# Abstracts

## Rosa Badia, BSC

**Exploiting Multicore Processors and GPUs with OpenMP and OpenCL**

Abstract: With the appearance of multicore processors, considering between them GPUs and other accelerator devices, multiple programming models have been appearing. However, it is important to be able to propose approaches that enable a seamless integration of the applications with the different devices, without a need for rewriting applications each time.  We present OpenMPT, a programming model based on OpenMP and StarSs, and that can also incorporate the use of OpenCL or CUDA kernels.

From OpenMP we obtain high expressiveness to exploit parallelism using tasks. StarSs extensions allow runtime dependence analysis between tasks, and data transfers. And OpenCL and CUDA allow the programmer to easily write efficient and portable SIMD kernels to be exploited inside the tasks. We evaluate the proposal on three different architectures, SMP, Cell/B.E. and GPUs, showing the wide usefulness of the approach. The evaluation is done with four different benchmarks, Matrix Multiply, BlackScholes, Perlin Noise, and Julia Set. We compare the results obtained with the execution of the same benchmarks writtenin OpenCL, in the same architectures. The results show that OpenMPT greatly outperforms the OpenCL environment. It is more flexible to exploit multiple accelerators. And due to the simplicity of the annotations, it increases programmer’s productivity.

## Kirk Cameron, Virginia Tech

**Implications of SpecPower and the Green500 for HPC**

The SPECpower benchmark and the Green500 list make it possible to track energy-efficiency gains in both servers and supercomputers. In this talk we discuss what we have learned in the three years since first release of these two energy efficiency lists including trends and implications for HPC systems and applications.

## Ewa Deelman, ISI

**Scientific Workflows and Cloud Computing**

This talk will explore the potential benefits of cloud computing for scientific workflow applications. It will present the performance and cost of clouds from the perspective of three different scientific workflow applications.  We will show experiments of running these workflows on a commercial cloud and on a typical HPC system, and we analyze the various costs associated with running those workflows in the cloud.  Particular emphasis will be given to issues of data management in cloud environments. In grids and clusters, workflow data is often stored on network and parallel file systems. In this talk we investigate some of the ways in which data can be managed for workflows in the cloud. We discuss various storage and file systems options and analyze the resulting performance and cost of the workflows.

## Emmanuel Jeannot, LABRI

**Near-Optimal Placement of MPI processes on Hierarchical NUMA Architectures**

MPI process placement can play a deterministic role concerning the application performance. This is especially true with nowadays architecture (heterogenous, multicore with different level of caches, etc.). In this paper, we will describe a novel algorithm called TreeMatch that maps processes to resources in order to reduce the communication cost of the whole application. We have implemented this algorithm and will discuss its performance using simulation and on the NAS benchmarks.

## Thilo Kielmann, Vrije Universiteit

**Bag-of-Tasks Scheduling under Time and Budget Constraints**

Commercial cloud offerings, such as Amazon’s EC2, let users allocate compute resources on demand, charging based on reserved time intervals. While this gives great flexibility to elastic applications, users lack guidance for choosing between multiple offerings, in order to complete their computations within given time and budget constraints. In this talk, we present BaTS, our budget and time-constrained scheduler. BaTS can schedule large bags of tasks onto multiple clouds with different CPU performance and cost, minimizing completion time while respecting an upper bound fore budget to be spent. BaTS requires no a-priori information about task completion times, and learns task completion time distributions at run time.

## Alexey Lastovetsky, University College Dublin

**Design and implementation of parallel algorithms for highly heterogeneous HPC platforms**

The current trend in HPC platforms is that they employ increasingly heterogeneous processing units, the relative speeds of which cannot be accurately represented by positive constants independent on the problem size. Therefore, traditional heterogeneous parallel algorithms based on the constant performance models (CPMs) become less and less applicable on these platforms. In this talk, we introduce the functional performance model (FPM) and FPM-based partitioning algorithms proposed to address this issue. The FPM represents the speed of a processing unit by a function of problem size. We show that FPM-based algorithms significantly outperform their CPM-based counterparts on highly heterogeneous platforms given the full FPM of the processing units is provided as an input parameter. While useful in many applications, these algorithms cannot be directly used in self-adaptable applications due to a very high cost of construction of the full FPMs. For self-adaptable applications, we propose a special class of FPM-based algorithms that do not require the FPM as input parameter. Instead, they build a partial estimate of the FPM during their execution and use this estimate for sub-optimal distribution of computations. We present results of experiments with self-adaptable FPM-based applications on various heterogeneous platforms and compare their performance with traditional CPM-based applications

## Christine Morin, INRIA Rennes

**XtreemOS European Project: Achievements and Perspectives**

XtreemOS European project has been funded by the European Commission during the last four years. As part of this project, we have designed and implemented a Grid distributed operating system, called XtreemOS.

The XtreemOS operating system provides for Grids what a traditional operating system offers for a single computer: abstraction from the hardware and secure resource sharing between different users. When a user runs an application on XtreemOS, the operating system automatically finds all resources necessary for the execution, configures user’s credentials on the selected resources and starts the application.  It simplifies the work of users by giving them the illusion of using a traditional computer. XtreemOS is a Linux based open source software available for PC, clusters and mobile devices.

In this talk, we will present some innovative XtreemOS features, focusing mainly on the XtreemGCP service implementing fault tolerance for Grid applications. While originally designed as a Grid system, XtreemOS constitutes a sound basis to manage cloud federations. We  will then describe our research directions in the area of cloud computing that we will be further investigated in the framework of the Contrail European project starting in October 2010.

## Christian Obrecht, Centre de Thermique de Lyon

**The TheLMA Project: Multi-GPU Implementation of the Lattice Boltzmann Method**

The lattice Boltzmann method (LBM) is a novel approach in computational fluid dynamics, with numerous convenient features from a numerical and physical standpoint, such as explicitness or ability to deal with complex geometries. During the last decade, it has proven to be an effective alternative to the solving of the Navier-Stokes equations. Not the least, the LBM is interesting from the HPC point of view because of its inherent parallelism.

Several successful attempts of implementing the LBM on CUDA enabled GPUs were recently reported; nevertheless these works are of moderate practical impact since memory in existing computing devices is too small for large scale simulations. Multi-GPU implementations of the LBM are still at an early stage of development. In this contribution, we shall present a pthread based multi-GPU LBM solver developed using our TheLMA framework. This program shows excellent performance and scalability. We shall also discuss tiling and communication issues for present and forthcoming implementations.

## Manish Parashar, Rutgers U

**Exploring the Role of Clouds in Computational Science and Engineering**

Clouds provide on-demand access to computing utilities, an abstraction of unlimited computing resources, and a usage-based payment model, and are rapidly joining high-performance computing system and Grids as viable platforms for scientific exploration and discovery. In fact, production computational infrastructures are already integrating these paradigms. As a result, understanding application characteristics and usage modes that are meaningful in such a hybrid infrastructure, and how application workflows can effectively utilize it, is critical. In this talk, I will explore the role of clouds (along with traditional HPC) in science and engineering. I will also describe how science and engineering applications with non-trivial requirements can benefit from clouds. This talk is based on research that is part of the CometCloud project at the Center for Autonomic Computing at Rutgers.

## Jim Plank, U of Tennessee

**Storage as a First Class Citizen in HPC Environments**

We understand RAM.  We understand disks.  We kind of understand SSD's.  And we're really good at combining computational elements to get flashy performance numbers. However, we have no idea how storage really fits into high performance computing environments.  After 20+ years of working in the areas of fault-tolerance, storage and computation, the speaker will give his perspective on why storage is a second class citizen in HPC environments, and why that should change.

## Padma Raghavan, Pennsylvania State University

**Optimizing Scientific Software for Multicores**

Increasing core counts and decreasing feature sizes of current and future processors  pose several challenges for the  energy-aware tuning  of scientific codes for high performance and reliability. We  present some recent work on the modeling of cache and memory subsystem performance of scientific codes  and the impact of soft errors on the performance on linear algebra  kernels. We conclude with an overview of some energy-aware approaches to ensure reliability and high performance for such scientific kernels.

## Dan Reed

**Technical Clouds: Seeding Discovery**

We are now building cloud data centers and infrastructure far bigger than anything previously contemplated in high-performance computing. These clouds contain hundreds of thousands of cores and many petabytes of data, and they are superficially similar to technical computing systems. However, cloud architectures and software differ from those used for technical computing in some important ways.

What are the software structures and capabilities that best exploit cloud capabilities and economics while providing application compatibility and community continuity? How do we best balance ease of use and performance for research computing? What are the appropriate roles of public clouds relative to local computing systems, private clouds and grids? How can we best exploit cloud elasticity for peak demand? In a world where massive amounts of experimental and computational data are produced daily, how do we best extract insights from this data, both within and across disciplines, via clouds?

This talk will examine these and other issues from a technical perspective.

## Yves Robert, ENS Lyon and Institut Universitaire de France

## (Joint work with Franck Cappello and Henri Casanova)

**Checkpointing vs. Migration for Post-Petascale Supercomputers**

An alternative to classical fault-tolerant approaches for large-scale clusters is failure avoidance, by which the occurrence of a fault is predicted and a preventive measure is taken. We develop analytical performance models for two types of preventive measures: preventive checkpointing and preventive migration. We also develop an analytical model of the performance of a standard periodic checkpoint fault-tolerant approach. We instantiate these models for platform scenarios representative of current and future technology trends. We find that preventive migration is the better approach in the short term by orders of magnitude. However, in the longer term, both approaches have comparable merit with a marginal advantage for preventive checkpointing. We also find that standard non-prediction-based fault tolerance achieves poor scaling when compared to prediction-based failure avoidance, thereby demonstrating the importance of failure prediction capabilities. Finally, our results show that achieving good utilization in truly large-scale machines (e.g., $2^{20}$ nodes) for parallel workloads will require more than the failure avoidance techniques evaluated in this work.

## Rajeev Thakur, Argonne National Laboratory

**Future Directions in MPI**

This talk will discuss issues that need to be addressed in the MPI standard as well as in MPI implementations to enable MPI to scale well to exascale. I will present some cases where we already encountered (and fixed) some of these issues on Argonne's leadership-class IBM Blue Gene/P. I will also give an update on recent activities of the MPI Forum and what features are being considered for inclusion in MPI-3.

## Rich Vuduc, GATech

**Should I port my code to a GPU?**

In this talk, I make the perhaps provocative and surprising claim that a GPU is no more powerful than a CPU, even in cases where one might have expected a GPU to win big.

In particular, I summarize our recent experience in analyzing and tuning both multithreaded CPU and GPU implementations of three computations in scientific computing. These computations—(a) iterative sparse linear solvers; (b) sparse Cholesky factorization; and (c) the fast multipole method—exhibit complex behavior and vary in computational intensity and memory reference irregularity. In each case, algorithmic analysis and prior work might lead us to conclude that an idealized GPU can deliver significantly better performance, but we find that for at least equal-effort CPU tuning and consideration of realistic workloads and calling-contexts, we can with two modern quad-core CPU sockets roughly match one or two GPUs in performance.

These conclusions are not intended to dampen interest in GPU acceleration; on the contrary, they should do the opposite: they partially illuminate the boundary between CPU and GPU performance, and ask architects to consider application contexts in the design of future coupled on-die CPU/GPU processors.

## Hans P. Zima, Jet Propulsion Laboratory, California Institute of Technology and University of Vienna, Austria

**Enhancing the Dependability of Extreme-Scale Applications**

Emerging massively parallel extreme-scale systems will be providing the superior computational capability required for dramatic advances in fields such as DNA analysis, drug design, climate modelling, and astrophysics. These systems will be composed of devices less reliable than those used today, and faults will become the norm, not the exception. This will pose significant problems for users, who for half a century have enjoyed an execution model that largely relied on correct behavior by the underlying computing system. In this talk we outline an approach that will enhance the dependability of extreme-scale applications. Such applications and the systems they run on must be introspective and adaptive, actively searching for errors in their program state with hardware mechanisms and new software techniques. An introspection framework, which is built around an inference engine and an associated knowledge base, will monitor the execution of the system, analyze errors and subsystem failures, and provide feedback to both the application and the operating system. Domain knowledge will be leveraged in the form of user input, knowledge about applicable fault tolerance mechanisms, and compiler analysis as a way to reduce the need for recovery mechanisms. The result is a self-aware system that can deal with errors in a sophisticated way, ignoring them if possible, attempting corrections, or reverting to an earlier checkpoint when necessary.