Monte Carlo (DSMC) method. In this method the real gas is approximated by a system of numerical particles that move according to the laws of mechanics and collide according to stochastic rules based on the intermolecular potential. This method has been successfully applied in aerospace engineering. A challenge is to extend the method to applications, e.g., low Mach number flows, where the signal-to-noise ratio is small. Some recent results related to weighted particles will be presented.
A second application area of stochastic particle methods is chemical engineering, e.g., soot modelling, production of nanoparticles, granulate processing. The governing equations are various generalizations of Smoluchowski's coagulation equation, including fragmentation, growth, inception, and other processes. An application related to a five-dimensional granulation model will be considered. A particular challenge is the strongly fluctuating number of numerical particles in the direct simulation method. Alternatives are provided by methods using weighted particles.

Agents in Grid System – Design and Implementation

K. Wasielewska, M. Drozdowicz, P. Szmeja, M. Ganzha,
M. Paprzycki, I. Lirkov, D. Petcu, C. Badica

We are in the process of developing an agent-based intelligent middleware for the Grid. Our approach is based on utilizing agent teams as resource brokers and managers. We also plan to introduce autonomous SLA negotiations. Our earlier work resulted in a prototype implementation. However, our recent research led to a complete redesign of the system. The aim of this note is to present the new design, as well as outline the new approach to its implementation.

Distributed Heterogenous High-Performance Computing with ViennaCL

J. Weinbub, K. Rupp, S. Selberherr

The Vienna Computing Library (ViennaCL) provides standard data types for linear algebra operations on GPUs and multi-core CPUs. The library is based on OpenCL, from which ViennaCL inherited the unified parallel programming approach for personal computers, servers, handhelds, and embedded devices. The ViennaCL represents a drop-in replacement for the Boost uBlas library, which is a generic template class library with BLAS level 1, 2, 3 support. Therefore, existing uBlas implementations can access, with a minimum of effort, the vast computing resources of, for instance, GPUs.
Typically, OpenCL enabled applications or libraries use a single host, for example, one personal computer with an OpenCL enabled graphics adapter. To utilize the common

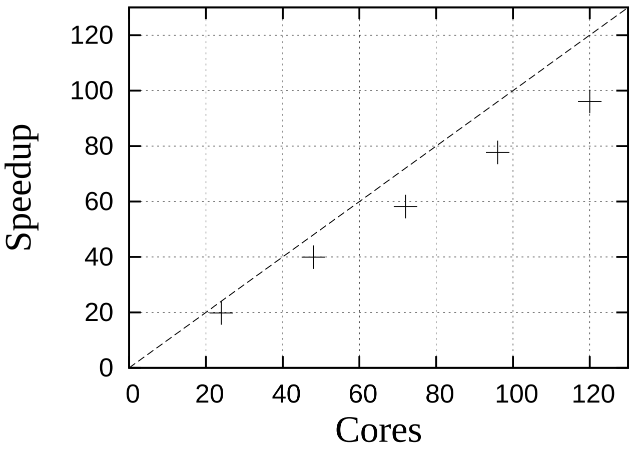

Figure 1: Scalability of the speedup.

OpenCL API on machines with different hardware, we developed a distributed computing layer in ViennaCL based on the Message Passing Interface (MPI). In general, the data is partitioned and distributed to various nodes by an MPI communication layer. Each node is accessed using the ViennaCL which encapsulates the underlying processing hardware. ViennaCL can be utilized in a heterogenous environment consisting of GPUs from AMD and/or NVIDIA and CPUs from AMD and/or INTEL, due to the ATI Stream SDK and OpenCL enabled NVIDIA drivers. This work presents the current state of development of this approach. The discussion is based on basic linear algebra computations, being the dense matrix-vector product and the dense matrix-matrix product, both of various dimensions. Performance analysis and comparisons with established linear algebra packages are given. A special focus is laid on the heterogeneity of the presented approach. Therefore, the investigations target two different computing environments to analyze the capabilities of this distributed computing approach. One environment is a MPI cluster environment consisting of AMD Opteron nodes, where each node offers 24 cores. The nodes are connected by a high-performance

DDR Infiniband network. The scalability of the implementations is investigated for up to 120 cores. The second computing environment consists of personal computers equipped with GPUs of different vendors and types.

The investigated GPUs range from mid-range graphics adapters, like the ATI Radeon 5850, to high-end products, like the NVIDIA Tesla C2050. The communication overhead of the approach has been analyzed as well as approaches to partition the data for graphics adapters. This is of particular relevance, as the memory limitations are among the major drawbacks of high-performance computing on graphics boards. Furthermore, the price of the computing environments and their provided performance are discussed. Figure 1 presents some results of the scalability of the speedup of the dense matrix-matrix multiplication for dimensions 16000×16000 recorded on the AMD Opteron Cluster. The speedups are shown for different core sizes(crosses) and related to an ideal, extrapolated speedup of a reference uBlas implementation(line) on a single core.

Robust Preconditioners for General SPD Operators

J. Willems

An abstract setting for robustly preconditioning symmetric positive definite (SPD) operators is presented. The method belongs to the class of additive Schwarz preconditioners, and it requires only rather mild assumptions naturally satisfied by operators resulting from the discretization of several important partial differential equations. The term "robust" refers to the property of the condition numbers of the preconditioned systems being independent of mesh parameters and problem parameters. Important instances of such problem parameters are in particular (highly varying) coefficients. The core of this method is the construction of the coarse space based on the solution of local generalized eigenvalue problems. Several numerical examples are presented to illustrate the properties of the method, and some possible extensions are addressed.

Using BlueGene/P and GPUs to Accelerate Computations in the EULAG Model

R. Wyrzykowski, K. Rojek, Ł. Szustak

EULAG (Eulerian/semi-Lagrangian fluid solver) is an established computational model developed by the group headed by Prof. Piotr K. Smolarkiewicz for simulating thermo-fluid flows across a wide range of scales and physical scenarios. The EULAG model is an ideal tool to perform numerical experiments in a virtual laboratory with time-dependent adaptive meshes and within complex, and even time-dependent model geometries.

This paper presents perspectives of the EULAG parallelization based on the MPI, OpenMP and OpenCL standards. We focused on development of the kernel, which consist of the most time-consuming calculations of the EULAG model, which are two algorithms: Laplacian algorithm (laplc) and multidimensional positive definite advection transport algorithm (MPDATA).

The first challenge of our work was the parallelization of laplc subroutine using MPI across nodes, and OpenMP within nodes on the BlueGene/P supercomputer located in the Institute for Parallel Processing, Bulgarian Academy of Sciences.

The second challenge of our work was to accelerate computations of the Eulag model using modern GPUs, and to find an optimal load balance between GPU and CPU.