2 (OTF2). It is a major re-design based on the experiences of its predecessor formats, the Open Trace Format (Version 1) and the EPILOG trace format. It comes with a new file encoding, a close integration in the trace recording system Score-P, and a number of improvements for performance and scalability. Besides the actual format, it consists of a read/write support library with a powerful API plus supportive tools, which are distributed as Open Source software. Furthermore, OTF2 will serve as a joint data source for the analysis tools Scalasca and Vampir in the near future.

# A4: ALGORITHMS II

# DESIGN AND EVALUATION OF A PARALLEL EXECUTION FRAMEWORK FOR THE CLEVER CLUSTERING ALGORITHM

*Chun-Sheng Chen, Nauful Shaikh, Panitee Charoenrattanaru, Christoph F. Eick, Nouhad Rizk, Edgar Gabriel*

University of Houston, United States of America

**Abstract**

Data mining is used to extract valuable knowledge from vast pools of data. Due to the computational complexity of the algorithms applied and the problems of handling large data sets themselves, data mining applications often require days to perform their analysis when dealing with large data sets. This paper presents the design and evaluation of a parallel computation framework for CLEVER, a prototype-based clustering algorithm which has been successfully used for a wide range of application scenarios. The algorithm supports plug-in fitness functions and employs randomized hill climbing to maximize a given fitness function. We explore various parallelization strategies using OpenMP and CUDA, and evaluate the performance of the parallel algorithms for three different data sets. Our results indicate a nearly linear scalability of the parallel algorithm using multi-core processors, reducing the execution time and allowing to solve problems which were considered not feasible with the sequential version of CLEVER.

# THE BL-OCTREE: AN EFFICIENT DATA STRUCTURE FOR DISCRETIZED BLOCK-BASED ADAPTIVE MESH REFINEMENT

*Ashley Nicole Zebrowski, Frank Löffler, Erik Schnetter*

Louisiana State University, United States of America

**Abstract**

Adaptive mesh refinement (AMR) algorithms require efficient structures to store and operate upon volumetric grid data that will be used to distribute work among processes. When inefficient algorithms are used, AMR-enabled software may have its scaling curve negatively impacted as processes are starved for work. We present a variable-resolution octree-based structure which efficiently handles the set operations needed for AMR operations such as calculating process communication schedules and buffer zones. Results indicate a marked improvement in execution times when large amounts of volumetric data are processed, but at the price of a structure creation and traversal penalty for smaller amounts of data. The BL-octree ("blocktree") data structure presented in this paper will therefore be of use in massively parallel AMR environments, where the cost of distributing AMR workloads across many tens of thousands or more processes can be reduced

# B4: MULTICORES I

# THE PEPPHER APPROACH TO PROGRAMMABILITY AND PERFORMANCE PORTABILITY FOR HETEROGENEOUS MANY-CORE ARCHITECTURES

*Sabri Pllana, Siegfried Benkner, Jesper Traff, Philippas Tsigas, Andrew Richards, Raymond Namyst, Beverly Bachmaier, Christoph Kessler, David Moloney, Peter Sanders*

University of Vienna, Austria

**Abstract**

The European FP7 project PEPPHER is addressing programmability and performance portability for current and emerging heterogeneous many-core architectures. As its main idea, the project proposes a multi-level parallel execution model comprised of potentially parallelized components existing in variants suitable for different types of cores, memory configurations, input characteristics, optimization criteria, and couples this with dynamic and static resource and architecture aware scheduling mechanisms. Crucial to PEPPHER is that components can be made performance aware, allowing for more efficient dynamic and static scheduling on the concrete, available resources. The flexibility provided in the software model, combined with a customizable, heterogeneous, memory and topology aware run-time system is key to efficiently exploiting the resources of each concrete hardware configuration. The project takes a holistic approach, relying on existing paradigms, interfaces, and languages for the parallelization of components, and develops a prototype framework, a methodology for extending the framework, and guidelines for constructing performance portable software and systems -- including paths to migration of existing software -- for heterogeneous many-core processors. This paper gives a high-level project overview, and presents a specific example showing how the PEPPHER component variant model and resource-aware run-time system enable performance portability of a numerical kernel.

# AN EFFICIENT PARALLEL SET CONTAINER FOR MULTICORE ARCHITECTURES

*Álvaro de Vega, Diego Andrade, Basilio B. Fraguela*

University of A Coruña, Spain

**Abstract**

Multicores are now the norm and new generations of software must take advantage of the presence of several cores in a given architecture. Parallel programming requires specific skills beyond from those required for the development of traditional sequential programs. The usage of parallel libraries is one of the best ways to facilitate parallel programming, as they do not require new compilers and they allow to parallelize sequential codes without big efforts by the programmer. This paper presents an efficient and portable parallel set container. This container has been used to program parallel versions of several algorithms. The experimental results show that the container facilitates the programmability and achieves a good performance on multicore systems.

# C4: PARAFPGA MINI-SYMPOSIUM I

# ACCELERATING HMMER SEARCH USING FPGA GRID

*Tokokazu Takagi, Tsutomu Maruyama*

University of Tsukuba, Japan

**Abstract**

HMMER is one of the most used software tools for sensitive profile HMM searches of biological sequence databases. HMMER can be accelerated by comparing a profile HMM with sequences in parallel using a CPU grid. HMMER is, however, still a cpu-intensive program. The search on each node on the grid can be further accelerated by using dedicated hardware. In this paper, we describe an approach for accelerating HMMER search using an FPGA grid. In our system, two prototype circuits which share the same basic processing units are prepared in advance, and the circuit for each size FPGA is generated from the prototype circuits. In the FPGA grid, the difference of the processing speed and data transfer speed becomes much larger than CPU grids. In order to maintain the load balance on each node, new load balancing methods are introduced and evaluated.

# CIRCUIT DESIGN IN CΛASH

*Gerard J.M. Smit, Jan Kuper, Christiaan P.R. Baaij*

University of Twente, Netherlands

**Abstract**

The hardware design environment CλaSh is recently developed at the University of Twente. Features of CλaSH are: 1) it is based on a mathematical language to specify hardware architectures, 2) it offers various abstraction mechanisms such as higher order functions, functional abstraction, various composition mechanisms, polymorphism, type derivation, 3) it offers an immediate simulation of hardware designs to test functional correctness, 4) it offers correctness preserving transformations and 5) it is able to translate specifications into synthesizable VHDL for implementation on an FPGA or an ASIC. Because of it is close to mathematics, CλaSH is written in the functional programming language Haskell. Specifications in CλaSH are concise and well readable, thus increasing a designer's productivity as experience showed. We will discuss the approach of CλaSH and present some non-trivial examples of specifications.

# A5: MASSIVE PARALLELISM I

## PROCESSING WITH A MILLION CORES

*Jeffrey Reeve, Andrew Brown, Steve Furber, David Lester*

University of Southampton, United Kingdom

**Abstract**

Traditionally, the performance-limiting attributes of a multi-processor machine are memory bandwidth, the need to maintain state coherence, and simple synchronisation issues, or some combination of these. As the system size increases, so too does the relative cost overhead of solving these problems. The SpiNNaker engine is a million core system with over seven terabytes of distributed memory whose fundamental design axioms neatly sidestep these problems: the processors are not synchronised, the memory is not coherent, and the inter-processor communication is non-deterministic. How, then, can we perform meaningful computations with this architecture?

# USING FAST AND ACCURATE SIMULATION TO EXPLORE HARDWARE/SOFTWARE TRADE-OFFS IN THE MULTI-CORE ERA

*Wim Heirman, Trevor E. Carlson, Souradip Sarkar, Pieter Ghysels, Wim Vanroose, Lieven Eeckhout*

Ghent University, Belgium

**Abstract**

Writing well-performing parallel programs is challenging in the multi-core processor era. In addition to achieving good per-thread performance, which in itself is a balancing act between instruction-level parallelism, pipeline effects and good memory performance, multi-threaded programs complicate matters even further. These programs require synchronization, and are affected by the interactions between threads through sharing of both processor resources and the cache hierarchy. At the Intel Exascience Lab, we are developing an architectural simulator called Sniper for simulating future exascale-era multi-core processors. Its goal is twofold: Sniper should assist hardware designers to make design decisions, while simultaneously providing software designers with a tool to gain insight into the behavior of their algorithms and allow for optimization. By taking architectural features into account, our simulator can provide more insight into parallel programs than what can be obtained from existing performance analysis tools. This unique combination of hardware simulator and software performance analysis tool makes Sniper a useful tool for a simultaneous exploration of the hardware and software design space for future high-performance multi-core systems.

# B5: GPU APPLICATIONS II

# LATTICE BOLTZMANN FOR LARGE-SCALE GPU SYSTEMS

*Alan Gray, Kevin Stratford, Alistair Hart, Alan Richardson*

The University of Edinburgh, United Kingdom

**Abstract**

We describe the enablement of the Ludwig lattice Boltzmann parallel fluid dynamics application, designed specifically for complex problems, for massively parallel GPU-accelerated architectures. NVIDIA CUDA is introduced into the existing C/MPI framework, and we have given careful consideration to maintainability in addition to performance. Significant performance gains are realised on each GPU through restructuring of the data layout to allow memory coalescing and the adaptation of key loops to reduce off-chip memory accesses. The halo-swap communication phase has been designed to efficiently utilise many GPUs in parallel: included is the overlapping of several stages using CUDA stream functionality. The new GPU adaptation is seen to retain the good scaling behaviour of the original CPU code, and scales well up to 256 NVIDIA Fermi GPUs (the largest resource tested). The performance on the NVIDIA Fermi GPU is observed to be up to a factor of 4 greater than the (12-core) AMD Magny-Cours CPU (with all cores utilised) for a binary fluid benchmark.

# HIGH-FIDELITY REAL-TIME ANTISHIP CRUISE MISSILE MODELING ON THE GPU

*Christopher Scannell*

US Naval Research Laboratory, United States of America

**Abstract**

The United States Navy is actively researching techniques for creating high-fidelity, real-time simulations of antiship cruise missiles (ASCM) in order to develop improved defensive countermeasures for Navy ships. One active area of investigation is the combined use of OpenMP and MPI to reach real-time constraints on stand-alone cluster computers with high-speed interconnect fabrics. The separate compute nodes of the supercomputer calculate the successive responses of a single cruise missile to successive reflections of the RF transmitter radar returns from the target ship in a pipeline fashion using MPI. Numerically intensive portions of the calculation of the missile-ship system behavior for an individual RF pulse can be calculated in parallel simultaneously on the individual nodes of the supercomputer using OpenMP. The speed at which these portions can be calculated directly determines the length of the pipeline and thus the total number of computing nodes required. This approach incurs some approximations into the simulation that are proportional to the length of the pipeline because there is a feedback from the ship-radar response back to the missile guidance. While this use of OpenMP has proven effective, it is limited by the number of cores available at each node. This code, however, presents opportunities for parallelism well beyond the available computational resources at each node. Additionally, the ratio of computation to data transfer for this portion of the simulation is very high. These two factors have led us to investigate executing the most compute-intensive portion, the calculation of the RF responses of the individual ship scatterers, on Graphics Processing Units (GPUs).

## C5: PARAFPGA MINI-SYMPOSIUM II

# DYNAMIC RECONFIGURABLE PATTERN MATCHER FOR REGULAR EXPRESSIONS ON FPGA

*Tom Davidson, Mattias Merlier, Karel Bruneel, Dirk Stroobandt*

Ghent University, Belgium

**Abstract**

In this article we describe how to expand a partially dynamic reconfigurable pattern matcher for regular expressions presented in previous work by Divyasree and Rajashekar. The resulting, extended, pattern matcher is fully dynamically reconfigurable. First, the design is adapted for use with parameterisable configurations, a method for Dynamic Circuit Specialization. Using parameterisable configurations allows us to achieve the same area gains as the hand crafted reconfigurable design, with the benefit that parameterisable configurations can be applied automatically. This results in a design that is more easily adaptable to specific applications and allows for an easier design exploration. Additionally, the parameterisable configuration implementation is also generated automatically, which greatly reduces the design overhead of using dynamic reconfiguration. Secondly, we propose a number of expansions to the original design to overcome several limitations in the original design that constrain the dynamic reconfigurability of the pattern matcher. We propose two different solutions to dynamically change the character that is matched in a certain block. The resulting pattern matcher, after these changes, is fully dynamically reconfigurable, all aspects of the implemented regular expression can be changed at run-time.

# A FRAMEWORK FOR SELF-ADAPTIVE COLLABORATIVE COMPUTING ON RECONFIGURABLE PLATFORMS

*Michiel Van Tol, Zdenek Pohl, Milan Tichy*

University of Amsterdam, Netherlands

**Abstract**

As the number and complexity of computing devices in the environment around us increases, it is interesting to see how we could exploit that and glue them together to create larger co-operative distributed systems. This paper describes a framework for dynamically aggregating and configuring processing resources in order to meet local requirements and constraints. The capability of this framework is demonstrated by a case study using an adaptive least mean squares filter (ALMS) application. ALMS improves convergence of least mean squares filters at the cost of more resources, and allows us to demonstrate abilities of the framework such as task offloading and run-time adaptation to available resources.

# A6: NUMERICAL ALGORITHMS II

# ACCELERATING GRID KERNELS FOR VIRTUAL SCREENING ON GRAPHICS PROCESSING UNITS

*Irene Sánchez-Linares, Horacio Pérez-Sánchez, José Manuel García*

University of Murcia, Spain

**Abstract**

The discovery of new drugs can be drastically accelerated with the use of Virtual Screening (VS) methods, ongoing trend in medical research. VS methods need to screen chemical compound databases with millions of ligands but they are com- pletely constrained by the access to computational resources. We worked on the adaptation of their main bottlenecks to GPUs using a grid approach with different interpolation methods. In our first studies for medium size proteins, our implemen- tation runs around 30 times faster than previous GPU implementations.

# PARALLELISM ON THE NON-NEGATIVE MATRIX FACTORIZATION

*Edgardo M. Mejia-Roa, Carlos Garcia, Jose Ingacio Gomez, Manuel Prieto, Christian Tenllado, Alberto Pascual-Montano, Francisco Tirado*

Complutense University of Madrid (UCM). CIF: Q-2818014-I, Spain

**Abstract**

A great interest has been given to the Nonnegative Matrix Factorization (NMF) due to its ability of extracting highly-interpretable parts from data sets. Nonetheless, its usage is hindered by the computational complexity when processing large matrices. In this paper, we present three implementations of NMF to analyze the benefits that parallelism can provide to this method on data sets of different size. Our first implementation is based on Message Passing Interface (MPI). Input data is distributed among multiple processors. The second version uses CUDA in Graphics Processing Units (GPUs). Large data sets are blockwise transferred and processed. Finally, we combine both paradigms. Compared to the single-GPU version, it achieves super linear speedups when data portions assigned to each GPU fit into their memory and can be transferred only once.

# B6: HETEROGENEOUS COMPUTING

# PARALLEL LIKELIHOOD FUNCTION EVALUATION ON HETEROGENEOUS MANY-CORE SYSTEMS

*Alfio Lazzaro, Sverre Jarp, Julien Leduc, Andrzej Nowak, Yngve Sneen Lindal*

CERN, Switzerland

**Abstract**

In the work presented in this paper we show a parallel implementation which allows to run the evaluation of the likelihood function for data analysis methods on CPU and GPU cooperatively, so that the workload can be split and balanced in the two devices (hybrid likelihood function evaluation). Therefore the data analysis applications can take full advantage from users commodity systems, like desktops and laptops, using entirely the hardware at disposal. We implemented a scheduler of the tasks so that we can statically balance the workload between the CPU and GPU for the evaluation of likelihood function. The CPU parallelization is implemented using OpenMP, and the GPU implementation is based on OpenCL. We show the comparison of the performance of these implementations on different hardware systems from different vendors and different likelihood function definitions. Finally we present the results when running the hybrid evaluation, considering the different possible combinations of the CPU and GPU implementations, and we show the scalability tests for real data analysis carried out in the high energy physics community.

# A MODELBASED SOFTWARE GENERATION APPROACH QUALIFIED FOR HETEROGENEOUS GPGPU-ENABLED PLATFORMS

*Holger Endt, Lothar Stolz, Martin Wechs, Walter Stechele*

BMW Research and Technology, Germany

**Abstract**

Being meanwhile rather common coprocessors for accelerating graphics, Graphics Processing Units (GPUs) today get more and more used to compute general purpose tasks, too. Having successfully shown usecases over the whole spectrum of applications, the GPU might become one standard device within a platform construction kit. Also in automotive environments, dedicated GPUs will be introduced in the next generation of navigation and multimedia-platforms to enhance the user experience within the infotainment domain and to satisfy the growing demand for 3D navigation. Embedded Plattforms typically consist of DSPs, CPUs, FPGAs and/or ASICs. Each of those processing devices not only offers different capabilities but does therefore also require different approaches to implement a specific algorithm. This makes it interesting to describe a function in a very abstract manner and then use automated software generation tools to generate source-code for each of the specific target platforms. Hence, to apply highly parallel GPU devices as a general purpose coprocessor within such an embedded platform construction kit, too, we investigate a design methodology based on a model-driven development (MDD) approach. For automatically deploying code to future platforms with built-in GPUs, our subsequently following software generation approach hereby automatically exploits data-level concurrencies in these models, and thus applies the goal of automatically generating applications taking benefit of GPGPU acceleration.

# C6: PARAFPGA MINI-SYMPOSIUM III

# RECONFIGURABLE COMPUTING CLUSTER PROJECT: A FIVE-YEAR PERSPECTIVE

*Ron Sass, Scott Buscemi*

University of North Carolina at Charlotte, United States of America

**Abstract**

The Reconfigurable Computing Cluster project has been investigating broad question of what, if any, role FPGAs may play in parallel computing. In 2006, a small-scale experimental machine consisting of 64 FPGAs connected in 3-D torus was constructed and has been at core of many more focused investigations. The article provides a summary of the architecture, the scientific and pragmatic discoveries enabled by the machine, and two more FPGA-based parallel computers that are presently under development.

# FROM MONO-FPGA TO MULTI-FPGA EMULATION PLATFORM FOR NOC PERFORMANCE EVALUATIONS

*Junyan Tan, Virginie Fresse, Frederic Rousseau*

Hubert Curien Laboratory, France

**Abstract**

Experimental approaches used for architecture exploration and validation are often based on configurable logic device such as Field Programmable Gate Array (FPGA). System on Chip (SoC) and Network on Chip (NoC) architectures require multi-FPGA platforms as the resources of a single FPGA may not be big enough. Partitionning a NoC on a multi-FPGA platform requires special techniques for allocating communication channels, physical links and suitable resource allocation scheme. It is a time consuming process reducing the exploration space and validation. In this paper, we present a scalable emulation platform and its associated design flow based on a multi FPGA approach that allows quick exploration, evaluation and comparison of a wide range of NoC solutions. The efficiency of our approach is illustrated through the deployment of the Hermes NoC and its exploration on a mono FPGA and multi-FPGA platform with several traffic implementations.

# I: INDUSTRIAL SESSION

# CRAY'S APPROACH TO HETEROGENEOUS COMPUTING

*Roberto Ansaloni, Alistair Hart*

Cray, Italy

**Abstract**

There seems to be a general consensus among the HPC community, about the impossibility to reach hexascale performance with systems based only on multi-core chips. Heterogeneous nodes where the traditional CPU is combined with many-core accelerators have the potential to provide a much more energy efficient solution capable to overcome the power consumption challenge. However this emerging hybrid node architecture is expected to pose significant challenges for applications developers in order to efficiently program these systems and achieve a significant fraction of the available peak performance. This is certainly the case for today's GPU-based accelerators with separate memory space, but it also holds true for future unified nodes with CPU and many-core accelerator on chip sharing common memory. In this talk we'll describe Cray's approach to heterogeneous computing and the first Cray hybrid supercomputing system, the Cray XK6, with its unified X86/GPU programming environment. We'll also describe Cray's proposal to extend the OpenMP standard to provide a portable accelerator programming environment capable of supporting a wide range of accelerators and being forward scalable in time. Finally we'll show how the current set of OpenMP accelerator directives can be used to implement a GPU version of the Himeno benchmark on a Cray XK6 early prototype system.

# INTEGRATED SIMULATION WORKFLOWS IN COMPUTER AIDED ENGINEERING ON HPC RESOURCES

*Florian Niebling, Andreas Kopecki, Martin Aumueller*

HLRS, Germany

**Abstract**

With the availability and easy accessibility of high performance computing resources, product development in engineering application shifted from experiments and model tests to computer simulations almost exclusively. Even after the initial construction of a satisfying design in the rapid prototyping phase, usually much potential for optimizations remains. To achieve an optimal design requires the execution of multiple simulation workflows with different parameters, concerning optimal operation at different operating points. Gaining insight from simulation results into how the changes to a model affect the performance of the simulated machine requires an interactive exploration of these datasets through - ideally - in situ post-processing. As the complexity of simulation data increases, traditional post-processing of datasets on single workstations is no longer possible due to limitations in system memory, network bandwidth and compute power. In this paper, we present how remote HPC resources can be used for interactive design, simulation and interactive analysis to support engineering workflows in product development.

# EXASCALE NEEDS AND CHALLENGES FOR INDUSTRIAL APPLICATIONS

*Jean-Claude André*

EESI, France

**Abstract**

The EESI (European Exascale Software Initiative), a support action under Famework Programme 7 of the European Commission, is aimed at preparing, and possibly organizing, the transition to exascale high-performance computing. It includes the discussion of industrial roadmaps to exascale, and challenges to be addressed, together with roadmaps corresponding to more traditional academic and scientific domains. Here we shall present and discuss a few industrial roadmaps, including: - aeronautics: improved predictions of complex flow phenomena around full aircraft configurations, multi-disciplinary analysis and design, real time simulation of maneuvering aircraft, ...; - seismic, oil and gas: largely embarrassingly parallel, major problems are programming model, memory access and data management, need up to zetaflop for full inverse problem; - combustion: turbulent combustion modeling, LES in large scale reactors, coupling multi-physics, multi-cycle engine, ...; - nuclear energy: steady CFD calculations on complex geometries, RANS, LES and quasi-DNS type calculations under uncertainties, Monte Carlo ultimate neutronic transport calculation. Corresponding common main issues to be addressed will also be discussed:

# A7: CLOUD COMPUTING I

# ENERGY AWARE CONSOLIDATION POLICIES

*Mehdi Sheikhalishahi, Ignacio Martin Llorente, Lucio Grandinetti*

University of Calabria, Italy

**Abstract**

Green computing denotes energy efficiency in all components of computing systems i.e. hardware, software, local area, etc. Placement of jobs is a critical decision to address PoorPerformance, ResourceContention and HighEn- ergyConsumption problems. Consolidation policies are one of the sources of information for effective placement of jobs in any computing paradigm. We design two Energy Aware consolidation policies i.e. SimpleEAConsolidation and EA- ConsolidationOverLeaseTimeHorizon according to workload characteristics to model mainly the resource contention among jobs and implement them as host selection policies of a sched- uler. We evaluate our policies in HPC and cloud computing paradigms with traces from Parallel Workload Archives. The simulation results show improvements on resource contention metric against the Greedy host selection policy.

# REMOTE UTILIZATION OF OPENCL FOR FLEXIBLE COMPUTATION OFFLOADING USING EMBEDDED ECUS, CE DEVICES AND CLOUD SERVERS

*Holger Endt, Kay Weckemann*

BMW Research and Technology, Germany

**Abstract**

In the automotive sector, the product cycles are by means longer than in the mobile markets. This often leads to long timeframes for introducing innovations of the infotainment domain in automobiles. In this paper, we present an approach that allows to downsize on-board ECUs to support the typical case of running a specified number of applications while featuring the opportunity to back-off computational power in worst case scenarios. Furthermore, this concept enables car manufacturers to offer new functionality to customers over the complete product-lifetime without the need to upgrade the hardware of ECUs or to have cost intensive automotive high performance hardware available on-board. This is achieved by enabling automotive applications to utilize either built-in computational power, CE (Consumer Electronics) devices like the customer's smartphone or the computational power of cloud servers. Therefore, we extend OpenCL, an industry standard that is supported by many different companies, to work locally on automotive platforms as well as remotely over the network. This becomes possible with the emergence of IP-based in-car networks.

# B7: LANGUAGES II

# SAC ON A NIAGARA T3-4 SERVER: LESSONS AND EXPERIENCES

*Clemens Grelck, Roeland Douma*

University of Amsterdam, Netherlands

**Abstract**

The Sparc T3-4 server provides up to 512 concurrent hardware threads, a degree of concurrency that is unprecedented in a single server system. This paper reports on how the automatically parallelising compiler of the data-parallel functional array language SAC copes with up to 512 execution units. We investigate three different numerical kernels that are representative for a wide range of applications: matrix multiplication, convolution and 3-dimensional FFT. We show both the high-level declarative coding style of SAC and the performance achieved on the T3-4 server. Last not least, we draw conclusions for improving our compiler technology in the future.

# DECLARATIVE PARALLEL PROGRAMMING FOR GPUS

*Eric Holk, William Byrd, Nilesh Mahajan, Jeremiah Willcock, Arun Chauhan, Andrew Lumsdaine*

Indiana University, United States of America

**Abstract**

GPUs are increasingly being used for general purpose computing and show tremendous potential in shaping future high performance computers with their highly energy efficient architectures. Unfortunately, GPGPU programming is notoriously difficult, and programming hybrid clusters doubly so. While frameworks such as NVIDIA's Compute Unified Device Architecture (CUDA) and Khronos Group's OpenCL, have made programming GPUs easier than before these frameworks still afford only a very low-level programming model. For example, programmers must write boilerplate code and must handle low-level details of data layout and data movement across various types of memory, determine how many blocks and threads are required for computation, etc. Programming hybrid clusters is even harder. We present a novel declarative approach to programming GPUs, in which the programmer specifies "what" but not "how" of computation, data layout, and memory movement. This gives the programmers sufficient control to write efficient code, while freeing them from error-prone low-level programming. With well-defined and verifiable semantics, our declarative language eliminates common programming mistakes, such as deadlocks, and paves a way for sophisticated compiler optimizations such as intelligent partitioning of computation across CPU and GPU, and minimizing data movement across memory types.

# C7: EXASCALE MINI-SYMPOSIUM I

## HYBRID PARALLEL PROGRAMMING WITH MPI/STARSS

*Jesus Labarta, Rosa Badia, Eduard Ayguade, Marta Garcia, Vladimir Marjanovic*

BSC, Spain

**Abstract**

This talk will describe how hybrid MPI/StarSs programming provides an efficient way to program future clusters of multicore nodes. The use of the programming model results in achieving overlap between communication and computation while maintaining a simple source code structure. Additional benefits include tolerance to limited bandwidth availability and tolerance to OS noise. The model also supports the implementation of automatic load balancing techniques at the level of the runtime that are complementary to whatever user level load balancing is applied. Examples with different applications parallelized within the TEXT project will be described.

# GPI -- GLOBAL ADDRESS SPACE PROGRAMMING INTERFACE. MODEL, EXPERIENCES, SCALABILITY AND THE FUTURE

*Mirko Rhan*

Fraunhofer ITWM, Germany

**Abstract**

Systems based on a Partitioned Global Address Space (PGAS) offer significant performance and productivity advantages over the Message Passing approach. Not only they enable the best-possible network performance, by not copying data, they also allow for full memory bandwidth while keeping a single programming model. The Global Address Space Programming Interface (GPI) is the PGAS-API developed at the Fraunhofer ITWM. It is a thin layer, that delivers the full performance of RDMA-enabled networks directly to the application without interrupting the CPU. Besides the capability to put or get data from remote nodes with maximum speed, the GPI comes with several useful functions, such as fast collective operations or easy to use global atomic counters. We will introduce the GPI briefly and compare the application performance between GPI implementations of a large CFD code (TAU) and of a quantum physics code (BQCD) and corresponding MPI implementations. We will show which steps are necessary to (re-)implemented existing MPI Codes in GPI. We will see that the GPI implementations not only perform much better and are much more robust but more importantly, they scale better at the same time. We will argue, that heterogeneous systems with many cores cannot longer rely on message passing and propose the use of the GPI instead.

# A8: NUMERICAL ALGORITHMS III

# EXPLOITING FINE-GRAIN PARALLELISM IN RECURSIVE LU FACTORIZATION

*Jack Dongarra, Mathieu Faverge, Hatem Ltaief, Piotr Luszczek*

University of Tennessee, United States of America

**Abstract**

The LU factorization is an important numerical algorithm for solving system of linear equations. This paper proposes a novel approach for computing the LU factorization in parallel on multicore architectures. It improves the overall performance and also achieves the numerical quality of the standard LU factorization with partial pivoting. While the update of the trailing submatrix is computationally intensive and highly parallel, the inherently problematic portion of the LU factorization is the panel factorization due to its memory-bound characteristic and the atomicity of selecting the appropriate pivots. We remedy this in our new approach to LU factorization of (narrow and tall) panel submatrices. We use a parallel fine-grained recursive formulation of the factorization. It is based on conflict-free partitioning of the data and lock-less synchronization mechanisms. Our implementation lets the overall computation naturally flow with limited contention. Our recursive panel factorization provides the necessary performance increase for the inherently problematic portion of the LU factorization of square matrices. A large panel width results in larger Amdahl's fraction as our experiments have revealed which is consistent with related efforts. The performance results of our implementation reveal superlinear speedup and far exceed what can be achieved with equivalent MKL and/or LAPACK routines.

# PARAREAL ACCELERATION OF MATRIX MULTIPLICATION

*Toshiya Takami, Akira Nishida*

Kyushu University, Japan

**Abstract**

The parareal-in-time algorithm has been used in time-domain parallelization of scientific problems described by ordinary differential equations and partial differential equations, where this algorithm is considered as parallelism at an application level. In this paper, applicability of the parareal algorithm to simple linear transformations is studied in convergence and speedup ratio. In particular, matrix-vector multiplication which is often used in Krylov subspace methods is the target of this study, where a single matrix is separated into a sum of two matrices, a sparse matrix and its perturbation. The convergence of the procedure is analyzed for calculations of a series of vectors defined by the matrix multiplication. Because of the linearity of this problem, convergence property is easily analyzed by the spectral radius of those separated matrices. The next analysis is to measure the actual speedup ratio in a distributed parallel machine and compared to theoretical prediction. A real-symmetric matrix case and a unitary matrix case are measured in various matrix sizes and length of the series on a machine with hundreds of CPU's. Based on the results, appropriate sizes of matrices and length of the series for this scheme will be discussed.

# B8: MULTICORES II

# USE OF HIGH ACCURACY AND INTERVAL ARITHMETIC ON MULTICORE PROCESSORS

*Carlos Amaral Holbig, Andriele Busatto Do Carmo, Viviane Linck Lara, Luis Paulo Arendt*

Universidade de Passo Fundo, Brazil

**Abstract**

In this paper we describe the integration and use of a high accuracy and interval arithmetic library (C-XSC) on multicore processors. With this integration we have a parallel environment supporting high accuracy and verified computing. The purpose of this work is to provide an environment where scientific computing problems can be solved with speed, accuracy and reliability, where the result verification may be carried out automatically by the computer. To validate the environment developed in this work some tests were developed and their results will be discussed in this paper.

# ENGINEERING CONCURRENT SOFTWARE GUIDED BY STATISTICAL PERFORMANCE ANALYSIS

*Clemens Grelck, Kevin Hammond, Heinz Hertlein, Philip Hölzenspies, Chris Jesshope, Raimund Kirner, Bernd Scheuermann, Alex Shafarenko, Iraneus te Boekhorst, Volkmar Wieser*

SAP AG, Germany

**Abstract**

This paper introduces the ADVANCE approach to engineering concurrent systems using a new component-based approach. A cost-directed tool-chain maps concurrent programs onto emerging hardware architectures, where costs are expressed in terms of programmer annotations for the throughput, latency and jitter of components. These are then synthesized using advanced statistical analysis techniques to give overall cost information about the concurrent system that can be exploited by the hardware virtualisation layer to drive mapping and scheduling decisions. Initial performance results are presented, showing that the ADVANCE technologies provide a promising approach to dealing with near- and future-term complexities of programming heterogeneous multi-core systems.

# C8: EXASCALE MINI-SYMPOSIUM II

# TEMANEJO - A DEBUGGER FOR TASK BASED PARALLEL PROGRAMMING MODELS

*Steffen Brinkmann, Christoph Niethammer, José Gracia, Rainer Keller*

University of Stuttgart, Germany

**Abstract**

We present a debugger for task based parallelisation models as e.g. StarSs. The challange in debugging StarSs applications lies in the fact that tasks are scheduled asynchronously at runtime. In this model tasks are scheduled dynamically in accordance to the data dependencies between them. Our tool assists the programmer in the debugging process by visualising the task dependency graph and allowing to control the scheduling of tasks. The toolset consists of the library AYUDAME which communicates with the StarSs runtime on one side and with the external tool TEMANEJO on the other side. TEMANEJO provides a graphical user interface with which the application can be analysed and controlled.

# CHARACTERIZING PARALLEL I/O PERFORMANCE USING THE TAU PERFORMANCE SYSTEM

*Sameer Shende, Allen D. Malony, Wyatt Spear, Karen Schuchardt*

BSC, Spain

**Abstract**

TAU is an integrated toolkit for performance instrumentation, measurement, and analysis. It provides a flexible, portable, and scalable set of technologies for performance evaluation on extreme-scale HPC systems. This paper describes the design and implementation of a new tool, tau_gen_wrapper, to wrap external libraries using three techniques - preprocessor based substitution, linker based instrumentation, and library preloading based replacement of routines. We demonstrate this wrapping technology in the context of intercepting the POSIX I/O and the HDF5 library. Runtime preloading of a wrapper interposition library simplifies instrumentation by spawning an uninstrumented application using a special script. Linker based instrumentation allows TAU to wrap external libraries by simply re-linking the object files using a compiler script and allows us to measure the volume and bandwidth of individual read and write operations. The paper will include I/O data collected from the Global Cloud Resolution Model (GCRM) application on the Cray XE6 system. This scheme allows TAU to track I/O using linker level instrumentation for statically linked executables and attribute the I/O to specific code regions. It also addresses issues encountered in collecting the performance data from large corecounts and representing this data to correctly identify sources of poor I/O performance.

# A9: AUTOMATIC PARALLELIZATION

# TOWARDS PARALLELIZING OBJECT-ORIENTED PROGRAMS AUTOMATICALLY

*Welf Löwe, Jonas Lundberg*

Linnaeus University, Sweden

**Abstract**

Today's multi-core processors speed up sequential programs only to a limited extent. Redesigning and re-implementing the huge amount of legacy software is not an option. We describe an approach to parallelize sequential object-oriented general purpose programs automatically by adapting well-known analysis techniques and combine them with context-aware composition. In detail: (1) Parallelizable components are identified applying known results from compiler theory. (2) Aggressively parallelized components are added to but do not replace the original sequential variants. (3) Context-aware composition selects between sequential and generated parallel components dynamically, depending on the execution context. Thus, we do not rely on the static analysis to decide exactly what parts of the program to parallelize. Instead our static analysis only identifies parts that are possible to parallelize. The actual selection of a parallel or sequential variant is postponed to an online decision based on machine learning. In a certain context (e.g., no threads available) the sequential variant is selected, in another context (e.g., many threads available and heavy work ahead) a parallel variant. A first set of experiments demonstrates the potential speed-up. This approach allows sequential programs to benefit from modern hardware developments without bearing unacceptable re-engineering costs.

# HEAP DEPENDENCE ANALYSIS FOR SEQUENTIAL PROGRAMS

*Barnali Basak, Sandeep Dasgupta, Amey Karkare*

IIT Kanpur, India

**Abstract**

In this paper we demonstrate a novel technique for detecting heap dependences in sequential programs that uses recursive data structures. The novelty of our technique lies in the way we compute, for each statement, abstract heap access paths that approximate the locations accessed by the statement, and the way we convert these paths into equations that can be solved using traditional tests, e.g. GCD test, Banerjee test and Lamport test. The dependence test also uses a field sensitive shape analysis to detect dependences among heap locations arising out of sharing within the data structure. In presence of loops, the technique can be used to discover loop dependences. i.e. the dependence among two different iterations of the same loop. This information can be used by a parallelizing compiler to transform sequential input program for better parallel execution.

# B9: PERFORMANCE MODELING AND ANALYSIS II

# TOOLS FOR ANALYZING THE BEHAVIOR AND PERFORMANCE OF PARALLEL APPLICATIONS

*Frederik Vandeputte*

Alcatel-Lucent, Belgium

**Abstract**

Designing efficient scalable parallel applications is hard. As a result, proper tools for analyzing the behavior and performance of parallel applications are valuable assets for the application developer during the entire development cycle. To assist the application developer or researcher in analyzing the behavior of a parallel application, we have developed two analysis tools, called MICA-MT and Perfex-MT. MICA-MT is a program behavior analysis tool that captures and visualizes micro-architectural independent program execution characteristics of multithreaded applications. Perfex-MT is a performance analysis tool that captures a programmable set of performance statistics for each thread using hardware performance counters, both interval-based as well as full execution-based. Both tools are fully automated and provide many detailed statistics concerning the performance and behavior of a multi-threaded shared memory application. By linking all results back to the original source code, bottlenecks can easily be tracked and eliminated during application development. The strength of the two tools lies in combining the results from both tools, providing a complete picture of the characteristics and potential bottlenecks of the application.

# BENCHMARKS BASED ON ANTI-PARALLEL PATTERNS FOR THE EVALUATION OF GPUS

*Jan G. Cornelis, Jan Lemeire*

Vrije Universiteit Brussel, Belgium

**Abstract**

We put forward ``anti-parallel patterns'' to guide the parallel performance analysis process. Anti-parallel patterns or APPs are common parts of parallel programs that cause these programs to have less than ideal performance, where the ideal speedup equals the number of processors. We present benchmarks to model the behavior of APPs on parallel platforms. Each benchmark contains only one APP and is configurable to mimic all its instances. We show how benchmarks can be used to qualitatively and quantitatively compare parallel hardware. Experiments with NVIDIA and AMD GPUs reveal their differences.

# C9: EXASCALE MINI-SYMPOSIUM III

# PARALLELIZING DENSE MATRIX FACTORIZATIONS ON CLUSTERS OF MULTICORE PROCESSORS USING SMPSS

*Rosa M. Badia, Jesus Labarta, Vladimir Marjanovic, Alberto F. Martin, Rafael Mayo, Enrique S. Quintana-Orti, Ruyman Reyes*

Universidad Jaume I, Spain

**Abstract**

We investigate the use of the SMPSs programming model to leverage task parallelism in the execution of message-passing numerical libraries for dense matrix factorizations on clusters equipped with multicore processors. Our experience shows that the major difficulties to adapt the codes in legacy libraries, as e.g. those in ScaLAPACK, to the MPI/SMPSs instance of this programming model are due to the usage of the conventional column-major layout of matrices in numerical libraries. On the other hand, the experimental results show a considerable increase in the performance and scalability of our solution when compared with the standard options based on the use of a pure MPI approach or a hybrid one that combines MPI/multi-threaded BLAS.

# A10: MASSIVE PARALLELISM II

# A PROGRAM EXECUTION MODEL FOR MASSIVELY PARALLEL COMPUTING

*Jack B. Dennis, Guang R. Gao, Xiao X. Meng, Brian Lucas, Joshua Slocum*

MIT Computer Science and Atrificial Intelligence Laboratory, United States of America

**Abstract**

Two major challenges face system designers wishing to achieve progress toward efficient and programmable massively parallel computer systems: the ability to spread a work load over huge numbers of processing cores and a memory model that facilitates seamless data access over the entire memory hierarchy. The Fresh Breeze program execution model (PXM) addresses these challenges using fixed-size chunks of memory to build trees of chunks that represent arbitrary data structures. Chunks are write-once, so computation proceeds by building new data structures instead of modifying those provided as input. This paper extends previous work reporting simulations of the dot product algorithm to matrix multiplication and the fast Fourier transform, algorithms that stress the PXM in different ways. The paper includes new material explaining how mappings of problem data structures to trees of chunks are chosen to expose opportunities for efficient fine-grain concurrency and exploitation of data locality. This work provides further demonstration of the ability of the Fresh Breeze PXM to support distribution of even relatively small computations over large numbers of processors and achieve high processor utilization.

# A MASSIVE DATA PARALLEL COMPUTATIONAL FRAMEWORK ON PETASCALE/EXASCALE HYBRID COMPUTER SYSTEMS

*Marek Blazewicz, Steven R. Brandt, Peter Diener, David M. Koppelman, Krzysztof Kurowski, Frank Löffler, Erik Schnetter, Jian Tao*

Louisiana State University, United States of America

**Abstract**

In this paper we present a massive data parallel computational framework that can be used to develop large scale scientific applications on petascale/exascale hybrid systems. The framework is built upon the highly scalable Cactus computational framework that has been used by scientists and engineers from various fields to solve a wide spectrum of real world scientific problems. Within Cactus, we further explore the performance and scalability of hybrid systems by making use of the computing power of the accelerating devices, in particular GPU's connected to many-core CPU's. As the one of the first non-trivial scientific applications, we successfully developed a new 3D CFD code to demonstrate the capability and improved performance of the framework. We also discuss optimization strategies and scaling behavior of our code which has been tested on a GPGPU cluster. The simulated CFD model is based on the Navier-Stokes equations that are discretized using the finite-difference method and distributed on a rectangular, staggered grid. A homogeneous distribution of computations for the lid-driven cavity problem was used to benchmark the overall performance of the framework and verify the numerical implementation.

# B10: PARALLEL I/O

# A FIRST IMPLEMENTATION OF PARALLEL IO IN CHAPEL FOR BLOCK DATA DISTRIBUTION

*Rafael Larrosa, Rafael Asenjo, Angeles Navarro, Bradford L. Chamberlain*

University of Malaga, Spain

**Abstract**

This paper presents our preliminary implementations of parallel IO routines in Chapel, a high-productivity parallel language for large-scale systems. The IO functions are implemented using standard Chapel features, taking POSIX as the IO middleware layer and Lustre as the target parallel file system. In this work, we focus on the Chapel Block data distribution, for which we propose different algorithms to perform the file read and write operations. We pay particular attention to the collective writing operation, for which we evaluate different strategies. Our algorithms take into account some of the Lustre file system parameters that can have an impact on the write operation performance. Through our evaluation we find that a more careful selection of the stripe pattern (a feature of the Lustre system), as well as the appropriate selection of the number of IO processes that perform the IO calls, should be taken into account to better tune the performance. In addition, we compare our implementations with the ROMIO library, the de facto parallel IO standard, finding that we achieve comparable results with our higher level approach.

# OPTIMIZATIONS FOR TWO-PHASE COLLECTIVE I/O

*Michael Kuhn, Julian Kunkel, Yuichi Tsujita, Hidetaka Muguruma, Thomas Ludwig*

University of Hamburg, Germany

**Abstract**

The performance of parallel distributed file systems suffers from many clients executing a large number of operations in parallel, because the I/O subsystem can be easily overwhelmed by the sheer amount of incoming I/O operations. This, in turn, can slow down the whole distributed system. Many optimizations exist that try to alleviate this problem. Client-side optimizations perform preprocessing to minimize the amount of work the file servers have to do. Server-side optimizations use server-internal knowledge to improve performance. This paper provides an overview of existing client-side optimizations and presents new modifications of the Two-Phase protocol. Interleaved Two-Phase is a modification of ROMIO's Two-Phase protocol, which iterates over the file differently to reduce the number of seek operations on disk. Pipelined Two-Phase uses a pipelined scheme which overlaps I/O and communication phases to utilize the network and I/O subsystems concurrently.

# C10: GPU APPLICATIONS III

# PERFORMANCE MODEL FOR A CELLULAR AUTOMATA IMPLEMENTATION ON A GPU CLUSTER

*Paul Albuquerque, Pierre Künzli, Xavier Meyer*

University of Applied Sciences of Western Switzerland, Switzerland

**Abstract**

The always increasing complexity and size of problems tackled by the industry drives the demand for more computational power. GPUs have gained a lot of popularity as they offer an opportunity to accelerate algorithms having an architecture well-adapted to the parallel GPU model. Cellular Automata fall into this category because of their intrinsic massive parallelism. The heavy computations involved and their parallel nature make them ideal candidates for a GPU implementation. The goal of this paper is to quantify the benefit that stems from parallelizing a cellular automata on a GPU cluster. Towards this end, we propose a performance model for implementing a cellular automata on a GPU and then extend it to the case of a GPU cluster. We validate it by performing a series of measurements. To focus on the methodology, we chose a simple and well-known cellular automata: the Game of Life. Our GPU implementation is based on the CUDA technology of NVIDIA, and at the cluster level we use MPI to manage inter-GPU communications.

# GPU-BASED IMAGE PROCESSING USE CASES: A HIGH-LEVEL APPROACH

*Volkmar Wieser, Clemens Grelck, Holger Schöner, Peter Haslinger, Karoly Bosa, Bernhard Moser*

Software Competence Center Hagenberg, Austria

**Abstract**

This paper addresses the gap between envisioned hardware-virtualized techniques for GPU programming and a conventional approach from the point of view of an application engineer taking software engineering aspects like maintainability, understandability and productivity, and resulting achieved gain in performance and scalability into account. This gap is discussed on the basis of use cases from the field of image processing, and illustrated by means of performance benchmarks as well as evaluations regarding software engineering productivity.

# A11: CLOUD COMPUTING II

# MAPREDUCE FOR SCIENTIFIC COMPUTING

*Pelle Jakovits, Satish Narayana Srirama, Eero Vainikko*

University of Tartu, Estonia

**Abstract**

Scientific computing deals with large-scale scientific modelling and simulation in different domains like astrophysics, climate research, mechanical engineering and bio-informatics. Execution of large and accurate simulations in these domains require significant computing resources. As such, scientific computing has always been closely connected to High Performance Computing (HPC) and Distributed Systems, utilising the computing resources of supercomputers, computer clusters and grids to perform the large scale calculations needed. Cloud is not different from it's predecessors as a resource for computing infrastructure, but provides even easier access to public resources with the increasing popularity cloud computing and the success of many Infrastructure as a Service (IaaS) cloud providers, who rent out virtual infrastructure as utility. Cloud computing frameworks like MapReduce provide tools for implementing algorithms and can automatically parallelise them, which can greatly simplify the work of researchers. But MapReduce is focused on huge data processing and is less suitable for complex algorithms of scientific computing. The SciCloud project in the University of Tartu studies using frameworks based on the MapReduce model for scientific computing, comparing them to other distributed computing frameworks. This paper outlines the motivation for the designing our own distributed scientific computing framework and provides the initial design of the framework design.

# AN AUTONOMIC MANAGEMENT SYSTEM FOR CHOREOGRAFY-BASED WORKFLOWS ON GRIDS AND CLOUDS

*Giandomenico Spezzano, Giuseppe Papuzzo*

CNR-ICAR, Italy

**Abstract**

Future computing environments will integrate cloud platforms with existing computational infrastructures providing frameworks to combine traditional Grid services with on-demand Cloud services, so that additional resources can be requested during peak loads and released after that. Nowadays, workflows are the preferred means for the composition of services into added value service chains representing functional business processes or complex scientific experiments. This paper describes Sunflower an innovative P2P agent-based framework for configuring, enacting, managing and adapting workflows on hybrid computing infrastructures. Sunflower supports service-level parallelism using a QoS-aware service-level schedule algorithm and flow-level parallelism by partitioning a workflow into sub-flows and running the sub-flows in multiple peers in parallel. To orchestrate Grid and Cloud services, Sunflower uses a bio-inspired autonomic choreography model and integrates the scheduling algorithm with a provisioning component that can dynamically launch virtual machines in a Cloud infrastructure to provide on-demand services in peak-load situations.

# B11: SKELETON PROGRAMMING

# FLEXIBLE RUNTIME SUPPORT FOR EFFICIENT SKELETON PROGRAMMING ON HYBRID SYSTEMS

*Usman Dastgeer, Christoph Kessler, Samuel Thibault*

Linköping University, Sweden

**Abstract**

SkePU is a skeleton programming framework for multicore CPU and multi-GPU systems. StarPU is a runtime system that provides dynamic scheduling and memory management support for heterogeneous, accelerator-based systems. We have implemented support for StarPU as a possible backend for SkePU while keeping the generic SkePU interface intact. The mapping of a SkePU skeleton call to one or more StarPU tasks allows StarPU to exploit independence between different skeleton calls as well as within a single skeleton call. Support for different StarPU features, such as data partitioning and different scheduling policies (e.g. history based performance models) is implemented and discussed in this paper. The integration proved beneficial for both StarPU and SkePU. StarPU got a high level interface to run data-parallel computations on it while SkePU has achieved dynamic scheduling and hybrid parallelism support. Several benchmarks including ODE solver, separable Gaussian blur filter, Successive Over-Relaxation (SOR) and Coulombic potential are implemented. Initial experiments show that we can even achieve super-linear speedups for realistic applications and can observe clear improvements in performance with the simultaneous use of both CPUs and GPU (hybrid execution).

# DATA PARALLEL SKELETONS FOR GPU CLUSTERS AND MULTI-GPU SYSTEMS

*Steffen Ernsting, Herbert Kuchen*

Westfälische Wilhelms-Universität Münster, Germany

**Abstract**

CUDA made general-purpose computing for Graphics Processing Units (GPU) popular. But still GPU programming is error-prone and there are a lot of peculiarities to be considered for developing efficient GPU accelerated applications. Algorithmic skeletons encapsulate typical parallel programming patterns. They have emerged to be an efficient approach to simplifying the development of parallel and distributed applications. In this paper, we present an extension of our skeleton library Muesli in terms of GPU accelerated data parallel skeletons for multi-GPU systems and GPU clusters using CUDA. Besides the computation on the GPU, these skeletons also fully hide data transfer between different GPU devices as well as network transfer between different compute nodes. Experimental results demonstrate competitive performance results for some example applications.

# NETWORK MONITORING ON MULTI CORES WITH ALGORITHMIC SKELETONS

*Marco Danelutto, Luca Deri, Daniele De Sensi*

Univ. of Pisa, Italy

**Abstract**

Monitoring network traffic on 10 Gbit networks requires very efficient tools suitable to exploit modern multi-core computing architectures. Specialized network cards can accelerate packet capture and thus reduce the processing overhead, but are not enough for granting adequate packet analysis performance. This is the reason why most monitoring tools cannot cope with high network speeds. This paper describes the design and implementation of ffProbe, a network traffic monitoring application built on top of FastFlow, a multi core programming framework providing the programmer with non-blocking and lock-free synchronization mechanisms, as well as several optimized parallel programming patterns. During the validation phase ffProbe has been compared with two popular network monitoring probes, and it has been demonstrated that it can scale significantly better with the number of cores. As a consequence, ffProbe may be used to to monitor 10 Gbit networks using a commodity server.

# C11: GPU APPLICATIONS IV

# EGOMOTION COMPENSATION AND MOVING OBJECTS DETECTION ALGORITHM ON GPU

*Juan Gómez Luna, Holger Endt, Walter Stechele, José María González Linares, José Ignacio Benavides Benítez, Nicolás Guil Mata*

University of Córdoba, Spain

**Abstract**

Moving objects detection algorithms need real-time processing for diverse applications such as rescue robots and driving assistance. Moreover, in these applications, they should deal with strong levels of egomotion, which limit the reliability of most of the existing techniques. These requirements make them very compute intensive and memory bound, what enforces the use of hardware acceleration, such as FPGA or GPU. In this work, a GPU implementation of an optical flow based moving objects detection algorithm is presented. This algorithm is applicable in scenarios with weak and strong egomotion, thanks to egomotion compensation and two alternative detection methods. Our implementation includes novel approaches on GPU to widely-used techniques as RANSAC and region growing. It also solves image processing parallelization problems, as divergent execution paths, by using compaction and sorting primitives, with a significant impact on performance. Finally, our implementation has been compared to a previous FPGA implementation. From the performance point of view, results on the newest GPUs clearly outperform the FPGA.

# AUTHOR INDEX