Program

Program is also available in Google Calendar [ical]

Conference proceedings are available on Springer website.

Recordings available on YouTube.

Show abstracts

WEDNESDAY 26.08.2020

14:00 - 14:30 — **Opening**

14:30 - 15:30

Multicore and Manycore Parallelism (A)

Chairs: Witold Rudnicki [Slack channel]

NVPhTM: An Efficient Phase-Based Transactional System for Non-Volatile Memory
Alexandro Baldassin, Rafael Murari, João Paulo Carvalho, Guido Araujo, Daniel Castro, João Barreto and Paolo Romano
Download paper from Springer LNCS. [video]

Non-Volatile Memory (NVM) is an emerging memory technology aimed to eliminate the gap between main memory and stable storage. Nevertheless, today's programs will not readily benefit from NVM because crash failures may render the program in an unrecoverable and inconsistent state. In this context, the use of durable transactions has been proposed so as to ease the adoption of NVM. It leverages on the well-know semantics of database transactions to simplify the task of programming NVM systems. This is achieved by logging NVM writes using software (SW) or hardware (HW) transaction primitives. Although SW transactions are flexible and unbounded, they may significantly hurt the performance of short-lived transactions. On the other hand, HW transactional memories provide low-overhead but are resource-constrained. In this paper we present NVPhTM, a transactional system for NVM that delivers the best out of both HW and SW transactions by dynamically selecting the best execution mode according to the application's characteristics. NVPhTM is comprised of a set of heuristics to guide online phase transition. Furthermore, a careful design of the phase transition step is devised to guarantee persistency when transitioning between HW and SW phases. To the best of our knowledge, NVPhTM is the first phase-based system to provide durable transactions. Experimental results with the STAMP benchmark show that the proposed heuristics are efficient in guiding phase transitions with low overhead. In particular, the NVM-aware heuristics provided an average speedup of up to 10\% when compared to a system using NVM-oblivious heuristics, with only 1\% of transition overhead in the worst case.

Enhancing Resource Management through Prediction-based Policies
Antoni Navarro, Arthur F. Lorenzon, Eduard Ayguadé and Vicenç Beltran
Download paper from Springer LNCS. [video]

Task-based programming models are emerging as a promising alternative to make the most of multi-/many-core systems. These programming models rely on runtime systems, and their goal is to improve application performance by properly scheduling application tasks to cores. Additionally, these runtime systems offer policies to cope with application phases that lack in parallelism to fill all cores. However, these policies are usually static and favor either performance or energy efficiency. In this paper, we have extended a task-based runtime system with a lightweight monitoring and prediction infrastructure that dynamically predicts the optimal number of cores required for each application phase, thus improving both performance and energy efficiency. Through the execution of several benchmarks in multi-/many-core systems, we show that our prediction-based policies have competitive performance while improving energy efficiency when compared to state of the art policies.

Accelerating Overlapping Community Detection: Performance Tuning a Stochastic Gradient Markov Chain Monte Carlo Algorithm
Ismail Elhelw, Rutger Hofman and Henri Bal
Download paper from Springer LNCS. [video]

Building efficient algorithms for data-intensive problems requires deep analysis of data access patterns. Random data access patterns exacerbate this process. In this paper, we discuss accelerating a randomized data-intensive machine learning algorithm using multi-core CPUs and several GPUs. A thorough analysis of the algorithm's data dependencies enabled a 75% reduction in its memory footprint. We created custom compute kernels via code generation to identify the optimal set of data placement and computational optimizations per compute device. An empirical evaluation shows up to 245x speedups compared to an optimized sequential version. Another result from this evaluation is that achieving peak performance does not always match intuition: e.g., depending on the GPU architecture, vectorization may increase or hamper performance.

Cluster, Cloud and Edge Computing (B)

Chairs: Paweł Czarnul [Slack channel]

TorqueDB: Distributed Querying of Time-series Data from Edge-local Storage
Dhruv Garg, Prathik Shirolkar, Anshu Shukla and Yogesh Simmhan
Download paper from Springer LNCS. [video]

The rapid growth in edge computing devices as part of Internet of Things (IoT) allows real-time access to time-series data from 1000's of sensors. Such observations are queried to optimize the health of the infrastructure. Recently, edge-local storage are helping retain data on the edge rather than move them centrally to the cloud. However, such systems do not support flexible querying over the data spread across 10--100's of devices. There is also a lack of distributed time-series databases that can run on the edge devices. Here, we propose TorqueDB, a distributed query engine over time-series data that operates on edge and fog resources. TorqueDB leverages our prior work on ElfStore, a distributed edge-local file store, and InfluxDB, a time-series database, to enable temporal queries to be decomposed and executed across multiple fog and edge devices. Interestingly, we move data into InfluxDB on-demand while retaining the durable data within ElfStore for use by other applications. We also design a cost model that maximizes parallel movement and execution of the queries across resources, and utilizes caching. Our experiments on a real edge, fog and cloud deployment show that TorqueDB performs comparable to InfluxDB on a cloud VM for a smart city query workload, but without the associated costs.

Data-Centric Distributed Computing on Networks of Mobile Devices
Pedro Sanches, João A. Silva, António Teófilo and Hervé Paulino
Download paper from Springer LNCS. [video]

In the last few years we have seen a significant increase both in the number and capabilities of mobile devices, as well as in the num- ber of applications that need more and more computing and storage re- sources. Currently, in order to deal with this growing need for resources, applications make use of cloud services. This brings some problems: high latencies, considerable use of energy and bandwidth, and the unavail- ability of connectivity infrastructures. Given this context, for some ap- plications it makes sense to do part, or all, of the computing locally on the mobile devices themselves. In this paper we present Oregano, a framework for distributed computing on mobile devices. Oregano is capable of processing sets or streams of data generated on mobile de- vice networks, without requiring centralized services. Contrary to the current state of the art, where computing and data are sent to worker mobile devices, our Oregano performs the computation where the data is located, significantly reducing the amount of data exchanged.

WPSP: a multi-correlated weighted policy for VM selection and migration for Cloud computing
Sergi Vila Almenara, Josep Lluis Lerida, Fernando Cores, Fernando Guirado and Fabio Verdi
Download paper from Springer LNCS. [video]

Using virtualization, cloud environments satisfy dynamically the computational resource necessities of the user. The dynamic use of the resources determines the demand of working hosts. Through virtual machine (VM) migrations, datacenters perform load balancing to optimise the resource usage and solve saturation. In this work, a policy, named as WPSP (Weighted Pearson Selection Policy), is implemented to choose which virtual machines are more suitable to be migrated. The policy evaluates, for each VM, both the CPU load and the Network traffic influence on the assigned host. The corresponding Pearson correlation coefficients are calculated for each one of the VMs and then weighted in order to provide a relationship between the values and the host behaviour. The main goal is to clearly identify and then migrate the VMs that are responsible of the Host saturation but also considering their communications. Using the CloudSim simulator, the policy is compared with the rest of heuristic techniques in the literature, resulting in a reduction of 85% in the number of migrations, and thus reducing the use of bandwidth (6%), network saturation (24%) and over-saturated hosts (50%). Additionally, it is presented an improved VM allocation technique to reduce the distance the VMs must travel in order to be migrated, obtaining an average reduction of 43% in the quantity of migrated data.

15:30 - 16:00 — **Break**

16:00 - 17:20

Support Tools and Environments (A)

Chairs: Bartosz Baliś [Slack channel]

Skipping Non-essential Instructions Makes Data-dependence Profiling Faster
Nicolas Morew, Mohammad Norouzi, Ali Jannesari and Felix Wolf
Download paper from Springer LNCS. [video]

Data-dependence profiling is a dynamic program-analysis technique to discover potential parallelism in sequential programs. Unlike purely static analysis, which may overestimate the number of dependences because it does not know many pointers values and array indices at compile time, profiling has the advantage of recording data dependences that actually occur at runtime. But it has the disadvantage of significantly slowing down program execution, often by a factor of 100. In our earlier work, we lowered the overhead of data-dependence profiling by excluding polyhedral loops, which can be handled statically using certain compilers. However, neither does every program contain polyhedral loops, nor are statically identifiable dependences restricted to such loops. In this paper, we introduce an orthogonal approach, focusing on data dependences between accesses to scalar variables - across the entire program, inside and outside loops. We first analyze the program statically and identify memory-access instructions that create data dependences that would appear in any execution of these instructions. Then, we exclude these instructions from instrumentation, allowing the profiler to skip them at runtime and avoid the associated overhead. We evaluate our approach with 49 benchmarks from three benchmark suites. We improved the profiling time of all programs by at least 38%, with a median reduction of 61% across all the benchmarks.

A toolchain to verify the parallelization of OmpSs-2 applications
Simone Economo, Sara Royuela Alcázar, Eduard Ayguadé Parra and Vicenç Beltran Querol
Download paper from Springer LNCS. [video]

Programming models for task-based parallelization based on compile-time directives are very effective at uncovering the parallelism available in HPC applications. Despite that, the process of correctly annotating complex applications is error-prone and may hinder the general adoption of these models. In this paper, we target the OmpSs-2 programming model and present a novel toolchain able to detect parallelization errors coming from non-compliant OmpSs-2 applications. Our toolchain verifies the compliance with the OmpSs-2 programming model using local task analysis to deal with each task separately, and structural induction to extend the analysis to the whole program. To improve the effectiveness of our tools, we also introduce some ad-hoc verification annotations, which can be used manually or automatically to disable the analysis of specific code regions. Experiments run on a sample of representative kernels and applications show that our toolchain can be successfully used to verify the parallelization of complex real-world applications.

High Performance Architectures and Compilers (A)

Chairs: Bartosz Baliś [Slack channel]

Modelling Standard and Randomized Slimmed Folded Clos Networks
Cristóbal Camarero, Carmen Martinez, Ramon Beivide and Javier Corral
Download paper from Springer LNCS. [video]

Fat-trees (FTs) are widely known topologies that, among other advantages, provide full bisection bandwidth. However, many implementations of FTs are made slimmed to cheapen the infrastructure, since most applications do not make use of this full bisection bandwidth. In this paper Extended Generalized Random Folded Clos (XGRFC) interconnection networks are introduced as cost-efficient alternatives to Extended Generalized Fat Trees (XGFT), which is a widely used topological description for slimmed FTs. This is proved both by obtaining a theoretical model of the performance and evaluating it using simulation. Among the results, it is shown that a XGRFC is able to connect 20k servers with 27% less routers than the corresponding XGFT and still providing the same performance under uniform traffic.

OmpMemOpti: Optimized Memory Movement for Heterogeneous Computing
Prithayan Barua, Jisheng Zhao and Vivek Sarkar
Download paper from Springer LNCS. [video]

The fast development of acceleration architectures and applications has made heterogeneous computing the norm for high-performance computing. The cost of high volume data movement to the accelerators is an important bottleneck both in terms of application performance and developer productivity. Memory management is still a manual task performed tediously by expert programmers. In this paper, we develop a compiler analysis to automate memory management for heterogeneous computing. We propose an optimization framework that casts the problem of detection and removal of redundant data movements into a partial redundancy elimination (PRE) problem and applies the lazy code motion technique to optimize it. We chose OpenMP as the underlying parallel programming model and implemented our optimization framework in the LLVM toolchain. We evaluated it with ten benchmarks and obtained a geometric speedup of 2.3$\times$, and reduced on average 50\% of the total bytes transferred between the host-GPU.

Scheduling and Load Balancing (B)

Chairs: Joanna Berlińska [Slack channel]

Parallel Scheduling of Data-Intensive Tasks
Xiao Meng and Lukasz Golab
Download paper from Springer LNCS. [video]

Workloads with precedence constraints due to data dependencies are common in various applications. These workloads can be represented as directed acyclic graphs (DAG), and are often data-intensive, meaning that data loading costs are the dominant factor and thus cache misses should be minimized. We address the problem of parallel scheduling of a DAG of data-intensive tasks to minimize makespan. To do so, we propose greedy online scheduling algorithms that take load balancing, data dependencies, and data locality into account. Simulations and an experimental evaluation using an Apache Spark cluster demonstrate the advantages of our solutions.

A Makespan Lower Bound for the Scheduling of the Tiled Cholesky Factorization based on ALAP scheduling
Olivier Beaumont, Julien Langou, Willy Quach and Alena Shilova
Download paper from Springer LNCS. [video]

Due to the advent of multicore architectures and massive parallelism, the tiled Cholesky factorization algorithm has recently received plenty of attention and is often referenced by practitioners as a case study. It is also implemented in mainstream dense linear algebra libraries and is used as a testbed for runtime systems. However, we note that theoretical study of the parallelism of this algorithm is currently lacking. In this paper, we present new theoretical results about the tiled Cholesky factorization in the context of a parallel homogeneous model without communication costs. Based on the relative costs of involved kernels, we prove that only two different situations must be considered, typically corresponding to CPU and GPU costs. By a careful analysis on the number of tasks of each type that run simultaneously in the ALAP (As Late As Possible) schedule without resource limitation, we are able to determine precisely the number of busy processors at any time (as degree 2 polynomials). We then use this information to find a closed form formula for the minimum time to schedule a tiled Cholesky factorization of size n on p processors. We show that this bound outperforms classical bounds from the literature. We also prove that ALAP(p), an ALAP-based schedule where the number of resources is limited to p, has a makespan extremely close to the lower bound, thus proving both the effectiveness of ALAP(p) schedule and of the lower bound on the makespan.

Optimal GPU-CPU Offloading Strategies for Deep Neural Network Training
Olivier Beaumont, Lionel Eyraud-Dubois and Alena Shilova
Download paper from Springer LNCS. [video]

Training Deep Neural Networks is known to be an expensive operation, both in terms of computational cost and memory load. Indeed, during training, all intermediate layer outputs (called activations) computed during the forward phase must be stored until the corresponding gradient has been computed in the backward phase. These memory requirements sometimes prevent to consider larger batch sizes and deeper networks, so that they can limit both convergence speed and accuracy. Recent works have proposed to offload some of the computed forward activations from the memory of the GPU to the memory of the CPU. This requires to determine which activations should be offloaded and when these transfers from and to the memory of the GPU should take place. We prove that this problem is NP-hard in the strong sense, and we propose two heuristics based on relaxations of the problem. We perform extensive experimental evaluation on standard Deep Neural Networks. We compare the performance of our heuristics against previous approaches from the literature, showing that they achieve much better performance in a wide variety of situations.

Improving mapping for sparse direct solvers: A trade-off between data locality and load balancing
Changjiang Gou, Ali Al Zoobi, Anne Benoit, Mathieu Faverge, Loris Marchal, Grégoire Pichon and Pierre Ramet
Download paper from Springer LNCS. [video]

In order to express parallelism, parallel sparse direct solvers take advantage of the elimination tree to exhibit tree-shaped task graphs, where nodes represent computational tasks and edges represent data dependencies. One of the pre-processing stages of sparse direct solvers consists of mapping computational resources (processors) to these tasks. The objective is to minimize the factorization time by exhibiting good data locality and load balancing. The proportional mapping technique is a widely used approach to solve this resource-allocation problem. It achieves good data locality by assigning the same processors to large parts of the elimination tree. However, it may limit load balancing in some cases. In this paper, we propose a dynamic mapping algorithm based on proportional mapping. This new approach relaxes the data locality criterion to improve load balancing. In order to validate the newly introduced method, we perform extensive experiments on the PASTIX sparse direct solver. It demonstrates that our algorithm enables better static scheduling of the numerical factorization while keeping good data locality.

17:20 - 17:30 — **Break**

17:30 - 18:20 — **Keynote Ewa Deelman (A)**

Automating Science Workflows:Challenges and Opportunities

Chair: Rizos Sakellariou
Abstract available on keynotes page

[Slack channel]

18:20 - 19:00 — **Welcome reception**

THURSDAY 27.08.2020

13:00 - 14:30 — **industry: Huawei (A)**

Details available on "industry: Huawei" page

[Slack channel]

14:30 - 15:00 — **Best paper (A)**

Chairs: Morris Riedel [Slack channel]

Maximizing I/O Bandwidth for Reverse Time Migration on Heterogeneous Large-Scale Systems
Tariq Alturkestani, Hatem Ltaief and David Keyes
Download paper from Springer LNCS. [video]

Reverse Time Migration (RTM) is an important scientific application for oil and gas exploration. The 3D RTM simulation generates terabytes of intermediate data that does not fit in main memory. In particular, RTM has two successive computational phases, i.e., the forward modeling and the backward propagation, that necessitate to write and then to read the state of the computed solution grid at specific time steps of the time integration. Advances in memory architecture have made it feasible and affordable to integrate hierarchical storage media on large-scale systems, starting from the traditional Parallel File Systems (PFS) to intermediate fast disk technologies (e.g., node-local and remote-shared Burst Buffer) and up to CPU main memory. To address the trend of heterogeneous HPC systems deployment, we introduce an extension to our Multilayer Buffer System (MLBS) framework to further maximize RTM I/O bandwidth in presence of GPU hardware accelerators. The main idea is to leverage the GPU's High Bandwidth Memory (HBM) as an additional storage media layer. The objective of MLBS is ultimately to hide the application's I/O overhead by enabling a buffering mechanism operating across all the hierarchical storage media layers. MLBS is therefore able to sustain the I/O bandwidth at each storage media layer. By asynchronously performing expensive I/O operations and creating opportunities for overlapping data motion with computations, MLBS may transform the original I/O bound behavior of the RTM application into a compute-bound regime. In fact, the prefetching strategy of MLBS allows the RTM application to believe that it has access to a larger memory capacity on the GPU, while transparently performing the necessary housekeeping across the storage layers. We demonstrate the effectiveness of MLBS on the Summit supercomputer using 2048 compute nodes equipped with a total of 12288 GPUs by achieving up to 1.4X performance speedup compared to the reference PFS-based RTM implementation for large 3D solution grid.

15:00 - 15:30 — **Best artifact (A)**

Chairs: Maciej Szpindler

A Prediction Framework for Fast Sparse Triangular Solves
Najeeb Ahmad, Buse Yilmaz and Didem Unat
Download paper from Springer LNCS. [video]

Sparse triangular solve (SpTRSV) is an important linear algebra kernel, finding extensive uses in numerical and scientific computing. The parallel implementation of SpTRSV is a challenging task due to the sequential nature of the steps involved. This makes it, in many cases, one of the most time-consuming operations in an application. Many approaches for efficient SpTRSV on CPU and GPU systems have been proposed in the literature. However, no single implementation or platform (CPU or GPU) gives the fastest solution for all input sparse matrices. In this work, we propose a machine learning-based framework to predict the SpTRSV implementation giving the fastest execution time for a given sparse matrix based on its structural features. The framework is tested with six SpTRSV implementations on a state-of-the-art CPU-GPU machine (Intel Xeon Gold CPU, NVIDIA V100 GPU). Experimental results, with 998 matrices taken from the SuiteSparse Matrix Collection, show the classifier prediction accuracy of 87% for the fastest SpTRSV algorithm for a given input matrix. Predicted SpTRSV implementations achieve average speedups (harmonic mean) in the range of 1.2-2.4x against the six SpTRSV implementations used in the evaluation.

15:30 - 15:40 — **Break**

15:40 - 16:40

Data Management, Analytics and Machine Learning (A)

Chairs: Morris Riedel [Slack channel]

Accelerating Deep Learning Inference with Cross-Layer Data Reuse on GPUs
Xueying Wang, Guangli Li, Xiao Dong, Jiansong Li, Lei Liu and Xiaobing Feng
Download paper from Springer LNCS. [video]

Accelerating the deep learning inference is very important for real-time applications. In this paper, we propose a novel method to fuse the layers of convolutional neural networks (CNNs) on Graphics Processing Units (GPUs), which applies data reuse analysis and access optimization in different levels of the memory hierarchy. To achieve the balance between computation and memory access, we explore the fu- sion opportunities in the CNN computation graph and propose three fusion modes of convolutional neural networks: straight, merge and split. Then, an approach for generating efficient fused code is designed, which goes deeper in multi-level memory usage for cross-layer data reuse. The effectiveness of our method is evaluated with the structures from state- of-the-art CNNs on two different GPU platforms, NVIDIA TITAN Xp and Tesla P4. The experiments show that the average speedup is 2.02 × on representative structures of CNNs, and 1.57× on end-to-end inference of SqueezeNet.

Optimizing FFT-based convolution on ARMv8 multi-core CPUs
Qinglin Wang, Dongsheng Li, Xiandong Huang, Siqi Shen, Songzhu Mei and Jie Liu
Download paper from Springer LNCS. [video]

Convolutional Neural Networks (CNNs) are widely applied in various machine learning applications and very time-consuming. Most of CNNs' execution time is consumed by convolutional layers. A common approach to implementing convolutions is the FFT-based one, which can reduce the arithmetic complexity of convolutions without losing too much precision. As the performance of ARMv8 multi-core CPUs improves, they can also be utilized to perform CNNs like Intel X86 CPUs. In this paper, we present a new parallel FFT-based convolution implementation on ARMv8 multi-core CPUs. The implementation makes efficient use of ARMv8 multi-core CPUs through a series of computation and memory optimizations. The experiment results on two ARMv8 multi-core CPUs demonstrate that our new implementation gives much better performance than two existing approaches in most cases.

Distributed Fine-Grained Traffic Speed Prediction for Large-Scale Transportation Networks based on Automatic LSTM Customization and Sharing
Ming-Chang Lee, Jia-Chun Lin and Ernst Gunnar Gran
Download paper from Springer LNCS. [video]

Short-term traffic speed prediction has been an important research topic in the past decade, and many approaches have been introduced. However, providing fine-grained, accurate, and efficient traffic-speed prediction for large-scale transportation networks where numerous traffic detectors are deployed has not been well studied. In this paper, we propose DistPre, which is a distributed fine-grained traffic speed prediction scheme for large-scale transportation networks. To achieve fine-grained and accurate traffic-speed prediction, DistPre customizes a Long Short-Term Memory (LSTM) model with an appropriate hyperparameter configuration for a detector. To make such customization process efficient and applicable for large-scale transportation networks, DistPre conducts LSTM customization on a cluster of computation nodes and allows any trained LSTM model to be shared between different detectors. If a detector observes a similar traffic pattern to another one, DistPre directly shares the existing LSTM model between the two detectors rather than customizing an LSTM model per detector. Experiments based on traffic data collected from freeway I5-N in California are conducted to evaluate the performance of DistPre. The results demonstrate that DistPre provides time-efficient LSTM customization and accurate fine-grained traffic-speed prediction for large-scale transportation networks.

Accelerator Computing (B)

Chairs: Łukasz Szustak [Slack channel]

cuDTW++: Ultra-Fast Dynamic Time Warping on CUDA-enabled GPUs
Bertil Schmidt and Christian Hundt
Download paper from Springer LNCS. [video]

Dynamic Time Warping (DTW) is a widely used distance measure in the field of time series data mining. However, calculation of DTW scores is compute-intensive since the complexity is quadratic in terms of time series lengths. This renders important data mining tasks computationally expensive even for moderate query lengths and database sizes. Previous solutions to accelerate DTW on GPUs are not able to fully exploit their compute performance due to inefficient memory access schemes. In this paper, we introduce a novel parallelization strategy to drastically speed-up DTW on CUDA-enabled GPUs based on using low latency warp intrinsics for fast inter-thread communication. We show that our CUDA parallelization (cuDTW++) is able to achieve over 90% of the theoretical peak performance of modern Volta-based GPUs, thereby clearly outperforming the previously fastest CUDA implementation (cudaDTW) by over one order-of-magnitude. Furthermore, cuDTW++ achieves two-to-three orders-of-magnitude speedup over the state-of-the-art CPU program UCR-Suite for subseqeunce search of ECG signals. We plan to make cuDTW++ publicly available upon acceptance of this paper.

Heterogeneous CPU+iGPU Processing for Efficient Epistasis Detection
Rafael Campos, Diogo Marques, Sergio Santander-Jiménez, Leonel Sousa and Aleksandar Ilic
Download paper from Springer LNCS. [video]

Epistasis detection represents a fundamental problem in bio-medicine to understand the reasons for occurrence of complex phenotypic traits (diseases) across a population of individuals. Exhaustively examining all possible interactions of multiple Single-Nucleotide Polymorphisms provides the most reliable way to identify accurate solutions, but it is both computationally and memory intensive task. To tackle this challenge, this work proposes a modular and self-adaptive framework for high-performance and energy-efficient epistasis analysis on modern tightly-coupled heterogeneous platforms composed of multi-core CPUs and integrated GPUs. To fully exploit the capabilities of these systems, the proposed framework incorporates both task- and data-parallel approaches specifically tailored to enhance single and multi-objective epistasis detection on each device architecture, along with allowing efficient collaborative execution across all devices. The experimental results show the ability of the proposed framework to handle the heterogeneity of an Intel CPU+iGPU system, achieving performance and energy-efficiency gains of up to 5x and 6x in different parallel execution scenarios.

SYCL-Bench: A Versatile Cross-Platform Benchmark Suite for Heterogeneous Computing
Sohan Lal, Aksel Alpay, Philip Salzmann, Biagio Cosenza, Alexander Hirsch, Nicolai Stawinoga, Peter Thoman, Thomas Fahringer and Vincent Heuveline
Download paper from Springer LNCS. [video]

The SYCL standard promises to enable high productivity in heterogeneous programming of a broad range of parallel devices, including multicore CPUs, GPUs, and FPGAs. Its modern and expressive C++ API design, as well as flexible task graph execution model give rise to ample optimization opportunities at run-time, such as the overlapping of data transfers and kernel execution. However, it is not clear which of the existing SYCL implementations perform such scheduling optimizations, and to what extent. Furthermore, SYCL's high level of abstraction may raise concerns about sacrificing performance for ease of use. Benchmarks are required to accurately assess the performance behavior of high-level programming models such as SYCL. To this end, we present SYCL-Bench, a versatile benchmark suite for device characterization and runtime benchmarking, written in SYCL. We experimentally demonstrate the effectiveness of SYCL-Bench by performing device characterization of the NVIDIA TITAN X GPU, and by evaluating the efficiency of the hipSYCL and ComputeCpp SYCL implementations.

16:40 - 17:00 — **Break**

17:00 - 18:00 — **Keynote Geoffrey Fox (A)**

Advancing Science with Deep Learning, HPC, Data Benchmarks and Data Engineering

Chair: Christan Lengauer
Abstract available on keynotes page

[Slack channel]

18:00 - 19:10

Parallel Numerical Methods and Applications (A)

Chairs: Hatem Ltaief [Slack channel]

Efficient Ephemeris Models for Spacecraft Trajectory Simulations on GPUs
Fabian Schrammel, Florian Renk, Arya Mazaheri and Felix Wolf
Download paper from Springer LNCS. [video]

When a spacecraft is released into space, its start condition and future trajectory in terms of position and speed cannot be precisely predicted. To ensure that the object does not violate space debris mitigation or planetary protection standards, such that it causes potential damage or contamination of celestial bodies, spacecraft-mission designers conduct a multitude of simulations to verify the validity of the set of all probable trajectories. Such simulations are usually independent, making them a perfect match for parallelization. The European Space Agency (ESA) developed a GPU-based simulator for the exact purpose and achieved reasonable speedups in comparison with the established multi-threaded CPU version. However, we noticed that the performance starts to degrade as the spacecraft trajectories diverge in time. Our empirical analysis using GPU profilers showed that the application suffers from poor data locality and high memory traffic. In this paper, we propose an alternative data layout, which increases data locality within thread blocks. Furthermore, we introduce alternative model configurations that lower both algorithmic effort and the number of memory requests without violating accuracy requirements. Our experiments show that our method is able to achieve speedups of up to 2.6x.

Multiprecision block-Jacobi for Iterative Triangular Solves
Fritz Goebel, Hartwig Anzt, Terry Cojean, Goran Flegar and Enrique S. Quintana-Orti
Download paper from Springer LNCS. [video]

Recent research efforts have shown that Jacobi and block- Jacobi relaxation methods can be used as an effective and highly parallel approach for the solution of sparse triangular linear systems arising in the application of ILU-type preconditioners. Simultaneously, a few orthogonal (independent) works have focused on designing efficient high performance adaptive-precision block-Jacobi preconditioning (block-diagonal scaling), in the context of the iterative solution of sparse linear systems, on manycore architectures. In this paper, we bridge the gap between relaxation methods based on regular splittings and preconditioners by demonstrating that iterative refinement can be leveraged to construct a relaxation method from the preconditioner. In addition, we exploit this insight to construct a highly-efficient sparse triangular system solver for graphics processors that combines iterative refinement with the block- Jacobi preconditioner available in the Ginkgo library.

Parallel Finite Cell Method with Adaptive Geometric Multigrid
S. Saberi, A. Vogel and G. Meschke
Download paper from Springer LNCS. [video]

The generation of appropriate computational meshes in the context of numerical methods for partial differential equations is technical and laborious and has motivated a class of advanced discretization methods commonly referred to as unfitted finite elements. To this end, the finite cell method (FCM) combines high-order FEM, adaptive quadrature integration and weak imposition of boundary conditions to embed a physical domain into a structured background mesh. While unfortunate cut configurations in unfitted finite element methods lead to severely ill-conditioned system matrices that pose challenges to iterative solvers, such methods allow for optimized data patterns and for a scalable implementation. In this work, we employ linear octrees for handling the FCM discretization that allow for parallel scalability, adaptive refinement and efficient computation on the commonly regular background grid. We present a parallel adaptive geometric multigrid with Schwarz smoothers for the solution of the resultant system of the Laplace operator. We focus on exploiting the hierarchical nature of space tree data structures for the generation of the required multigrid spaces and discuss scalable and robust extension of the methods across process interfaces. We present both weak and strong scaling of our implementation up to a billion degrees of freedom on distributed-memory clusters.

Parallel and Distributed Programming, Interfaces, and Languages (B)

Chairs: Phil Trinder [Slack channel]

Managing Failures in task-based parallel workflows in distributed computing environments
Jorge Ejarque, Marta Bertran, Javier Álvarez Cid-Fuentes, Javier Conejero and Rosa M. Badia
Download paper from Springer LNCS. [video]

Current scientific workflows are large and complex. They normally perform thousands of simulation whose results combined with searching and data analytics algorithms, in order to infer new knowledge, generate a very large amount of data. To this end, workflows are composed by large amounts of tasks from which is very likely that, at least, one of them fails. Most of the work done about failure management in workflow managers and runtimes focuses on recovering from failures caused by resources (retrying or resubmitting the failed computation in other resources, etc.) However, some of these failures can be caused by the application itself (corrupted data, algorithms which are not converging, etc.), and these fault tolerance mechanisms are not sufficient to perform a successful workflow execution. In these cases, the developer has to add some code in their applications to prevent and manage the possible failures. In this paper, we propose a simple interface and a set of transparent runtime mechanism to simplify how scientists deal with application-based failures in task-based parallel workflows. We have validated our proposal with a set of e-science use cases to show the benefits of the proposed interface and mechanism in terms of programming productivity and performance.

Accelerating Nested Data Parallelism: Preserving Regularity
Lars B. van den Haak, Trevor L. McDonell, Gabriele K. Keller and Ivo Gabe de Wolff
Download paper from Springer LNCS. [video]

Irregular nested data-parallelism is a powerful programming model which enables the expression of a large class of parallel algorithms. However, it is notoriously difficult to compile such programs to efficient code for modern parallel architectures. Regular data-parallelism, on the other hand, is much easier to compile to efficient code, but too restricted to express some problems conveniently or in a manner to exploit the full parallelism. We extend the regular data-parallel programming model to allow for the parallel execution of array level conditionals and iterations over irregular nested structures. We present two novel static analyses to optimise the code generation to reduce the costs of this more powerful irregular model. We present benchmarks to support our claim that these extensions are effective as well as feasible, as they enable to exploit the full parallelism of an important class of algorithms, and together with our optimisations lead to an improvement in absolute performance over an implementation limited to exploiting only regular parallelism.

Using Dynamic Broadcasts to improve Task-Based Runtime Performances
Alexandre Denis, Emmanuel Jeannot, Philippe Swartvagher and Samuel Thibault
Download paper from Springer LNCS. [video]

Task-based runtimes have emerged in the HPC world to take benefit from the computation power of heterogeneous supercomputers and to achieve scalability. One of the main bottlenecks for scalability is the communication layer. Some task-based algorithms need to send the same data to multiple nodes. To optimize this communication pattern, libraries propose dedicated routines, such as MPI_Bcast. However, MPI_Bcast requirements do not fit well with the constraints of task-based runtime systems: it must be performed simultaneously by all involved nodes, and these must know each other, which is not possible when each node runs a task scheduler not synchronized with others. In this paper, we propose a new approach, called dynamic collectives to overcome these constraints. The broadcast communication pattern required by the task-based algorithm is detected automatically, then the broadcasting algorithm relies on active messages and source routing, so that participating nodes do not need to know each other and don't need to synchronize. Receiver receives data the same way as it receives point-to-point communication, without having to know it arrives through a collective. We have implemented the algorithm in the StarPU runtime system using the NewMadeleine communication library. We performed benchmarks using the Cholesky factorization that is known to use broadcasts and observed a sensible improvement of its performance.

A Compression-Based Design for Higher Throughput in a Lock-Free Hash Map
Pedro Moreno, Miguel Areias and Ricardo Rocha
Download paper from Springer LNCS. [video]

Lock-free implementation techniques are known to improve the overall throughput of concurrent data structures. A hash map is an important data structure used to organize information that must be accessed frequently. A key role of a hash map is the ability to balance workloads by dynamically adjusting its internal data structures in order to provide the fastest possible access to the information. This work extends a previous lock-free hash-trie map design to also support \emph{lock-free compression}. The main goal is to significantly reduce the depth of the internal hash levels within the hash map, in order to minimize cache misses and increase the overall throughput. To materialize our design, we redesigned the existent search, insert, remove and expand operations in order to maintain the lock-freedom property of the whole design. Experimental results show that lock-free compression effectively improves the search operation and, in doing so, it outperforms the previous design, which was already quite competitive.

FRIDAY 28.08.2020

13:00 - 14:45 — **industry: IBM (A)**

[Slack channel]

14:45 - 15:00 — **Break**

15:00 - 15:50 — **Keynote Piotr Sankowski (A)**

Breaking the PRAM O(log n) complexity bounds on MPC

Abstract available on keynotes page

[Slack channel]

15:50 - 16:00 — **Break**

16:00 - 17:00

Theory and Algorithms for Parallel and Distributed Processing (A)

Chairs: Marek Klonowski [Slack channel]

On the Power of Randomization in Distributed Algorithms in Dynamic Networks with Adaptive Adversaries
Irvan Jahja, Haifeng Yu and Ruomu Hou
Download paper from Springer LNCS. [video]

This paper investigates the power of randomization in general distributed algorithms in dynamic networks where the network's topology may evolve over time, as determined by some adaptive adversary. In such a context, randomization may help algorithms to better deal with i) "bad" inputs to the algorithm, and ii) "bad" evolving topologies generated by the adaptive adversary. We prove that randomness offers limited power to better deal with "bad" evolving topologies. We define a simple notion of prophetic adversary for determining the evolving topologies. Such an adversary accurately predicts all randomness in the algorithm beforehand, and hence the randomness will be useless against "bad" prophetic adversaries. We prove that subject to some mild conditions, we can always convert a randomized algorithm P to a new algorithm Q with comparable time complexity, even when Q runs against prophetic adversaries. This implies that the benefit of P using randomness for dealing with the adaptive adversaries is limited.

LCP-Aware Parallel String Sorting
Jonas Ellert, Johannes Fischer and Nodari Sitchinava
Download paper from Springer LNCS. [video]

When lexicographically sorting strings, it is not always necessary to inspect all symbols. For example, the lexicographical rank of "europar" amongst the strings "eureka", "eurasia", and "excells" only depends on its so called relevant prefix "euro". The distinguishing prefix size D of a set of strings is the number of symbols that actually need to be inspected to establish the lexicographical ordering of all strings. Efficient string sorters should be D-aware, i.e. their complexity should depend on D rather than on the total number N of all symbols in all strings. While there are many D-aware sorters in the sequential setting, there appear to be no such results in the PRAM model. We propose a framework yielding a D-aware modification of any existing PRAM string sorter. The derived algorithms are work-optimal with respect to their original counterpart: If the original algorithm requires O(w(N)) work, the derived one requires O(w(D)) work. The execution time increases only by a small factor that is logarithmic in the length of the longest relevant prefix. Our framework universally works for deterministic and randomized algorithms in all variations of the PRAM model, such that future improvements in (D-unaware) parallel string sorting will directly result in improvements in D-aware parallel string sorting.

Mobile RAM and Shape Formation by Programmable Particles
Giuseppe Antonio Di Luna, Paola Flocchini, Nicola Santoro, Giovanni Viglietta and Yukiko Yamauchi
Download paper from Springer LNCS. [video]

In the distributed model Amoebot of programmable matter, the computational entities, called particles, are anonymous finite-state machines that operate and move on an hexagonal tasselation of the plane. In this paper we show how a constant number of such weak particles can simulate a powerful Turing-complete entity that is able to move on the plane while computing. We then show an application of our tool to the classical Shape-Formation problem, providing a new much more general distributed solution. Indeed, while the existing algorithms allow to form only shapes made of arrangements of segments and triangles, our algorithm allows the particles to form also more abstract and general con- nected shapes, including circles and spirals, as well as fractal objects of non-integer dimension. In lieu of the existing impossibility results based on the symmetry of the initial configuration of the particles, our result provides a complete characterization of the connected shapes that can be formed by an initially simply connected set of particles. Furthermore, in the case of non-connected shapes, we give almost-matching necessary and sufficient conditions for their formability.

Performance and Power Modeling, Prediction and Evaluation (B)

Chairs: Arnaud Legrand [Slack channel]

Operation-Aware Power Capping
Bo Wang, Christian Terboven, Matthias Mueller and Julian Miller
Download paper from Springer LNCS. [video]

Once the peak power draw of a large-scale high-performance-computing (HPC) cluster exceeds the capacity of its surrounding infrastructures, the cluster's power consumption needs to be capped to avoid hardware damage. However, power capping often causes a computational performance loss because the underlying processors are clocked down. In this work, we developed an operation-aware management strategy, called OAM, to mitigate the performance loss. OAM manages performance under a power cap dynamically at runtime by modifying the core and uncore clock rate. Using this approach, the limited power budget can be shifted effectively and optimally among components within a processor. The components with high computation activities are powered up while the others are throttled. The overall execution performance is improved. Employing the OAM on diverse HPC benchmarks and real-world applications, we observed that the hardware settings adjusted by OAM are having near to the optimal results of the static-setting approach. The achieved speedup in our work amounts to up to 6.3%.

Towards a Model to Estimate the Reliability of Large-scale Hybrid Supercomputers
Elvis Rojas, Esteban Meneses Rojas, Terry Jones and Don Maxwell
Download paper from Springer LNCS. [video]

Supercomputers stand as a fundamental tool for developing our understanding of the universe. State-of-the-art scientific simulations, big data analyses, and machine learning executions require high performance computing platforms. Such infrastructures have been growing lately with the addition of thousands of components, making them more prone to fail. It is crucial to solidify our knowledge on the way supercomputers fail. Several recent studies have highlighted the importance of characterizing failures on supercomputers. This paper aims at modelling component failures of a supercomputer based on Mixed Weibull distributions. The model is built using a real-life multi-year failure record from a leadership-class supercomputer. Using several key observations from the data, we designed an analytical model that is robust enough to represent each of the main components of supercomputers, yet it is flexible enough to alter the composition of the machine and be able to predict resilience of future or hypothetical systems.

A Learning-Based Approach for Evaluating the Capacity of Data Processing Pipelines
Maha Alsayasneh and Noel De Palma
Download paper from Springer LNCS. [video]

Data processing pipelines are made of various software components with complex interactions and a large number of configuration settings. Identifying when a pipeline has reached its maximum performance capacity is generally a non-trivial task. Metrics exported at the software and at the hardware levels can provide insightful information about the current state of the system, but it can be difficult to interpret the value of a metric, or even to know which metrics to focus on. Considering a popular pipeline composed of Kafka, Spark Streaming, and Cassandra, this paper proposes a learning-based approach to automatically infer the state of such a pipeline solely by analyzing metrics. Our results show that we are able to achieve a high prediction accuracy when predicting on new configurations and when the number of data sources changes. Furthermore, our analysis demonstrates that the best prediction results are obtained when metrics of different types are combined.

17:00 - 17:30 — **Break**

17:30 - 18:10

Theory and Algorithms for Parallel and Distributed Processing (A)

Chairs: Marek Klonowski [Slack channel]

Approximation Algorithm for Estimating Distances in Distributed Virtual Environments
Tobias Castanet, Olivier Beaumont, Nicolas Hanusse and Corentin Travers
Download paper from Springer LNCS. [video]

This article deals with the issue of guaranteeing properties in Distributed Virtual Environments (DVEs) without a server and without global knowledge of the system state and therefore only by exchanging messages. This issue is particularly relevant in the case of online games, that operate in a fully distributed framework and for which network resources such as bandwidth are the critical resources. In the context of games, players typically need to know the distance between their character and other characters, at least approximately. Players all share the same position estimation algorithm but, in general, do not know the current positions of others. We provide a synchronized distributed algorithm Alc to guarantee, at any time, that the estimated distance between any pair of characters A and B is always a 1 + ε approximation of the current distance. Our result is twofold: (1) we prove that if characters move randomly on a d-dimensional grid, or follow a random continuous movement on up to three dimensions, the number of messages of Alc is optimal up to a constant factor; (2) in a more practical setting, we also observe that the number of messages of Alc for actual game traces is much less than the standard algorithm sending actual positions at a given frequency.

3D Coded SUMMA: Communication-Efficient and Robust Parallel Matrix Multiplication
Haewon Jeong, Yaoqing Yang, Christian Engelmann, Vipul Gupta, Tze Meng Low, Pulkit Grover, Viveck Cadambe and Kannan Ramchandran
Download paper from Springer LNCS. [video]

In this paper, we propose a novel fault-tolerant parallel matrix multiplication algorithm called 3D Coded SUMMA that is communication efficient and achieves higher failure-tolerance than replication-based schemes for the same amount of redundancy. This work bridges the gap between recent developments in coded computing and fault-tolerance in high-performance computing (HPC). The core idea of coded computing is the same as algorithm-based fault-tolerance (ABFT), which is weaving redundancy in the computation using error-correcting codes. In particular, we show that MatDot codes, an innovative code construction for distributed matrix multiplications, can be integrated into three-dimensional SUMMA (Scalable Universal Matrix Multiplication Algorithm [22]) in a communication-avoiding manner. To tolerate any two node failures, the proposed 3D Coded SUMMA requires ~50 % less redundancy than replication, while the overhead in execution time is only about 5-10 %.

Performance and Power Modeling, Prediction and Evaluation (B)

Chairs: Arnaud Legrand [Slack channel]

A Comparison of the Scalability of OpenMP Implementations
Tim Jammer, Christian Iwainsky and Christian Bischof
Download paper from Springer LNCS. [video]

OpenMP implementations must exploit current and upcoming hardware for performance. Overhead must be controlled and kept to a minimum to avoid low performance at scale. Previous work has shown that overheads do not scale favourably in commonly used OpenMP implementations. Focusing on synchronization overhead, this work analyses the overhead of core OpenMP runtime library components for GNU and LLVM compilers, reflecting on the implementation's source code and algorithms. In addition, this work investigates the implementation's capability to handle current CPU-internal NUMA structure observed in recent Intel CPUs. Using a custom benchmark designed to expose synchronization overhead of OpenMP regardless of user code, substantial differences between both implementations are observed. In summary, the LLVM implementation can be considered more scalable than the GNU implementation, but the GNU implementation yields lower overhead for lower threadcounts in some occasions. Neither implementation reacts to the system architecture, although the effects of the internal NUMA structure on the overhead can be observed.

Evaluating the Effectiveness of a Vector-Length-Agnostic Instruction Set
Andrei Poenaru and Simon McIntosh-Smith
Download paper from Springer LNCS. [video]

In this paper we evaluate the efficacy of the Arm Scalable Vector Extension (SVE) instruction set for HPC workloads using a set of established mini-apps. Exploiting the vector capabilities of SVE will be a key factor in achieving high performance on upcoming generations of Arm-based processors. SVE is a flexible instruction set, but its design is fundamentally different from other contemporary SIMD extensions, such as AVX or NEON, which could present a challenge to its adoption. We use a selection of mini-apps that covers a wide range of scientific application classes to investigate SVE, using a combination of static and dynamic analysis. We inspect how SVE capabilities are used in the mini-apps' kernels, as generated by all SVE compilers available at the time of writing, for both arithmetic and memory operations. We compare our findings against similar data gathered on currently available processors. Although the extent to which vector code is generated varies by mini-app, all compilers tested successfully utilise SVE to vectorise more code than they are able to when targeting NEON. For most mini-apps, we expect performance improvements as SVE width is increased.

18:10 - 18:30 — **Closing (A)**