Perspectives of GPU computing in Science

September 26 -28, 2016 - Sapienza Università di Roma

Book of Abstracts

Contents

Invited Talks

Author: Piero Altoè (NVIDIA)
Title: *Recent trends in GPU architectures*

Nearly a decade ago, NVIDIA® pioneered the use of GPUs to accelerate computationally-intensive workloads with the introduction of the G80 GPU and the CUDA® parallel computing platform. Today, Tesla® GPUs accelerate thousands of High Performance Computing (HPC) applications across many areas including computational fluid dynamics, medical research, machine vision, financial modeling, quantum chemistry, energy discovery, and several others.
NVIDIA Tesla GPUs are installed in many of the world's top supercomputers, accelerating discovery and enabling increasingly complex simulations across multiple domains. Datacenters are using GPUs to speed up numerous HPC and Big Data applications, while also enabling leading-edge Artificial Intelligence (AI) and Deep Learning systems.
NVIDIA's new NVIDIA Tesla P100 accelerator using the groundbreaking new NVIDIA® Pascal™ GP100 GPU takes GPU computing to the next level. This presentation details both the Tesla P100 accelerator and the Pascal GP100 GPU architectures.

Author: Michael Bader (Technische Universität München, Institut für Informatik)
Title: *Optimizing the Dynamic Rupture Simulation Software SeisSol for Xeon Phi (Co-)Processors*

The open-source software SeisSol is able to simulate the dynamic rupture process of earthquakes together with seismic wave propagation with high accuracy on complicated geometries. It features high-order discretization in space and time, allowing a compute-bound, highly scalable implementation that can achieve multi-PFLOPS performance on latest supercomputers. This presentation will focus on aspects of optimizing SeisSol for large Xeon Phi platforms, and discuss using off-loading vs. symmetric mode for Xeon Phi co-processor (Knights Corner) platforms, as well as present first results on Knights Landing systems.

Author: Matteo Bauce (Sapienza Università di Roma, INFN Roma)
Title: *GPU integration in High Energy Physics experiment online event selection systems*

The fast development of GPUs for gaming industry in the past decades made quite intriguing their application for scientific purpose: GP-GPU deployment in particle physics is constantly increasing, based on successful results reported by some pioneer experiments. GPUs has been used in several aspects of the offline data analysis of typical High Energy Physics (HEP) experiments: their parallel computation power is suitable for Monte-Carlo simulation, data-reconstruction algorithm execution, multivariate analysis techniques, etc. The significant benefit achieved from GPU deployment in these applications opened the challenge toward a new goal: integration of GPUs in real time environment of particle physics experiments. The online event selection system (trigger) of a HEP experiment imposes strict constraints on latencies, throughput and algorithm approximation. The role of GPU can be relevant to improve the purity of selected sample in dense environment which can be tackled in a parallel way. Experiments with different size and constraints investigated GPU deployment both at the lowest syncronous trigger levels, as an alternative to customized electronics, and to the higher asyncronous levels, based on algorithms traditionally executed on CPU farms. In this contribution will be reported the most recent results on this applications from different experiments. The prospects for the upcoming years will also

be discussed: experiments currently undergoing a design phase are considering the GPU integration in the data acquisition system as a feasible option for a triggerless system.

Author: Claudio Bonati (Università di Pisa and INFN Pisa)
Title: *The computational challenges of Lattice Quantum Chromodynamics*

In this talk I will present the general ideas about the lattice formulation of the theory of strong interactions between elementary particles. In particular I will focus on its computational aspects, discussing the typical numerical problems that are encountered in performing a state of the art simulation and the strategies that have been developed to attack these problems.

Author: Alexis Hérault (Département Ingénierie Mathématique, Laboratoire M2N, Conservatoire des Arts et Métiers, Paris, France; Istituto Nazionale di Geofisica e Vulcanologia, Catania, Italy)
Title: *Implementing the Smoothed Particles Hydrodynamics method on GPUs with CUDA*

We present GPUSPH, a full GPU implementation of the Smoothed Particle Hydrodynamics (SPH) method. GPUSPH is a multi-node multi-GPU SPH code written in C++/CUDA in which all the steps involved in SPH (neighbor search, forces computation and time integration) are done exclusively on GPU. We will specially focus on three main aspects: parallelization strategy, nu- merical precision and code maintainability. A multi-node multi-GPU code has three level of parallelization: the device (GPU) level, the mode level (inter GPU communication on the same node) and the cluster level (inter node communica- tion). Each level has it's specificities that should be taken care of during the code development. We will detail the strategy we developed for each level of paralellization and show how it's possible to hide inter device and inter node communication latency. Even if the GPUs are now fully double precision capable there is still a performance penalty factor of 2 or 3 between single and double precision floating point operation. In order to achieve best performance double precision must be used only when it's absolutely necessary. We will show that, for SPH, a wise use of single precision performs better than a straightforward one of double precision. Performance is important but code maintainability cannot be sacrified for it. With SPH we must deal with a large number of different options and data structures who depend on it. It could lead to poorly readable code with unused variables and dead code. We will show how to overcome these issues and obtain a maintainable code.

Author: Jeffrey Kelling (Helmholtz Zentrum Dresden Rossendorf)
Title: *Pushing the Limits of Lattice Monte-Carlo Simulations using GPUs*

Lattice Monte-Carlo methods are used to study out-of- and towards-equilibrium systems, like surface growth, spin systems and even phase separation in solid mixtures using kinetic Metropolis lattice Monte-Carlo (KLMC). Applications range from the study of universal scaling or aging behaviors to concrete systems, where coarsening of nanocomposites or self-organization of functional nanostructures is relevant, for example spinodal decomposition in solar cell absorber layers. In these systems, scaling needs to be followed for long times to allow structures to grow over orders of magnitude, which requires large-scale simulations. For the evolution of nanostructures, atomistic simulations at experimental spatiotemporal scales are often desired. This talk will give an overview over a variety of lattice Monte-Carlo algorithms, which have been found or made suitable for implementation on GPUs: Stochastic cellular automata can be implemented very efficiently [1-3] and are suitable for many systems. The efficient implementation of random sequential dynamics is more challenging. Solutions will be presented for a dimer lattice gas mapped to surface growth [4,5] and KLMC [6]. The latter was also extended to implement dynamics driven by ion-beam mixing triggering long-range interactions. However, these implementations hinge on the fact, that only a very small number of states need to be encoded at each lattice site. A more flexible implementation, employing a variation of multisurface-coding to enable vectorization, will be presented

for simulations of restricted solid-on-solid and Potts models with random sequential dynamics. [7] [1] Block, B., Virnau, P., Preis, T.: Comp. Phys. Comm. 181(9), 1549 (2010)
[2] Lulli, M., Bernaschi, M., Parisi, G.: Comp. Phys. Comm. 196, 290 (2015)
[3] Kelling, J., Ódor, G., Gemming, S.: 2016 IEEE Int. Conf. Intell. Eng. Syst., arXiv:1606.00310 (2016)
[4] Kelling, J., Ódor, G.: Phys. Rev. E 84, 061150 (2011)
[5] Ódor, G., Kelling, J., Gemming, S.: Phys. Rev. E 89, 032146 (2014)
[6] Kelling, J., Ódor, G., Nagy, M. F., Schulz, H., Heinig, K.: EPJST 210, 175 (2012)
[7] Kelling, J., Ódor, G., Gemming, S.: arXiv:1605.02620 (2016)

Author: Ivan Kisel (Goethe University of Frankfurt am Main, Frankfurt Institute for Advanced Studies, FIAS, GSI Helmholtz Center for Heavy Ion Research Darmstadt)
Title: *Heavy-ion physics on many-core computer architectures*

Modern and future heavy-ion experiments are focused on measurements of very rare particles at interaction rates up to 10 MHz with data flow of up to 1 TB/s, that cannot be fully stored on currently available storage devices. The data flow should be reduced by selecting collisions with potentially interesting physics. Therefore, full reconstruction of the collision topology including reconstruction of short-lived particles is required already in real time with the experiment. The algorithms for online reconstruction should be fast, vectorized, parallelized, and portable in order to utilize the full potential of different many-core CPU/GPU computer architectures. One of the most promising approaches to the search of particle trajectories (tracks) in the detector system is the Cellular Automaton (CA) track finder. The algorithm is based on consistent accumulation of the tracking information by building of short track segments, linking them according to the track model and selecting the best track candidates. After tracks are found, they parameters are properly estimated within the Kalman Filter (KF) approach. Next, short-lived particles are reconstructed covering signals from all physics cases: strange particles, strange resonances, hypernuclei, low mass vector mesons, charmonium, and open-charm particles. All discussed algorithms are fully vectorized and parallelized and show a strong linear scalability on many-core architectures.

Author: Dimitri Komatitsch, Vadim Monteiller, Bence Solymosi, Paul Cristini and Nathalie Favretto-Cristini (CNRS, Laboratory of Mechanics and Acoustics, Marseille, FR)
Title: *Using large GPU clusters and the race towards exaflops to improve high-resolution acoustic imaging*

We implement a high-order finite-element application, which performs the numerical simulation of seismic wave propagation resulting for instance from earthquakes at the scale of a continent or from active seismic acquisition experiments in the oil industry, on a large cluster of GPU cards using CUDA and non-blocking message passing based on MPI. We discuss the implementation and optimization of the code and compare it to an existing very optimized implementation in Fortran and MPI on a classical cluster of CPU nodes. We perform a number of numerical tests to validate the single-precision CUDA and MPI implementation and assess its accuracy. We then analyze performance measurements and depending on how the problem is mapped to the reference CPU cluster, we obtain a significant speedup. More generally speaking, we illustrate how the field of acoustic wave propagation modeling as well as related acoustic tomographic imaging can fully benefit from GPU computing, in particular on large machines. For the largest earthquake and seismic tomography simulations that we present, the calculations are performed on the Cray XK7 named *Titan*, a computer with 18,688 GPU accelerators housed at Oak Ridge National Laboratory (USA).

Author: Luca Leuzzi (NANOTEC, Institute of Nanotechnology CNR, Italy)
Title: *Statistical mechanical approaches on GPU to systems with continuous spin variables on random graphs with quenched disordered interactions.*

Statistical mechanical models with angular and phasor variables have been recently shown to reproduce and foresee fundamental features of stationary systems of non-linear waves, including lasing. This statistical approach is most useful in presence of randomness and frustration where standard photonic techniques become too complicated. Monte Carlo simulations and numerical resolution of analytic equations for these continuous spin models on random graphs can profit much from the use of GPU. We will present the theoretical framework, the algorithms used and a few examples of GPU applications and comparisons to CPU performances.

Author: Filippo Mantovani (Barcelona Supercomputing Center)
Title: *Scientific computing on ARM-based platforms: evaluation and perspectives*

Since 2011 the EU Mont-Blanc project pushes the development of ARM-based compute platforms following the vision of leveraging the fast growing market of mobile technology for performing scientific computation. The process started almost 5 years ago with the development of prototypes based on Android dev-kits is now evolving beyond the research project, towards commercial computational platforms based not only on mobile SoCs, but also on server and HPC technology. In this talk I will introduce the experience gained developing prototypes based on ARM technology within the Mont-Blanc project. Special attention will be given to the Mont-Blanc prototype and the last ARM 64 bit platforms appearing on the market. The goal of the talk is to give a panoramic view of ARM based scientific computing, supported by experience, lesson learned and test results.

Author: Joachim Stadel (University of Zurich)
Title: *A Two Trillion Particle Dark Matter Simulation and a Perspective on GPUs in Astrophysics*

The need to very precisely "nail down" the non-linear theory of dark matter clustering has become central to the success of the EUCLID satellite mission to launch in 2020. This need has led to the requirement of both very good mass resolution as well as very large simulation volumes for statistical accuracy. I describe the recently completed two trillion particle simulation on the Piz Daint supercomputer in Lugano, Switzerland and how GPUs and the fast multipole method (FMM) were used to make this possible. This simulation represents the finest calculation of its type and I will describe how its data will be used in the context of the EUCLID mission. The code used was PKDGRAV3 which tries to use all resources in the system, the CPUs, GPUs, memory, HPC network and disk to maximum efficiency. There will always be one component of the system which limits the calculation in some way and much of the development effort and the design of such a simulation campaign are involved with addressing such limits. I will discuss these issues and take a more critical look at the future of GPU computing in this context. There is now a huge breadth of usage of GPU computing in astrophysical simulations and the GPU's role in these applications can be quite different. To contrast the large cosmological simulations, I take a look at planet formation simulations using the GENGA code where the incredibly huge number of time-steps required in the calculations forces a rather different perspective on GPU computing.

Author: Ana Varbanescu (IVI, University of Amsterdam)
Title: *Heterogeneous Computing: the CPU can also help!*

Heterogeneous systems combining GPUs and CPUs are becoming mainstream. The performance potential of these processors working together is significant, yet most applications choose to use such systems as homogeneous, exploiting either GPUs or CPUs exclusively. This choice is due to the perceived imbalance between the high difficulty of deploying heterogeneous applications and the unknown performance gain. In this talk, we will demonstrate that implementing and deploying heterogeneous applications is simpler than expected, is supported by easy-to-use tools, and leads to significant performance gain for many applications.

Author: Yifeng Cui (San Diego Supercomputer Center, University of California, San Diego, USA)
Title: *Regional Scale Earthquake Simulations on OLCF Titan and NCSA Blue Waters*

Three-dimensional numerical simulations of dynamic rupture and wave propagation allow deterministic predictions of near-source ground motions during large, rare earthquakes, which are not well represented in observed data. High Performance GeoComputing Lab (HPGeoC) at San Diego Supercomputer Center has developed a petascale earthquake simulation code that is capable of fully exploiting the computational capability of several heterogeneous systems powered by NVIDIA GPUs. The technical developments are integrated to support for innovative computational research coordinated by Southern California Earthquake Center (SCEC), one of the largest open research collaborations in geoscience. The CUDA-based AWP-ODC code has demonstrated ideal scalability up to 16K GPUs, now publically available via GitHub. The code was used to carry out regional scale ground motion simulations, such as a realistic 0-10 Hz ground motion synthetics along a rough fault embedded in a 3D velocity structure with small-scale heterogeneities described by a statistical model, and recent 0-4 Hz nonlinear simulations of a M7.7 earthquake scenario on the southern San Andreas fault. Using the modeling capability, we have been able to improve the speedup by a factor of 110 in a key strain tensor calculation critical to probabilistic seismic hazard analysis, and generated physics-based seismic hazard maps of the Los Angeles region at frequency as high as 1-Hz.

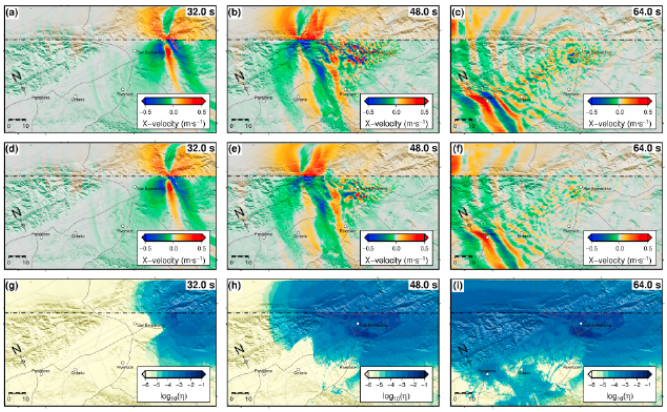


Figure (color online): Snapshots from the 4 Hz San Andreas simulation inside the blue rectangle in (a-c) and (d-f) show fault-parallel velocity for the linear and nonlinear cases, respectively, and (g-i) depict the evolution of permanent plastic strain at the surface obtained from the nonlinear simulation. The dashed line shows the fault trace (Roten et al., SC'16).

Adding advanced physics helps to produce more realistic seismograms at high frequencies, and to better understand directivity and focusing effects in long-period surface waves, which may represent a major hazard to high-rise structures during a future large earthquake. The computational cost of such models, however, has always imposed a limit on the maximum resolvable frequency. The computational challenges due to memory and scalability issues can result in costly misguidance for structural design in earthquake-prone regions. In collaboration with SCEC researchers at San Diego State University, we

have implemented nonlinearity using a Drucker-Prager yield condition in the finite difference code. The optimized CUDA kernels help efficiently utilize the GPU's memory bandwidth, resulting in a significant increase in the model region and accuracy for state-of-the-art earthquake simulations in a realistic earth structure. The performance improvements allowed us to efficiently use the hybrid many-core systems including DOE OLCF Titan at OakDOE OLCF Titan at Oak Ridge National Laboratory, and NSF Blue Waters at National Center for Super-computing Applications. The future extreme-scale systems are expected to exhibit a substantially different balance among processor speed, system interconnect and memory bandwidth. Earthquake applications will need to be radically re-designed and re-implemented for new programming models, new algorithms, and system tools to ensure a continued growth in sustained performance and productivity. This presentation will conclude with a discussion on how the seismology community can prepare ready for this sea changes in architectures towards Exascale computing.

Contributed Talks

Author: Akitaka Ariga (Albert Einstein Center for Fundamental Physics, Laboratory for High Energy Physics, University of Bern, Switzerland)
Title: *High speed 3D track reconstruction in particle physics with GPUs*

High speed 3D tack reconstruction is a key procedure in particle physics to discover new particles and to study interactions between them. In particular, nuclear emulsion detectors, which are widely used in particle physics (e.g. neutrino, antimatter experiments) and have the best position resolution among all particle detectors, require an enormous computation to process the data taken by microscopes. With advances in data acquisition technology, the data throughput is of the order of 200 TB /microscope /day or 1 PB /facility /day. Such a huge data rate should be analyzed in a real-time manner to reconstruct 3D trajectories of particles, thus the GPUs play an essential role. In this talk, a novel implementation of the GPU computing for the analysis of particle physics data from the nuclear emulsion detectors will be presented. The results show that a two orders of magnitude faster processing has been achieved with GPUs respect to a single CPU computing. The algorithm of 3D track reconstruction is general and can be applied for any other particle detectors with 3D data to be processed in a real-time. The system is being used by several physics experiments and also some other fields of research such as a geological study of Swiss alpine glaciers by means of cosmic-ray muons with particle detectors (Muon radiography).

Author: A. Augusto Alves Jr. and Michael Sokoloff (University of Cincinnati)
Title: *MCBooster and Hydra: two libraries for high performance computing and data analysis in massively parallel platforms*

MCBooster is a templatized header-only, C++11-compliant library for the generation of large samples of phase-space Monte Carlo events on massively parallel platforms. It was released on GitHub in the spring of 2016. The library core algorithms implement the Raubold-Lynch method; they are able to generate the full kinematics of decays with up to nine particles in the final state. The library supports the generation of sequential decays as well as the parallel evaluation of arbitrary functions over the generated events. The output of MCBooster completely matches that of popular and well-tested software packages such as GENBOD (W515 from CERNLIB) and TGenPhaseSpace from the ROOT framework. Hydra is a templatized header-only, C++11-compliant library for the generation of Monte Carlo, function evaluation, data fitting, multidimensional adaptive numerical integration and histograming on massively parallel platforms. Hydra is under intense development and most of its core features are already implemented and tested. Hydra deploys a series of techniques in order to achieve the optimal performance in both computing and management of memory resources. The overall design exploits heavily C++ variadic templates in order to implement static polymorphism, kernel fusion and coalesced memory access patterns, avoiding completely the usage of function pointers, virtual methods calls and other known potential performance degrading constructs. MCBooster and Hydra are developed on top of the Thrust library and run on Linux systems. Both deploy transparently on NVidia CUDA-enabled GPUs as well as multicore CPUs. This contribution summarizes the main features of MCBooster and Hydra. A basic description of the user interface and some examples of applications are provided, along with measurements of performance in a variety of environments.

Author: Julien Bernard , Damien Gratadour, Arnaud Sevin, Maxime Lainé, Denis Perret, Florian Ferreira and Nicolas Douceta (LESIA, Obs. Paris, CNRS, UPMC, Univ. Paris Diderot)
Title: *A GPU based RTC for E-ELT Adaptive optics: RTC prototype*

In ground-based astronomy, atmospheric turbulences create optical aberrations and have a huge impact on the image resolution. Adaptive Optics (AO) systems are developed to correct those aberrations by controlling a deformable mirror (DM) in real-time. A wavefront sensor (WFS) measures those aberrations and the RTC command law uses the resulting slope vector to obtain the voltages to apply on the DM. Most of the control laws used require a matrix inversion and matrix-vector multiply, at frequency around 1 kHz. With the Extremely Large Telescope (ELT) generation, the real-time control of the AO loop becomes one of the most challenging issue due to the high computation power required (large matrices and high frequency) and the energy consumption inherent to the telescope (based in isolated regions). The European Green Flash project is in line with this initiative and aims to build a prototype for a Real-Time Controller (RTC) at the ELT scale. We propose a GPU-based solution because of their great energy efficiency and throughtput capabilities. In order to respect the constraints (jitter, throughput), we choose a very low level approach using FPGA based network and use perpetual kernel to handle all the computation steps that include pixel calibration, slopes and command vector computation. This approach simplifies the latency management by reducing the communication but leads us to re-implement an entire AO control loop and some GPU's standard features : communication mechanisms (guard, peer-to-peer), algorithms (generalized matrix-vector multiplication, reduce/all reduce) and new synchronization mechanisms on a multi node - multi GPUs system. First, we will detail here the context and the data pipeline. Then we will expose the performance (time, jitter) we obtained and the scalability of the solutions. Finally, we will detail the validation of the whole loop with the COMputing Platform for Adaptive optics SystemS (COMPASS), that performs end-to-end AO simulations at ELT-scale.

Author: Author: R. Ammendola[1], A. Biagioni[2], P. Cretaro[2], O. Frezza[2], G. Lamanna[3], F. Lo Cicero[2], A. Lonardo[2], M. Martinelli[2], P. S. Paolucci[2], E. Pastorelli[2], L. Pontisso[4], F. Simula[2], P. Valente[2] and P. Vicini[2] ([1]INFN Roma - Tor Vergata; [2]INFN Roma - Sapienza; [3]INFN Laboratori Nazionali di Frascati; [4]INFN Pisa;)
Title: *NaNet: a family of PCIe based Network Interface Cards for High Energy Physics*

The state-of-the-art technology of future detectors requires the High Energy Physics community to address the problem of the management of astounding amount of data. By leveraging the parallel architecture of many-core computing devices such as Graphics Processing Units (GPU) or Many Integrated Core (MIC), it will be possibile to use new algorithms allowing an improved trigger selectivity and to scale concurrently down the size of computing farms.

From the point of view of communication, moving data between the detectors readout system and computing nodes in a fast and reliable way is one of the main critical issues. The goal of the NaNet project is the design and implementation of PCI Express (PCIe) FPGA-based Network Interface Cards (NICs) featuring low-latency, real-time data transport. Key features of the NaNet architecture are the flexibility in configuring the number and kind of its I/O channels; the full offloading of the network protocols stack, thus avoiding OS jitter effects and guaranteeing a deterministic behaviour of the communication latency; the I/O streams processing, such as to reorganize data coming from detectors on the fly (e.g. data compression/decompression and reformatting or merging of event fragments); the zero-copy networking capabilities, implemented by means of a memory copy engine that follows the RDMA paradigm for both CPU and GPU, with support for nVIDIA GPUDirect RDMA protocol. Moreover, the RDMA engine is assisted by a proprietary Translation Look-aside Buffer based on Content Addressable Memory performing virtual-to-physical memory address translations.

On the software side, a Linux kernel device driver offers its services to an application level library, which provides the user with a series of functions to: open/close the device; register and de-register circular lists of receiving buffers (CLOPs) in CPU and/or GPU memory; manage software events generated when a receiving CLOP buffer is full (or when a configurable timeout is reached) and received data are ready to be consumed.

A configuration of the NaNet design, featuring four 10GbE channels for the I/O and a PCIe x8

Gen2 host interface, has been integrated during the 2016 data taking at the CERN NA62 experiment to interface the readout of the RICH detector to a GPU accelerated server performing multi-ring pattern reconstruction. Results will be then sent to the central L0 processor, where the trigger decision is made taking into account information from other detectors, within the overall time budget of 1 ms.

Assessment of the real-time characteristics and performances of the system during the 2016 NA62 Run will be provided and analyzed.

Author: Everett Phillips[1], Xiaojue Zhu[2], Vamsi Spandan[2], John Donners[3], Gregory Ruetsch[1], Rodolfo Ostilla-Monico[4], Yantao Yang[2], Detlef Lohse[2], Roberto Verzicco[2,5], Massimiliano Fatica[1], Richard J.A.M. Stevens[2] ([1]NVIDIA, [2]PoF, University of Twente, [3]SURFsara, [4]Harvard University, [5]DII Uniroma2)
Title: *A Navier-Stokes Solver for Wall-Bounded Turbulent Flows on GPU Clusters*

Direct numerical simulations (DNS) provide a valuable tool for studying in detail the underlying, and currently not fully understood physical mechanisms behind turbulence. Turbulence is a dynamic and high dimensional process, in which energy is transferred (cascades) from large vortices into progressively smaller ones, until the scale of the energy is so small that they are dissipated by viscosity. In DNS all flow scales need to be resolved and for very turbulent systems this requires an immense computational power. Efficient code parallelization is essential to obtaining scientific results. In this work AFiD, a highly parallel energy conserving second-order finite-difference scheme for the incompressible Navier-Stokes equations, has been ported to GPU clusters. Finite-difference schemes have several advantages, i.e. they are very flexible, allow for complex boundary conditions and/or structures interacting through the immersed boundary method with relative ease. Because lower-order schemes are computationally very inexpensive, the grid resolution can in general be larger for the same computational cost compared to higher order schemes. The code is open sourced on http://www.afid.eu. The GPU porting has been carried out in CUDA Fortran with the extensive use of kernel loop directives (CUF kernels) in order to have a source code as close as possible to the original CPU version; just a few routines have been manually rewritten. A new transpose scheme has been devised to improve the scaling of the Poisson solver, the main bottleneck in incompressible solvers. The results of the GPU version have been carefully validated. The GPU version can reduce the wallclock time by an order of magnitude compared to the CPU version for large meshes. Due to the increased performance and efficient use of memory, the GPU port of AFiD can perform simulations in parameter ranges that are unprecedented in thermally-driven wall-bounded turbulence. In our applications, we focus on Rayleigh-Benard (RB) convection, the flow in a fluid layer between two parallel plates; one heated from below and cooled from above, with the aim of achieving a high enough thermal driving (Rayleigh number) to enter the "ultimate" regime, of relevance to geo- and astrophysics, and to help understand the discrepancies between high-Rayleigh number experiments. AFiD can easily be extended to other wall-bounded flow such as Channel flow or Taylor-Couette flow. In future work we plan to utilize the CPU cores (now completely idle) together with the GPUs in the implicit part of the solver. We are more interested in the CPU memory than the CPU flops, but depending on the node configuration, the CPU cores could give a good performance boost. Subdiving each vertical domain in two subdomains, we can process one on the CPU and one on the GPU. The relative size of the subdomains could be determined at runtime, since the workload per cell is constant. The split will be in the outermost dimension (z) and it will require additional (local memory) halo exchanges. In addition, IO performed by the CPU's can be made concurrently with flow calculations performed on the GPU.

Author: Enrico Calore, Alessandro Gabbana, Sebastiano Fabio Schifano and Raffaele Tripiccione (INFN and Università degli Studi di Ferrara)
Title: *Heterogeneous implementation of the D2Q37 Lattice Boltzmann Method*

Lattice Boltzmann Methods (LBM) are a family of CFD algorithms widely used to solve fluid flows in several regimes. This class of algorithms - discrete in space, time and momentum space and living on a discrete and regular grid of points - offers a large amount of easily identified available parallelism; LBM algorithms are therefore ideally suited for implementations using massively parallel codes, running on modern many-core HPC systems. Currently, hi-end HPC systems are more and more often based on massively parallel heterogeneous architectures, in which each of a large number of computing nodes has a host processor and one or more accelerators, based on GPUs, Xeon-Phi processors or - in some cases - FPGAs. Over the years, several efficient LBM codes have been developed for these architectures, mapping the parallel structure of the underlying algorithm onto the architectural features available on the target processor and system. In most cases, however, when considering accelerator based implementations, these codes offload almost all the intensive computational steps onto the accelerator; this choice is done to avoid bottlenecks associated to the limited communication bandwidth (and large latency) between host and accelerator, but does not exploit precious computing resources available on the host. In this contribution we go one step further, describing an integrated and carefully optimized implementation of a D2Q37 LBM code that uses concurrently the computing resources of the accelerator and those of the host node, and also automatically balances the workload between the two computing engines. The main advantages of this implementation are better single-node performance, increased data locality, reduced communication overheads for massively parallel implementations and better (strong and weak) scaling behavior. We provide performance results on a cluster where each node is powered by two conventional multi-core CPUs and two Xeon-Phi many-core accelerators and compare with more traditional optimized implementations of the same code for several HPC architectures, including GPUs and multi and many core processors.

Author: Gilles Grasseau, Thomas Strebler, Arnaud Chiron, Pascal Paganini, Florian Beaudette (Leprince-Ringuet Laboratory, Ecole polytechnique - France)
Title: *A MPI/OpenCL hybrid implementation of the Matrix Element Method in the context of the Higgs boson properties analyses*

In the proton-proton collisions at the LHC, the associate production of the Higgs boson with two top quarks has not been observed. This ttH channel allows directly probing the coupling of the Higgs boson to the top quark, the only Yukawa coupling of the Standard Model of particle physics expected to have a magnitude of the order of the unity. Thanks to the increased centre-of-mass energy of the collisions to 13 TeV, as well as the integrated luminosity collected, this process could in principle be observed during the ongoing Run II data taking. Given the complexity of the final state, the Matrix Element method (MEM) is well suited to separate the signal from the background. Compared to other machine learning approaches requiring training, the MEM allows a direct comparison between theory and observations. This sophisticated method is however very CPU time consuming and requires huge powerful computing platform to perform the CMS analyses carried out at our laboratory in a reasonable time. The Matrix Element method is based on the computation of high dimensional integrals. We will describe how we transform the main components of these computations into OpenCL kernels: VEGAS for the integration part, the MadGraph code generator to compute the Matrix Element terms and the parton distribution function calculations for LHAPDF part. The different parallelism levels: nodes, cores and vector (SIMD) involving a mix of the MPI and OpenCL standards will be presented too. Finally, the optimization tuning of the "big" kernels to achieve of a powerful and portable application for High Performance Computing platforms equipped with accelerator systems such as Intel, NVidia and/or AMD, will be described.

Author: Felix Höfling (Department of Mathematics and Computer Science, Freie Universität Berlin)
Title: *GPU-powered molecular dynamics simulations in statistical physics*

Targeting the massively parallel architecture of Nvidia's GPU accelerators, we have developed the software HAL's MD package [1]. It is a high-precision molecular dynamics package for large-scale simulations of complex dynamics in inhomogeneous liquids. The implementation achieves excellent conservation of energy and momentum at high performance by using an increased floating-point precision where necessary [2]. I will demonstrate that, using the example of a supercooled binary Lennard-Jones mixture, insufficient floating-point accuracy may result in quantitatively and even physically wrong results. Next, data locality is crucial for best efficiency. The software thus minimises disk usage by the in situ evaluation on the GPU of thermodynamic observables and dynamic correlation functions. Further, we have developed the H5MD file format as an open standard for molecular simulation data for the structured, compressed, and portable storage of multi-dimensional datasets [3]. The relevance of GPU accelerators in high-performance simulations will be exemplified for two cases. First, GPUs allowed us to obtain a fresh view on the structure of liquid-vapour interfaces (Fig. 1a), which are roughened due to capillary-wave and bulk-like fluctuations [4]. Having access to huge system sizes, unprecedented in the field, allowed us to unambiguously separate the simulated X-ray scattering intensity into an interface structure factor and a bulk-like background (Fig. 1b). The former diverges at small wavenumbers and carries information about the surface tension, for which we found an unexpected dependence on temperature and wavelength. Second, GPUs boosted our comprehensive study of the structure and dynamics of binary liquid mixtures near their continuous demixing transitions (Fig. 1c) [5]. Near the transition, the liquid mixture exhibits critical fluctuations of the local chemical composition and critical slowing down. Since the correlation length and the relaxation time diverge concomitantly, simulations of the critical behaviour must be able to cover both large system sizes and long simulation runs. The mixtures considered cover a variety of densities, a wide range of compressibilities, and different pair interactions. Nevertheless, our simulation results can be fully rationalised in terms of the dynamic universality class of the so-called model H' A particular challenge is the understanding of interdiffusion processes, for which we found that the corresponding trans- port coefficient displays super-diffusive behaviour (Fig. 1d). This surprising observation would have been difficult to make without the use of GPU accelerators.

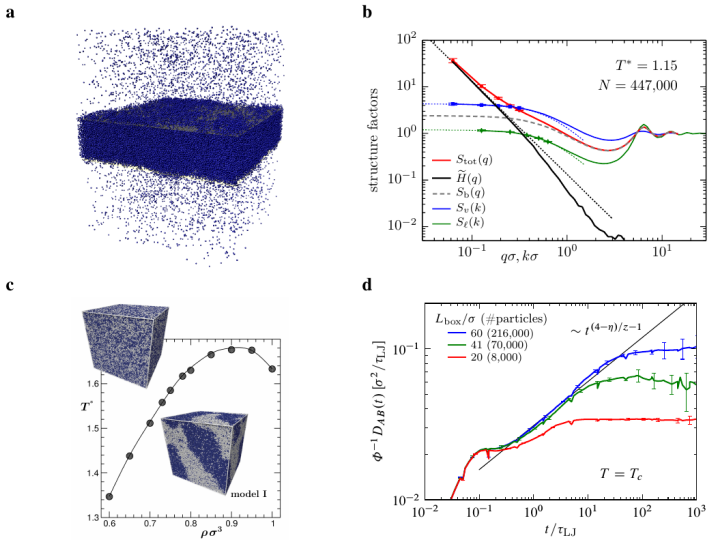


Figure (color online): (a) Simulation snapshot of a liquid-vapour interface and (b) corresponding interface and bulk structure factors [4]. (c) -line of the demixing transition in the phase diagram of a binary liquid mixture [5], and (d) the finite-size behaviour of the interdiffusion coefficient in such a mixture.

[1] Highly Accelerated Large-scale Molecular Dynamics package (2007-2016), see http://halmd.org.
[2] P. H. Colberg and F. Höfling, Comput. Phys. Commun. 182, 1120 (2011).
[3] P. de Buyl, P. H. Colberg, and F. Höfling, Comput. Phys. Commun. 185, 1546 (2014).
[4] F. Höfling and S. Dietrich, EPL 109, 46002 (2015).
[5] S. Roy, S. Dietrich, and F. Höfling, arXiv:1606.05595.

Author: Junichi Kanzaki (KEK)
Title: *Monte Carlo integration and event generation on GPU and their application to particle physics*

We use a graphics processing unit (GPU) for fast computations of Monte Carlo integrations and event generations. Two widely used Monte Carlo integration programs, VEGAS and BASES, are parallelized for running on a GPU. Event generation program, SPRING, that generates unweighted events based on BASES integration output, is also parallelized. Their algorithms are optimized to achieve high occupancy of parallel processors of GPU and the performance of Monte Carlo integration and event generation significantly improves especially for complicated and time consuming computations. We test the programs by comparing integrated cross sections and event process time obtained by programs running on CPU (written FORTRAN and/or C) and those running on GPU, by using W^+ plus multi-gluon production processes at LHC. The integrated results agree with each other within statistical accuracies and the programs run more than 50 times faster on the GPU than on the CPU. We apply the programs to fast calculations of helicity amplitudes and event generations of general Standard Model (SM) processes at LHC. We test all the codes by comparing cross sections of LHC multi-jet processes associated with production of heavy particles (single and double weak bosons, a top-quark pair, Higgs boson plus a weak boson or a top-quark pair, and multiple Higgs bosons via weak-boson fusion), where all the heavy particles are allowed to decay into leptons and light quarks. Compared with the usual CPU programs, we obtain more than 50 times better performance on the GPU, except for most complicated processes like vector bosons+4-jets, a top-quark pair+3-jets or H$t\bar{t}$+2g, for which the GPU gain over the CPU can still be above 20.

Author: Suren Khachatryan and Zareh Garapetian (American University of Armenia)
Title: *Formation of Steady-state Structure in Gravitating Disks by Nonlinear Density Perturbations*

We continue a decades-long study of nonlinear density perturbations in a rotating gravitating disk at the edge of Jeans instability. The system of two-dimensional equations describing the dynamics of an embedded thin gaseous disk includes the equations of motion of the gas, the continuity and the Poisson equations and the polytropic equation of state, where the gravitational potential of the non-flat subsystem contributes to the equilibrium of the disk, but does not directly influence the perturbations. The principal equation has been derived in the WKB approximation and the dynamics of a narrow wave packet excited near the minimum of the dispersion curve has been considered in preceding papers. It was analytically shown that in the cubic order approximation of small increments of linear waves the equation has finite steady-state solutions that can be interpreted in terms of the global spiral structure. The results of the nonlinear wave theory have been compared with observational data of several spiral galaxies with different types of rotation curves. Let the disk be gravitationally unstable in its certain regions, where weak or random fluctuations may trigger growth of density waves with exponentially increasing amplitudes. In the present paper we analyze the transient phase of solutions of the governing nonlinear hyperbolic equation to answer a question, if nonlinear interaction between different harmonics can result in a self-consistent spiral pattern with small but finite amplitude of density perturbation. We implement GPU parallelized integration along characteristics and discuss the influence of the initial conditions on the resulting steady-state patterns. We also validate them against the adopted assumptions and approximations. Finally, we propose GPU parallelization of the method extension that incorporates mesh adaptation and provides a straightforward strategy in recognition and treatment of most critical regions, such as discontinuities and shock fronts.

Author: Natalia Kolomoyets and Vadim Demchik (Dnepropetrovsk National University)
Title: *Multipart SU(N) mode in QCDGPU package for Monte Carlo lattice simulations*

QCDGPU is a GPU-accelerated open-source software suite for Monte-Carlo (MC) simula- tions of lattice gauge theory and O(N) models. Its application areas include investigating finite temperature and external chromomagnetic field effects. A key feature of QCDGPU is its sup- port for operations on large lattices. It implements the conventional decomposition mechanism for large lattice simulations - splitting large lattices into several parts for single or multi-GPU MC simulations - and can break lattices into unequal sized parts to optimally support GPUs with different performance. Lattice partioning is performed along X spatial axes to minimize redundant data transfer, and small lattices are detected and not partitioned when they are small enough for a single device to perform better. Multipart lattice splitting solves two important problems: overcoming GPU memory limi- tations, and reducing computing time by using multiple computing devices. However, splitting gives rise to problems caused by the semi-locality of MC lattice updating and non-locality of some measurement procedures (e.g., Wilson loop and Polyakov loop calculations). QCDGPU solves these problems in a non-trivial way. To ensure code portability, QCDGPU kernels are written in OpenCL. The host part of the program is implemented in C/C++ using of OpenMP. QCDGPU manages multi-GPU MC simulation processes using independent OpenMP host threads, which also require additional synchronization through a barrier mechanism. MC simulation parameters are stored in text files for each task. Most parameters (like lattice and sublattice sizes, etc.) are implemented via precompiler directives, which increases the overall performance of QCDGPU simulations. The rest of parameters are passed to the GPU kernels as either scalar or array arguments. Together with a custom kernel cache implementation, this also reduces the need to frequently recompile GPU kernels. Using text files for tasks furthermore makes it much easier to use in a distributed computing environment.

Author: Maxime Lainé (Observatoire de Paris LESIA CNRS)
Title: *A GPU based RTC for E-ELT Adaptive optics: addressing the latency / jitter constraints*

In a previous session our team advocated the use of GPUs for the future European Extremely Large Telescope (E-ELT) adaptive optics (AO) systems' real-time controller (RTC) in order to respond to the computational power needed, and to the constraints in cost, latency and jitter set by the European Southern Observatory (ESO) leading the telescope construction. We presented the work ongoing at our laboratory toward this direction. Using a custom designed FPGA and its direct memory access (DMA) capabilities as a network interface controller (NIC) we handled camera's data protocol (UDP and GigEVision) and eventually pre-processing operations (pixels reordering and thresholding) and then brought the wave front sensor (WFS) camera data stream directly into the GPU's memory. In combination with a GPU kernel running in perpetual fashion and polling its memory to detect incoming datas. We managed to reduce latencies induced by memory copies and kernel launches usually taking place in GPU programming paradigms. In this session we wish to present progresses made toward our unified platform. Particularly we developped the ability to send deformable mirror (DM) commands directly from the GPU's memory, controlling FPGA's DMA engine from a GPU kernel. Thus, our system fully bypass the CPU for its execution, said CPU is used exclusively for setup, configuration, execution control and terminaison operations. Making our system performances totally independent from CPU load and allows us to use the CPU for meta-data gathering and performance analysis during the system execution. Additionally we will present further results obtained on our prototype for high speed, low latency, image acquisition and processing system dedicated to AO systems, as well as a more fine-grained latency measurement on the whole chain, and a comparison in terms of latency and jitter with a system using classic data management and execution schemes seen in usual GPU programming.

Author: Matteo Lulli[1], Massimo Bernaschi[2] and Mauro Sbragaglia[1] ([1]Department of Physics and INFN, University of Tor Vergata; [2]Istituto per le Applicazioni del Calcolo, CNR;)
Title: *GPU Based Detection of Plastic Events in Lattice Boltzmann Emulsions*

Emulsions are complex systems which are formed by a number of droplets dispersed in a solvent. Such droplets do not coalesce but rather push against each other leading to non-trivial effects in the overall flowing dynamics. In fact, such systems possess a so-called yield stress below which the system has an elastic response to an external forcing, i.e. it deforms but it does not flow. While above the yield stress the system flows, the rheological properties are those of a non-Newtonian fluid, i.e. the stress is not proportional to the shear. In the solid-like regime the network of the droplets interfaces stores the energy coming from the work exerted by an external forcing. This energy can also be used to actually move the droplets in a non-reversible way named *plastic event.* The simplest bulk event involves four droplets and it is called T1. More formally plastic events are related to topological changes of the droplets configurations. There are some phenomenological models aiming at describing the collective behaviour of such systems that assume the proportionality between the rate Γ of T1 events and the so-called *fluidity* field $f = \dot{\gamma}/\sigma$, where $\dot{\gamma}$ is the shear rate and σ is the stress: $f \propto \Gamma$ [1]. Indeed, droplets rearrangements are the basis of the overall flowing behaviour. Topological changes can be defined with respect to the Delaunay triangulation of the centers of mass of the droplets in the emulsion, *e.g.*, whenver one of the edges of the triangulation changes then a topological change, and thus a plastic event, has happened. There exist GPU implementions of Lattice Boltzmann models [2] for simulating emulsions dynamics. The only way not to hamper such simulations is to resort to a GPU implementation for a detector of plastic activity avoiding unecessary device-to-host communication. We propose a "Trigger" mechanism [3], executing along the simulation, able to locate such events and to analyze them only when needed minimizing communication and disk space, while at the same time maximizing the number of detected events. The Trigger naturally decomposes in three tasks: finding the centers of mass, finding the Delaunay triangulation, comparison of two triangulations. Each task is executed on the GPU relying on library implementations, slight modifications of previous algorithms and new solutions exploiting the discrete nature of the lattice.

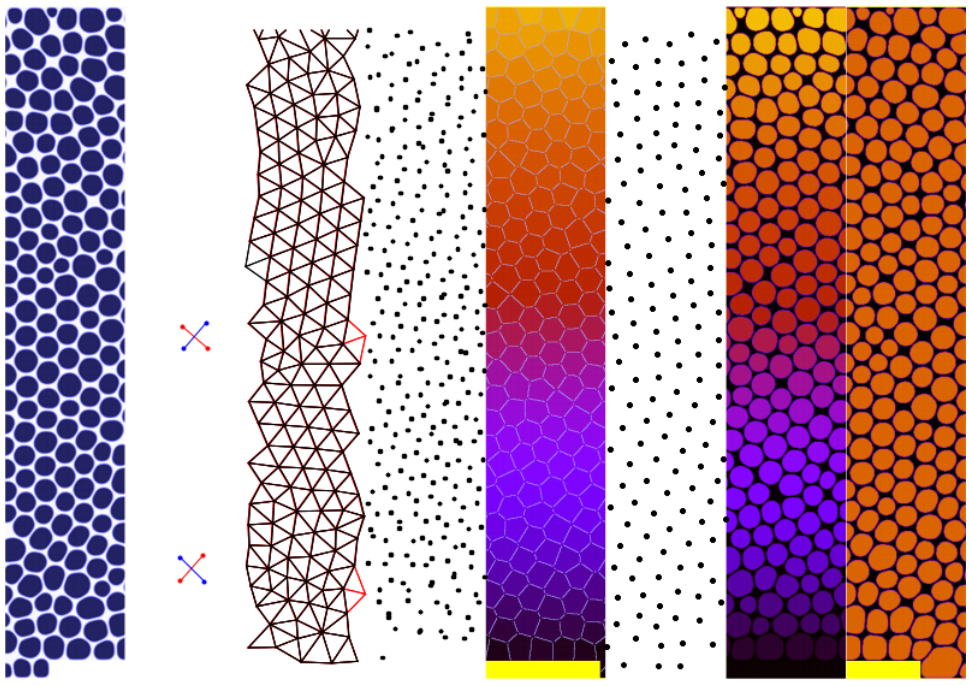


Figure (color online): The workflow of the trigger mechanism. On the left the Lattice Boltzmann density field is reported. From right to left: a) we determine which points of the lattice are in a droplet which are outside and which are obstacles (yellow colour); b) we uniquely label each droplet as depicted in the second panel, each label corresponds to a

different colour; c) we compute each center of mass; d) we compute the digital Voronoi tesselation; e) we extract the vertices of the Voronoi polygons as shown in the fifth panel; f) we compute the Delaunay triangulation and finally g) we compare two different triangulations to extract the topological changes.

[1] L. Bocquet et al., "Kinetic Theory of Plastic Flow in Soft Glassy Materials", Phys. Rev. Lett. 103, 036001 (2009)
[2] M. Bernaschi et al., "Graphics processing unit implementation of lattice Boltzmann models for flowing soft systems", Phys. Rev. E 80, 066707 (2009)
[3] Massimo Bernaschi, Matteo Lulli, Mauro Sbragaglia, "GPU Based Detection of Topological Changes in Voronoi Diagrams" arXiv:1607.00908

Author: Carmela Luongo[1,2], Niccolò Camarlinghi[1,2], Nicola Belcari[1,2], Gianluca Lamanna[2], Giancarlo Sportelli[1,2], Alberto Del Guerra[1,2] ([1] Department of Physics, University of Pisa; [2] INFN Pisa)
Title: *PET Image Reconstruction on GPU*

Iterative algorithms for 3D image reconstruction in Positron Emission Tomography (PET) improve image quality by allowing more accurate modeling of the data acquisition than analytical methods. However, the massive amount of computation involved in these algorithms in order to incorporate accurate system modeling makes their implementation a very challenging task. One of the key components of the iterative methods is the System Response Matrix (SRM) that describes the relation between the measured data and the image. Despite the fact that this matrix is sparse and presents symmetries that can be exploited to reduce its size, the memory footprint of the SRM is one of the main issue with the iterative PET reconstruction. In this work we present a parallel implementation of an iterative reconstruction algorithm that allows us to compute the SRM coefficients "on-the-fly". We tested our implementation of the algorithm using data acquired with the IRIS PET/CT scanner, a new preclinical system for mice and rats. The reconstructed Field Of View (FOV) of the scanner is made of 101x101x120 voxels of 0.855 mm x 0.855 mm x 0.855 mm each and the scanner features 24 million of Lines Of Response (LOR, which is the straight line connecting the centers of two detectors). We implemented the iterative Maximum Likelihood Expectation Maximization (MLEM) algorithm on Graphics Processing Units (GPU) using CUDA. We computed the SRM "on-the-fly" by using the raytracing algorithm proposed by Siddon in its Jacobs' fast version. In order to provide a better sampling of the image volume, we used a multiray approach implementing 256 rays per LOR. We performed the parallelization of each stage of the MLEM algorithm. We implemented two CUDA kernels: the forward projection, which computes the activity along one projection path, and the back projection, which re-distributes the activity back to its original path. We optimized the forward projection kernel by storing the image volume in a 3D texture that is optimized for spatial locality. Our implementation of the Siddon algorithm overcomes some GPU challenges such as thread divergence and limited bandwidth of global memory by exploiting GPU capabilities such as efficient use of texture memory. Atomic sum was used in order to avoid race conditions due to simultaneous access to the same memory element by parallel threads. LOR data were stored in the GPU global memory in a way that the memory accesses of threads within a warp are coalesced into a single memory transaction. We compared the GPU reconstruction with the CPU implementation distributed with the IRIS PET scanner with respect to image quality and processing time. The standard CPU reconstruction was run on a eight core CPU Intel i7 3770. We performed the GPU-CUDA reconstruction on two different architectures: Kepler GeForce GTX TITAN with Compute Capability 3.5 and Maxwell GeForce GTX980 Ti with Compute Capability 5.2. The maximum % difference between the pixel values of the reconstructed images is 10^{-4}. The reconstruction performed with the Maxwell architecture achieved a speed up factor of five with respect to CPU implementation. These results were obtained both in the Monte Carlo and in the experimental data on small animals.

Author: M. Mastropietro[1,3], D. Bastieri[2,4], S. Lombardi[1,3], L. A. Antonelli[1,3], for the ASTRI Collaboration[5] and CTA Consortium[6] ([1]INAF - Rome Astronomical Observatory; [2]GPU Research Center, University of Padova; [3]ASI Science Data Center, Rome; [4]Dept. Physics and Astronomy, Univ. Padova and INFN Padova; [5]`http://www.brera.inaf.it/astri` ; [6]`http://www.cta-observatory.org`)
Title: *ASTRI SST-2M data reduction software on low-power and parallel architectures*

In the framework of the Cherenkov Telescope Array (CTA) International observatory, the Italian National Institute for Astrophysics (INAF) is leading the "Astrofisica con Specchi a Tecnologia Replicante Italiana" (ASTRI) project devoted to the development of a next generation of small-sized dual-mirror (SST-2M) Imaging Atmospheric Cherenkov Telescopes (IACTs). A prototype of the ASTRI SST-2M telescopes has been installed in Italy at the INAF "M. G. Fracastoro" astronomical station located at Serra La Nave on Mount Etna, Sicily, and will be operational by the end of 2016. A mini-array of nine ASTRI SST-2M telescopes has been proposed to be installed at the southern site of the CTA, as an initial seed of the entire observatory. The ASTRI mini-array is a collaborative effort led by INAF in synergy with the University of Sao Paulo and FAPESP (Brazil) and the North-West University (South-Africa). IACTs are typically located in isolated places in order to avoid luminous sky background. In such a remote site, the capability of each telescope to process its own data before sending them to a central acquisition system provides a key advantage: carrying out preliminary data reduction on the telescope would greatly decrease the bandwidth and power required by an array installation. We implemented the complete analysis chain required by a single ASTRI SST-2M telescope on an NVIDIA Jetson TK1, a development board equipped with an heterogeneous NVIDIA Tegra System-on-Chip. Leveraging its efficient Kepler GPU and the four ARM cores, we show, on Monte Carlo simulation bases, that it can process twice as much of the required data flow (more than 2k events/s) with a power consumption lower than 10 W. This makes Jetson TK1 a promising embedded processing module for the on-line data analysis of gamma-ray astronomy with Imaging Atmospheric Cherenkov Telescopes.

Author: Yohei Miki and Masayuki Umemura (Center for Computational Sciences, University of Tsukuba)
Title: *GOTHIC: Gravitational Oct-Tree code accelerated by HIerarchical time step Controlling*

Collisionless N-body simulations are frequently employed to explore the formation and evolution of gravitational many-body systems such as large scale structure or galaxies. Employing a large number of particles is essential for performing N-body simulations that resolve astrophysical phenomena. Since the computational cost of the direct method is too high to investigate realistic phenomena in detail, many earlier studies have attempted to accelerate N-body simulations. The tree method (Barnes & Hut 1986) is a widely implemented algorithm for reducing the amount of computation. Many N-body simulations adopt a shared time step, where all particles share the time step that is required to track the orbital motion of particles evolving in the shortest time span. Therefore, adopting a shared time step demands unnecessary, additional computations to track the evolution of the system. To overcome this disadvantage, a scheme in which every particle has its own individual time step was introduced by Aarseth (1963). Because an individual time step method is not suitable for parallelization, McMillan (1986) proposed the use of block time steps (or sometimes called hierarchical time steps) in which a group of particles has the same time step. Adopting block time steps can reduce the number of computations by decreasing the number of moving particles in each time step. Many earlier studies showed that the tree method efficiently works on GPU (Nakasato 2012; Ogiya et al. 2013; Bédorf et al. 2012, 2014; Watanabe & Nakasato 2014). However, none of the studies has coupled with the block time step on GPU. We have developed a Gravitational Oct-Tree code accelerated by HIerarchical time step Controlling, named "GOTHIC" (Miki & Umemura, submitted to New Astronomy), which adopts both the tree method and the hierarchical time step. The presented code adopts the breadth-first search, and runs entirely on GPU, just like the public code Bonsai by Bédorf et al. (2012, 2014). The algorithm in the tree traversal is an improved version of the algorithm proposed by Ogiya et al. (2013), which used a depth-first search. The novel aspect of our code is that it is adaptively optimized by monitoring the execution time of each function on-the-fly and minimizes time-to-solution by balancing the measured time of multiple functions. We measured the performance of GOTHIC for the particle distributions based on a galaxy model on NVIDIA Tesla M2090, K20X, and GeForce GTX TITAN X, which are representative GPUs of the Fermi, Kepler, and

Maxwell generation of GPUs. The results of the performance measurements show that the hierarchical time step achieves a speedup by a factor of around 3–5 compared to the shared time step. The measured elapsed time per step of GOTHIC is 0.30 s or 0.44 s on GTX TITAN X for the particle distributions of the Andromeda galaxy or the NFW sphere, respectively, with 16,777,216 particles. The averaged performance of the code corresponds to 10-30% of the theoretical single precision peak performance of the GPU.

Author: Marco Palombo[1,2], Clémence Ligneul[1,2], Chloé Najac[1,2], Juliette Le Douce[1,2], Julien Flament[1,3], Carole Escartin[1,2], Philippe Hantraye[1,2,3], Emmanuel Brouillet[1,2], Gilles Bonvento[1,2], and Julien Valette[1,2] ([1]MIRCen, I[2]BM, DSV, CEA; [2]UMR 9199 (Neurodegenerative Diseases Laboratory), Université Paris-Saclay, CNRS; [3]UMS 27, INSERM)
Title: *GPU based modeling pipeline to extract brain cell morphology from in vivo diffusion-weighted MR spectroscopy data*

Here we report about a novel paradigm for noninvasive brain microstructure quantification we have recently introduced [1]. Original diffusion modeling is merged with cutting-edge diffusion-weighted magnetic resonance spectroscopy (DW-MRS) experiments to capture features of cellular morphology by using brain metabolites as molecular probe. Brain metabolites are endogenous molecules being mostly intracellular, which can diffuse in the long neuronal and astrocytic fibers, exploring the ramifications and experiencing restriction at the extremity of fibers. This makes their apparent diffusion coefficient (ADC) depend on these cellular morphological features. The diffusion of cell-specific metabolites (N-acetylaspartate and glutamate reside essentially in neurons; myo-inositol and choline compounds are preferentially in glial cells, especially in astrocytes; and the total creatine pool is evenly distributed between astrocytes and neurons) is here investigated at ultra-long diffusion times (td up to 2 s) in the healthy rodent and primate brain by DW-MRS in vivo, to specifically probe the intracellular compartment at increasing spatial scales. Then, massive Monte Carlo (MC) simulations of particles diffusing in many synthetic cells, parameterized by a small set of morphometric statistics allowing the generation of complex and heterogeneous morphologies, are run on graphics processing unit (GPU) and iterated to fit DW-MRS. Specifically, a set of four parameters is used to describe the cellular morphology: mean and S.D. of segment length (Lsegment and SDLsegment); and embranchments along cell processes (Nbranch and SDNbranch) generating synthetic branched cells. N=2000 particles are positioned in each cell and diffusion is simulated by MC (with effective intracellular diffusivity D intra), allowing to compute DW-MRS signal and corresponding ADC. 80 different cells are generated to account for cellular heterogeneity. Dintra, Lsegment, SDLsegment, Nbranch and SDNbranch are iteratively changed until simulated ADC matches measured ADC. The fitting procedure uses a combined Parallel-Tempering and Levenberg-Marquardt approach, for unsupervised initialization and quick convergence. Code was implemented in Matlab to manage the computation in parallel on GPU device (Nvidia Quadro K2000), making it possible to fit experimental ADC in 3 minutes. Fitting stability relative to noise was assessed by studying the bias and the coefficient of variation (CV) of the estimated parameters on 250 MC trials. After investigating model robustness, we apply it on metabolites ADC measured in vivo in the mouse and macaque brain. Resulting synthetic cellular compartments (tentatively neurons and astrocytes) are quantitatively compared with histological data. Very good agreement between Sholl analysis on real and synthetic astrocytes validates our approach and assumptions [1]. We also measure increased size and complexity of synthetic astrocytes in primate compared to mouse, while dendritic organization appears better conserved throughout species [1]. The new paradigm introduced here [1], exploiting the versatility and computational power of GPUs, opens new possibilities to noninvasively extract quantitative information about cell size, complexity and heterogeneity in the brain. [1] Palombo et al. PNAS 2016: 113(24); 6671-6676.

Author: Francisco Fernando Ramírez (DQIAyQF / INQUIMAE, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires)
Title: *The LIO project: optimized quantum molecular simulations*

Computational Chemistry is a branch of science that uses computers to solve the mathematical equations that describe the properties of atoms, molecules and materials. Within this broad subject, we will discuss here methods based on quantum mechanics that calculate electronic states and simulate the joint dynamics of nuclei and electrons in molecules. It is an area of research where the complexity of the model demands a compromise between the size of the system and the accuracy of the description, so the incorporation of new technologies that increase processing power becomes critical for its advancement. The LIO project 1,2,3 is a multidisciplinary effort arising from the collaboration of groups in Computer Science and Chemistry of the University of Buenos Aires. The LIO library we develop is currently able to both calculate the ground electronic state of the system (the most stable, or the one of minimum energy for a given set of nuclear positions), and to propagate electronic states through time. To obtain the ground state, an iterative method is employed which solves the self consistent equations of Density Functional Theory (DFT), and can be coupled with another package called Amber which handles nuclear movement. The process of successively moving the nuclei and calculating the electronic ground state is known as Born-Oppenheimmer molecular dynamics and, amongh other things, permits the simulation of a wide variety of chemical processes and reactions. To propagate the electronic states in time, the code uses the expression derived from the formalism of Time-Dependent DFT (TD-DFT), allowing to study the interaction of oscillating electric fields (such as light) with matter, to obtain from this the electronic spectra of molecules. We are now working, on the one side, in more controlled ways of obtaining excited states and analizing their properties, and on the other, in implementing a more general method called Ehrenfest Dynamics that allows the joint time evolution of nuclei and electrons beyond their ground state, thus enabling the full simulation of those chemical processes and reactions that are induced by light. The most computationally demanding procedures required by these aplications are operations on matrices and evaluation of integrals (both analitical and numerical). Optimized matrix algebra in GPU is performed by external libraries such as cuBLAS and MAGMA, whereas the optimization of the integration procedures required the design of specific kernels, most of which have been targeted at the numerical integrals. I will present here how through the use of general and ad-hoc optimizations, such as the use of mixed precision and CUDA specific implementations, this customized kernels have now reached a x8 speedup on a compute node equipped with GeForce GTX 980 and a single socket i5-3450, when comparing CPU only with the whole node (CPU + GPU). [1] U. N. Morzan, F. F. Ramírez, M. B. Oviedo, C. G. Sanchez, D. A. Scherlis, M. C. G. Lebrero, Electron dynamics in complex environments with real-time time dependent density functional theory in a QM-MM framework, J Chem Phys 140 (16) (2014) 164105. [2] M. A. Nitsche, M. Ferreria, E. E. Mocskos, M. C. González Lebrero, GPU Accelerated Implementation of Density Functional Theory for Hybrid QM/MM Simulations, Journal of Chemical Theory and Computation 10 (3) (2014) 959-967. [3] https://github.com/nanolebrero/lio

---

Author: David Rohr (Frankfurt Institute for Advanced Studies)
Title: *Title: Portable generic applications for GPUs and multi-core processors: An analysis of possible speedup, maintainability and verification at the example of track reconstruction for ALICE at LHC*

Fast compute clusters are a mandatory tool for today's research and industry. The ever-increasing demand of compute power can now longer be met by faster single processors with higher clock frequencies but only through parallelism. One approach leveraging parallel concepts to a very large extent are clusters with GPU-accelerated compute nodes with graphics cards as coprocessors. Even though GPU programming languages have matured and GPU hardware has become suited better for general purpose code in recent years, porting applications to GPUs poses several challenges. Verification and tuning of GPU code usually requires a CPU reference implementation of the GPU application, and portable programs that support different GPU architectures make the user vendor-independent. However, low level optimizations of algorithms are necessary to obtain maximum performance in many cases, while the optimum implementation usually varies between different GPU architectures. Maintaining different source codes for different GPU models and the CPU, possibly implementing different optimizations and using

different programming languages or models, is a great and often unacceptable effort. In order to minimize the maintenance effort and guarantee the greatest possible consistency between different hardware platforms, we suggest maintaining a generic common shared source code for all supported processors and accelerators. Many optimizations can be parameterized, or they can be enabled and disabled with compile time switches, while API specific code for certain platforms can be outsourced into small wrappers. Different language extensions to a common underlying language like C or C++ be encapsulated in headers using preprocessor definitions, conditional compilation, templates, or similar techniques. We present several GPU-accelerated applications implemented following this paradigm, which support different GPUs but also CPU-only systems, keeping the majority of the source code generic. On top of simplifying maintenance and verification, this also enables a direct performance comparison of the same algorithm on different hardware. We show and compare speedups achieved by several GPU-implementations and put them into relation to the maximum theoretically possible speedup derived from the hardware specification. Our primary example and focus is on a GPU-algorithm for the real-time reconstruction of particle trajectories (tracking) in collisions recorded by the ALICE experiment at CERN LHC. ALICE is a heavy-ion high energy physics experiment that records more than one thousand lead-lead collisions per second. The online tracking in the ALICE High Level Trigger is the most compute-intense task during real time event reconstruction. For the upcoming Run 3 of the Large Hadron Collider, ALICE is undergoing a major upgrade which will enable the data taking of lead-lead collisions at 50 kHz with 3 TB/s of incoming data from the TPC, the most important detector with respect to tracking. GPUs enable the track reconstruction at such high data rates, which CPU based compute farms of the size conceivable for the ALICE experiment are incapable of. From the experiment with the current GPU-based tracking we extrapolate to the situation after the upgrade and present possibilities to tune the GPU code accordingly.

Author: M. Senzacqua[1], G. Battistoni[2], V. Patera[1], A. Schiavi[1], S. Pioli[3], M. Ciocca[4], A. Mairani[4], S. Molinelli[4], G. Magro[4] ([1]Università degli studi di Roma "La Sapienza", SBAI Basic Sciences Applied to EngeneeringItaly; [2]INFN Milano ; [3]INFN Laboratori Nazionali di Frascati; [4]CNAO Pavia)
Title: *A fast - Monte Carlo toolkit on GPU for treatment plan dose recalculation in proton therapy*

An innovative cancer treatment technique called particle therapy is emerging. It consists of irradiating solid tumors with beams of protons or heavier ions, in particular carbons. Thanks to the peaked shape, called Bragg Peak, of charged particles energy deposition in matter it is possible to concentrate the dose on the tumor and to reduce the damage to healthy tissues. This selectivity in energy release involves the necessity of an high level of accuracy in the calculation and optimization of the beams to be sent to the patient. This computation is performed by Treatment Planning System (TPS), softwares that integrate the patient anatomic information, radiotherapist prescriptions and accelerator's parameters to compute a plan customized to the patient. Nowadays one of the major issues related to the Treatment Planing Softwares in Particle therapy is the large CPU time needed. We developed a software toolkit (FRED) for reducing dose recalculation time by exploiting Graphics Processing Units (GPU) hardware. Thanks to their high parallelization capability, GPUs significantly reduce the computation time, up to factor 100 respect to a standard CPU running software. In fact, a Full Monte Carlo tool such as FLUKA, running on CPU, can track about 750 primary/second, FRED CPU-version in the same condition reaches a simulation velocity of about 15000 primaries/second, whereas FRED running on a single GPU can simulate up to 80000 primaries/second, and this performance can be further improved by increasing the number of GPU exploited. The transport of proton beams in the patient is accurately described through Monte Carlo methods. Physical processes reproduced are: Multiple Coulomb Scattering, energy straggling and nuclear interactions of protons with the nuclei composing biological tissues. A further advantage of FRED with respect to commercial TPS used in clinical practice is that it does not rely on the water equivalent translation of tissues, but it exploits the Computed Tomography anatomical information by reconstructing and simulating the atomic composition of each crossed tissue. Thanks to the features described FRED can be used as an efficient tool for dose recalculation, on the day of the treatment. In fact it can provide in about one minute on standard hardware the dose map obtained combining the treatment plan, earlier computed by the TPS, and the current patient anatomic arrangement. Assessment results of FRED performance in terms of accuracy and calculation time in comparison to commercial TPS and the full MC approach (Fluka) will be presented.

Author: Dany Vohl[1], Christopher J. Fluke[1,2], Amr H. Hassan[1], David G. Barnes[2,3]
([1]Centre for Astrophysics & Supercomputing, Swinburne University of Technology; [2]Monash e-Research Centre, Monash University; [3]Faculty of Information Technology, Monash University, Clayton, Victoria)
Title: *Colouring redshift and velocity: a novel visual cue to inspect volumetric data in Astronomy*

Observational astronomers have increasing access to three dimensional (3D) data from instruments and facilities like integral field units and radio interferometers. A 3D astronomical data file is typically composed of two angular dimensions along with a spectral or a velocity dimension. Despite the popularity of GPU for general purpose computing, one should not neglect the suitability of GPU for fast scientific visualisation. Despite their growing availability, 3D visualisation techniques like volume rendering tend to be under utilised in astronomy. As most 3D visualisation techniques have been developed for fields of research like medical imaging and fluid dynamics, many transfer functions are not intuitive for astronomical data. Instead, most astronomers tend to visualise 3D data in 2D. Common practices consist of computing "moment maps", that is, representing statistical moments of spatial pixels based on their respective spectral channels. Moments give insight on physical quantities like overall gas distribution (0th moment), gas velocity field (1st moment) and gas velocity dispersion (2nd moment). Inspired by moment maps visualisation, we present novel transfer functions specifically designed to visualise astronomical data with volume rendering methods. By mapping spectral coordinates to color, and mapping voxel scalar or derived quantities to transparency and/or brightness, we provide astronomers intuitive visual cues to explore their data. We discuss how this can be achieved by utilising the parallelism of modern GPUs along with dynamic shading language, letting astronomers comprehensively explore their 3D data at interactive rate.

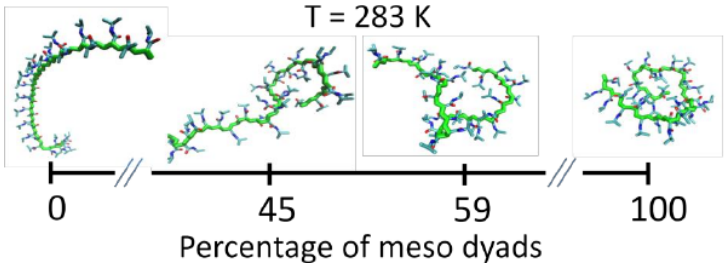
Posters

Author: Ebru Bozdag[1], Daniel Peter[2], Matthieu Lefebvre[3], Lion Krischer[4], Wenjie Lei[3], Youyi Ruan[3], James Smith[3], Dimitri Komatitsch[5], Jeroen Tromp[3], Judith Hill[6], Norbert Podhorski[6], David Pugmire[6] ([1]University of Nice Sophia Antipolis, France (bozdag@geoazur.unice.fr); [2]King Abdullah University of Science and Technology, Saudi Arabia; [3]Princeton University, NJ, USA; [4]University of Munich, Germany; [5]University of Aix-Marseille, France; [6]Oak Ridge National Laboratory, TN, USA)
Title: *Global Adjoint Tomography: Imaging Earth's interior using 3D wave simulations, big data & GPU accelerators*

Performing adjoint tomography, an iterative full-waveform inversion technique, to image Earth's interior at global scale is currently one of the most extreme projects in seismology in terms of its computational requirements and big data that can potentially be assimilated in inversions. There have been successful applications of adjoint tomography based on 3D wave simulations to regional-and continental-scale studies, however, so far has remained a challenge at global scale mainly due to computational requirements. We initiated global adjoint inversions using Cray XK7 *Titan*, a hybrid supercomputer with 18,688 Nvidia Tesla K20X GPU accelerators in addition to 18,688 conventional AMD Opteron 6274 16-core CPUs, housed at Oak Ridge National Laboratory. Synthetic seismograms are calculated using GPU-accelerated spectral-element simulations of global seismic wave propagation using the SPECFEM3D_GLOBE package (Komatitsch & Tromp 2002), accommodating effects due to 3D anelastic crust & mantle structure, topography & bathymetry, ellipticity, rotation, and self- gravitation. Fréchet derivatives are calculated in 3D anelastic models based on an adjoint-state method. Using 253 earthquakes, we have constructed the "first" global adjoint tomography model, GLAD-M15, which is the result of 15 conjugate gradient iterations with transverse isotropy in the upper mantle systematically reducing differences between observed and simulated three- component seismograms. Starting from a 3D model, we simultaneously inverted for crust and mantle eliminating the widely used 'crustal corrections'. Despite of using a small-set of data and performing 15 iterations only, GLAD-M15 clearly features enhancements of well-known slabs, plumes and hotspots. Resolution tests suggest that as a result of our multi-stage smoothing strategy, we are getting close to continental-scale resolution in global inversions in areas with dense coverage such as underneath North America and Europe. While we continue our iterations to construct the next-generation models with more general model parameterisations (i.e., accommodating azimuthal anisotropy, 3D attenuation), we move towards assimilating data from all available earthquakes within the magnitude range $5.5 \leq \mathrm{Mw} \leq 7.0$. To this end, we have optimised our workflow by 1) developing an Adaptive Seismic Data Format (ASDF, Krischer et al. 2016) that satisfies the needs of modern seismology providing efficient I/O and data provenance for reproducibility in science, 2) adopting ASDF to Pyhton-based pre-processing tools (i.e., data processing, data selection and measurements), 3) implementing ADIOS (Liu et al. 2011) format to our computational data (i.e., models, kernels, mesh files) 4) developing post-processing tools (i.e., pre-conditioning and smoothing of gradients, model updates) based on the ADIOS libraries. Meanwhile, in order to stabilize the entire process, we experiment with workflow management systems, such as Pegasus (Deelman et al. 2015). Our ultimate aim is to get ready to harness ORNL's next-generation supercomputer "Summit", an IBM with Power-9 CPUs and NVIDIA Volta GPU accelerators, to be ready by 2018 which will enable us to reduce the shortest period in our global simulations from 17 s to 9 s, and exascale systems will reduce this further to just a few seconds.

Author: Gaio Paradossi and Ester Chiessi (Department of Chemical Sciences and Technologies, University of Rome Tor Vergata)
Title: *Tuning the Hydrophobicity of Poly(N-isopropylacrylamide) by Tacticity: a Molecular Dynamics Simulation Study*

The research of stimuli triggered, artificial micro- and nano-devices identified poly(N- isopropylacrylamide), PNIPAM, as good candidate to obtain soft matrices with thermal responsivity [1]. In water environment this synthetic macromolecule displays a transition from a soluble state, stable at temperatures lower than a critical value, referred to as lower critical solution temperature, LCST, to an insoluble state. Such transition is reversible and implies both intramolecular coil-to- globule chain rearrangement and aggregation between chains, finally leading to phase separation. To move the phase boundary of the PNIPAM-water binary system without introducing hetero residues is feasible by modifying the polymer tacticity, indeed experimental findings show that meso dyads rich PNIPAMs are more hydrophobic as compared to the corresponding atactic polymers, the opposite occurring for racemo dyads rich PNIPAMs [2]. Therefore a small and local change of stereochemistry can adapt the thermal switch conditions of PNIPAM based systems to the specific application. This work aims to achieve a molecular interpretation of the tacticity dependent water affinity of PNIPAM, which is still missing. Atomistic detailed models with different tacticities were built and their behaviour in diluted aqueous solution was simulated both below and above the LCST. The space/time window accessible to this kind of simulations, a ten of nanometres/few hundreds of nanoseconds, respectively, allowed to monitor the coil-to-globule transition of PNIPAM chains with a degree of polymerization corresponding to about two Kuhn segments. The temperature influence on polymer conformation, detected in these models, agrees with the available experimental data and the analysis of the PNIPAM hydration highlights a cooperative pattern of hydrogen bonded water molecules, as forecast by the Tanaka's model [3,4]. The structural alterations of such hydration shell, induced by the different stereochemistry, are the main factor determining the modulation of PNIPAM solubility. Many-core computing allowed for this study and will be further exploited for modelling larger and topologically more complex PNIPAM based systems. **Acknowledgments.** The CINECA award under the ISCRA initiative is acknowledged for the availability of high performance computing resources. The University of Rome Tor Vergata is acknowledged for funding this research within the project AcouGraph (Consolidate the Foundations 2015).



(Color online)

[1] Halperin, A.; Kröger, M.; Winnik, F. M. Angew. Chem. Int. Ed. 2015, 54, 15342-15367.
[2] Nishi, K.; Hiroi, T.; Hashimoto, K.; Fujii, K.; Han, Y.-S.; Kim, T.-H.; Katsumoto, Y.; Shiayama, M. S. Macromolecules 2013, 46, 6225-6232.
[3] Okada, Y.; Tanaka, F. Macromolecules 2005, 38, 4465-4471.
[4] Chiessi, E.; Paradossi, G. J. Phys. Chem. B 2016, 120, 3765-3776.

Author: Simone Coscetti, Silvia Arezzini, Alberto Ciampa, Dario Fabiani, Maria Evelina Fantacci, Alessia Giuliano, Enrico Mazzoni, Alessandra Retico (INFN Pisa)
Title: *GPUs parallel computing exploitation for neuroimaging in the ARIANNA project*

In neuroimaging we are experiencing a fast grow of data and images resolution. So, the computing methods have become increasingly important in this research field. Modern graphical processing units (GPUs) can reduce computational times of the algorithms dedicated to neuroimaging with respect to traditional CPUs, thanks to their high-performance, data-parallel architecture. Today, various applications exist that are able to exploit GPUs parallel computing performance in the neuroimaging fields, such as SPM (Statistical Parametric Mapping) and FreeSurfer. The ultimate goal of the recently started italian project ARIANNA (Ambiente di Ricerca Interdisciplinare per l'Analisi di Neuroimmagini Nell'Autismo – Interdisciplinary research platform for neuroimaging analysis in Autism Spectrum Disorders) is to disentangle the heterogeneity of Autism Spectrum Disorders (ASD) by means of a strategy change in neuroimaging data analysis. The complexity and heterogeneity of ASD require dedicated multivariate analysis technique to get the most from the interrelationships among the many variables that describe affected individuals, spanning from clinical phenotypic characterization and genetic profile to structural and functional brain images. The ARIANNA research team will develop and validate new effective methods to analyse neuroimaging data acquired in multiple sites, and multivariate approaches to handle multimodal information allowing for partially missing data. ARIANNA will overcome the current fragmentation of ASD neuroimaging studies by developing a publicly available data analysis service, which will be easily accessible to Researchers in Psychiatry through the ARIANNA web portal. The ARIANNA service will connect and support the Psychiatric Community, providing the necessary technical expertise in neuroimaging data processing and analysis. By developing effective tools to achieve the stratification of ASD individuals, ARIANNA aims to provide a boost to the ASD research to open the way to early and tailored treatments and thus to improve the ASD people's daily life. In this paper we describe the dedicated computing farm, consisting of four nVidia K80 GPUs, and the storage appliance hosted by the Pisa INFN computing center, that will guarantee the secure data handling and storage, and the access to fast grid/cloud-based computational resources.

Author: Attilio Cucchieri and Tereza Mendes (IFSC-University of São Paulo)
Title: *Preconditioned CG algorithms in Lattice QCD*

Quantum Chromodynamics (QCD) is a quantum field theory with local gauge invariance that describes the strong force in nuclear matter. A unique feature of this force is that the particles that feel it directly (quarks and gluons) are never observed as free particles. This property, known as color confinement, makes it impossible to study QCD analytically-in the limit of small energies, or large spatial separations-and physicists must rely on numerical simulations, performed on supercomputers, using the lattice formulation of QCD. Among the quantities that can be evaluated on the lattice, the long-range behavior of Green's functions may offer crucial insight in our understanding of QCD. Due to the local gauge invariance of QCD, the evalu- ation of Green's functions requires to impose a specific gauge condition, such as the Landau gauge condition. In lattice Landau gauge, the gauge-field con- figurations are transverse and the Landau-gauge Faddeev-Popov (FP) matrix is positive definite. It is believed that this matrix may encode some of the relevant (non-perturbative) aspects of the theory. Moreover, the inverse of the FP matrix enters the evaluation of several QCD Green's functions. The numerical inversion of the FP matrix is rather time consuming, since it is a very large symmetric sparse matrix with an extremely small positive eigenvalue, thus requiring the use of a parallel preconditioned conjugate- gradient (CG) algorithm. Moreover, this inversion has to be done in double precision and for hundreds of different sources, corresponding to different kinematic combinations. At the same time, on the lattice, the smallest mo- mentum is proportional to 1/L, where L is the lattice side. Thus, numerical studies of Green's functions at small momenta require very large lattice vol- umes and the extrapolation of the data to the infinite-volume limit. In this poster we present an update of our study of several preconditioned CG al- gorithms and their implementation (through CUDA) in double and mixed precision for Tesla S1070 and Tesla K20 GPUs. In particular, we report on the performance of the code on multiple GPUs, interconnected by InfiniBand.

Author: Giacomo Fragione (Sapienza Università di Roma)
Title: *High Velocity Stars as Tools for Galactic Astrophysics: Star Clusters, Dark Halo and Planets*

High Velocity Stars (HVSs) are stars moving at velocities so high to require an acceleration mechanism involving binary systems or the presence of a massive central black hole. Since the production of HVSs involve different astrophysical frameworks and phenomena, it is possible to infer information on the environment that originated them. In the case that a star cluster infalls onto a supermassive black hole, the study of HVSs, originated thanks to such mechanism, can suggest the physics and kinematics of its progenitor. This goal can be achieved thanks to the comparison of data from telescopes and the results of gravitational N-body simulations, performed thanks to GPU- parallel codes. Moreover, the study of the proper motions of these fast-moving stars can improve the knowledge of different branches of physics, as the physics of the region near massive black holes and the Galaxy gravitational potential shape, in particular of its dark matter component. Finally, even if the origin of HVSs involve strong gravitational interactions, some of these stars may retain their planetary systems, giving peculiar observational signatures for the upcoming data from Gaia and TESS satellites.

Author: Christoph Hasse (Technical University Dortmund)
Title: *Amplitude analysis of four-body decays using a massively-parallel fitting framework*

GooFit, a GPU-friendly framework for performing maximum-likelihood fits, has been ex- tended in functionality to enable a full time-dependent amplitude analysis of scalar mesons decaying into four final state particles via various combinations of intermediate resonances. Recurring resonances in different amplitudes are recognized and only calculated once to save memory and execution time. As an example, this tool is being used to study the sen- sitivity to the charm mixing parameters in a full time-dependent amplitude analysis of the decay $D^0 \to K^+\pi^-\pi^+\pi^-$ by generating and fitting pseudo experiments. GooFit uses the Thrust library to launch all kernels with a CUDA back-end for nVidia GPUs and an OpenMP back-end for computing nodes with conventional CPUs. The performance of the algorithm will be compared for a variety of supported platforms.

Author: Andreas Herten, Dirk Pleiter, Dirk Brömmel (Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany)
Title: *Accelerating Plasma Physics with GPUs*

JuSPIC is a particle-in-cell (PIC) code, developed in the Simulation Lab for Plasma Physics of the Jülich Supercomputing Centre. The open source code is based on PSC by H. Ruhl, slimmed-down and rewritten in modern Fortran. JuSPIC simulates particles under the influence of electromagnetic fields, using the relativistic Vlasov equation and Maxwell's equations (integrated using the Finite Difference Time Domain scheme). The program uses a regular mesh for the Maxwell fields and the particle charge/current densities. Inside the mesh, quasi-particles with continuous coordinates are modeled via distribution functions. JuSPIC is part of the High-Q club, attesting that it can efficiently scale to the full JUQUEEN supercomputer (currently the #13 on the Top 500 list): 1.8 million threads running on 458 thousand cores can collaboratively compute plasma simulations. Local node-level parallelism is achieved by means of OpenMP, communication between nodes relies on MPI. To leverage the latest generation of supercomputers coming equipped with dedicated accelerator technologies (GPUs and other many-core architectures), JuSPIC is currently being extended. In this poster we present a GPU-accelerated version of the program, making use of different programming models. We show first results of performance studies, comparing OpenACC and CUDA. While OpenACC aims to offer portability and flexibility by means of few changes to the code, the performance of the generated program might suffer in practice. To measure the deficit, the compute- intensive parts of the program are in addition also implemented in CUDA Fortran. To explore scalability properties of the application for static particle distributions on a heterogeneous architecture, we make use of semi-empirical performance models.

Author: Parnian Kianfar and Nabiollah Abolfathi (Amirkabir University of Technology, Department of Biomedical Engineering, Tehran)
Title: *Steered Molecular Dynamic Simulation of Contactin Complex*

Cell adhesion is highly crucial in interactions of cells. The mechanical interactions between a cell and its environment can have an effect on its behavior and functions. Contactin is a cell adhesion molecule and a member of Immunoglobulin Super Family (Ig-SF). Members of Contactin family composed of six immunoglobulin-like (Ig) domains C2 type and four fibronectin type III (FNIII) repeats. Contactin has an important role in the formation, maintenance, and plasticity of neuronal networks. This study investigated the adhesive behavior of the Contactin protein complex under the application of the steered molecular dynamics simulations. Contactin in a complex configuration with the protein tyrosine phosphatase gamma (PTPRG) was selected. Protein-complex was subjected to constant pulling rate. Simulations were carried out using the NAMD program and the CHARMM22 force field. SMD simulations were performed in constant velocity mode with five different loading rates by using NAMD's GPU accelerating feature to speed up the simulations. The force evolution, H-bonds changes and salt bridges were monitored throughout the simulations. The findings showed a wide range of peak forces for detachment of complex, and agreed with the fact that bonds break with almost any level of force in a case that time is not matter.

Author: Daniel Peter[1], Brice Videau[2], Kevin Pouget[2], Dimitri Komatitsch[3] ([1]King Abdullah University of Science and Technology, Saudi Arabia; [2]University of Grenoble, France; [3]University of Aix-Marseille, France;)
Title: *Seismic wave propagation on emerging HPC architectures*

Recent advances in regional- and global-scale seismic inversions move towards full-waveform inversions which require accurate simulations of seismic wave propagation in complex 3D media, providing access to the full 3D seismic wavefields. Increasing resolutions in structural inversions however quickly becomes computationally very expensive. We must therefore exploit both algorithmic advances as well as new high-performance computing (HPC) hardware.

Here we will show that for ground motions simulations and seismic tomography the spectral-element packages SPECFEM3D and SPECFEM3D_GLOBE are well suited to follow new HPC trends. These software packages have been extended to make use of general-purpose hardware accelerators such as graphic processing units (GPUs). By incorporating a code generation tool BOAST into the existing spectral-element code, we use meta-programming of computational kernels and generate optimized source code for both CUDA and OpenCL hardware accelerators. Applications of forward and adjoint seismic wave propagation on CUDA/OpenCL GPUs will be shown, comparing performance and benchmark scaling for different simulation setups.

Author: R. Ammendola[1], A.Biagioni[2], P. Cretaro[2], Stefano Di Lorenzo [4,5], O. Frezza[2], G. Lamanna[3], F. Lo Cicero[2], A. Lonardo[2], M. Martinelli[2], P.S. Paolucci[2], E. Pastorelli[2], R. Piandani[4], L. Pontisso[4], F. Simula[2], M. Sozzi[4,5], P. Valente[2] and P. Vicini[2] ([1]INFN Sezione di Roma - Tor Vergata; [2] INFN Roma - Sapienza; [3]INFN Laboratori Nazionali di Frascati; [4] INFN Pisa; [5]Università di Pisa;)
Title: *Real-time RICH ring reconstruction techniques on GPUs*

In High Energy Physics (HEP) experiments the trigger system is employed for the on-line selection of interesting events, in order to reduce the bandwidth requirements and the disk space needed to collect data. Standard trigger systems rest upon the complexity and the quality of the trigger primitives to take a trigger decision, with limitations arising by latency constraints and possibility to sustain the given rate. Graphics Processing Units (GPUs) have a relevant computing power, so they could permit to implement more selective, high throughput event selection algorithms. A GPU-based Low level trigger for the Ring-imaging Cherenkov (RICH) detector of the CERN's NA62 experiment is being integrated in the experimental setup in a parasitic mode, taking into account that this is an innovative technique. The usage of the NaNet board guarantees the implementation of a low-latency, real-time data transport

mechanism between the board network multi-channel system and the host CPU and GPU accelerators memories. Low level trigger RICH reconstruction is very challenging. But the possibility to use in trigger decisions the information coming from Cherenkov rings would be very useful in building stringent conditions for data selection. We will describe two different multi-rings pattern recognition algorithms based only on geometrical considerations that we developed specifically to exploit the parallel architecture of GPUs. Results obtained during the NA62 2016 data taking will be discussed.

Author: Antonio Ragagnin (Leibniz Supercomputing Centre; Universitäts-Sternwarte München)
Title: *A multi node Barnes Hut solver on GPUs for Gadget3*

Gadget3 is one of the most used high performing parallel codes for cosmological hydrodynamic simulations. It is an N-Body code using a Tree-PM approach to solve for gravity and the hydrodynamics. Well resolved large scale simulations do not fit in the memory of single compute nodes, therefore, Gadget3's top level parallelization layer is based on MPI . A common challenge in porting tree based solver on the GPU is the slow down due to the distributed execution of the tree traversals, which usually are performed for every particle to process. In our approach we run one part of the solver on the GPU and the other on the CPU. For each particle, on the GPU we solve asynchronously the local interactions, while the CPU solves global interactions which requires information from other compute nodes. Additionally, a negligible overhead is required to recombine the results from the GPU and CPU computation parts. We test our approach with a high resolved cosmological hydrodynamic simulation from the Magneticum Projects, i.e. Box4/hr (see www.magneticum.org). The newly implemented gravity computation method on GPU gained a speedup of 1.26 with respect to the standard version. We obtained it by running the code over 8 nodes of the Piz Daint system (http://www.cscs.ch/computers/piz_daint_piz_dora/index.html), each using 1 MPI task.

Author: M. Volpe[1], P. Lanucara[2], S. Lorito[1], D. Melini[1], I. Molinari[3], A. Piatanesi[1], F. Romano[1], R. Tonini[1] ([1]Istituto Nazionale di Geofisica e Vulcanologia (INGV), Roma; [2]CINECA, SuperComputing Applications and Innovation Department - Roma; [3]Department of Earth Sciences, Institute of Geophysics, ETH Zurich)
Title: *Development of a Tsunami Scenario Database for the Assessment of the Probabilistic Tsunami Hazard Exploiting GPU Resources*

After the Indian Ocean 2004 tsunami, a global attention has grown towards the development of risk mitigation strategies. Related initiatives are based on the implementation of tsunami warning systems (TWS) as well as on the elaboration of probabilistic tsunami hazard analyses (PTHA). Both the approaches rely on numerical modeling and are highly computationally demanding. A commonly used strategy to limit the computational cost is to build up pre-calculated databases of tsunami impact scenarios corresponding to unit sources for the sea level elevation, which can be then queried to retrieve the scenario for a given source . With the present work we deal with the creation of a database of tsunami scenarios covering the North-Eastern Atlantic and Mediterranean (NEAM) region. This database is being employed for the assessment of probabilistic tsunami hazard (PTHA) in the framework of the TSUMAPS-NEAM project, funded by EC DG ECHO, which would provide the NEAM region with the first homogeneous PTHA, based on the consensus of a large community. In the future, the same database will be also used for tsunami warning applications. The numerical simulations of elementary scenarios are performed with the HySEA code, developed by the EDANYA Group (Differential Equations, Numerical Analysis and Applications) at the University of Malaga, Spain. It is a non-linear hydrostatic shallow-water model that implements a mixed finite difference/finite volume method. The code is written in CUDA and supports multi-GPU environments. Simulations are carried out both on an internal INGV cluster based on nVidia K20 accelerators (3GPU/node) and on the CINECA supercomputer Galileo (actually being the number 151 in the Top500 list) equipped with 2 nVidia K80 accelerators (corresponding to 4GPU K40/node). Due to the great number of simulations, we are able to compare the performances of the two systems and measure the scalability of the HySEA code.