JLESC Workshop

Schedule & Talks

June 27, 2016

Contents

1 SCHEDULE

Monday, June 27

8:45am-12pm: Steering committee meeting (by invitation)
Location: ENS Monod Level 2 Salle du Conseil

9:30am-12pm: Application meeting (by invitation)
Location: ENS Monod Level 4 Room B1

The workshop begins at 2pm. The first session (2pm-4:20pm) is plenary, and we need the big amphitheater located at ENS Descartes (see map given in the participant booklet). All following sessions (end of Monday, Tuesday and Wednesday) will take place at ENS Monod (see map given in the participant booklet)

Registration takes place at ENS Monod, 4:30pm-7pm

FIRST SESSION
2pm-3:30pm: Welcome and news from the six partner institutions
Location: ENS Descartes Amphi Descartes
2pm-2:10pm: Welcome (Franck Cappello, Christian Perez, Yves Robert)
2:10pm-2:20pm: News from INRIA (Thierry Priol)
2:20pm-2:30pm: News from UIUC/NCSA (Ed Seidel)
2:30pm-2:40pm: News from BSC (Jesus Labarta)
2:40pm-2:50pm: News from Riken (Mitsuhisa Sato)
2:50pm-3pm: News from JSC (Thomas Lippert)
3pm-3:10pm: News from ANL (Paul Hovland)
3:10pm-3:20pm: Little break
3:20pm-4:20pm: Special session on JLESC applications (Gabrielle Allen, ANL)
3:20pm-3:30pm: Lightning talk on application 1
3:30pm-3:40pm: Lightning talk on application 2
3:40pm-3:50pm: Lightning talk on application 3
3:50pm-4pm: Lightning talk on application 4
4pm-4:10pm: Lightning talk on application 5
4:10pm-4:20pm: Lightning talk on application 6

4:20pm-4:30pm: Walking to ENS Monod
4:30pm-7pm: Registration (ENS Monod, Salle Passerelle, 4th floor)
4:30pm-5pm: Coffee break (ENS Monod, Salle Passerelle, 4th floor)
5pm-6pm: Break-out Applications
Four parallel sessions on four applications (rooms will be given)

7pm: Dinner at ENS Monod, Salle Passerelle, 4th floor

Tuesday, June 28

Morning, 8:40am-12:30pm
Parallel sessions: Programming models and runtime & Numerical methods

Programming models and runtime: 2 project talks, 9 lightning talks
Numerical methods: 6 project talks, 2 lightning talks (one moved from Applications)

Programming models and runtime:
8:40am-9am: P0 (Rob Ross, ANL) - shifted from I/O session
9am-10:30am: P1, P2, L1, L2, L3, L4, L5
10:30am-11am: Coffee break
11am-11:40am: L6, L7, L8, L9
11:40am-12:30pm: Discussion

Numerical methods:
9am-10:30am: P1, P2, P3, P4, L1
10:30am-11am: Coffee break
11am-11:50am: P5, P6, L2
11:50am-12:30pm: Discussion

Lunch, 12;30pm-2pm
Location will be given

Afternoon, 2pm-5pm
Parallel sessions: Resilience & Performance tools
Resilience: 3 project talks, 6 lightning talks
Performance tools: 2 project talks, 7 lightning talks

Resilience:
2pm-3pm: P1, P2, P3
3pm-3:30pm: Coffee break
3:30pm-4:30pm: L1, L2, L3, L4, L5, L6
4:30pm-5pm: Discussion

Performance tools:
2pm-3pm: P1, P2, L1, L2
3pm-3:30pm: Coffee break
3:30pm-4:20pm: L3, L4, L5, L6, L7
4:20pm-5pm: Discussion

6pm: Bus leaves to gala dinner
Gala dinner

Wednesday, June 29

Morning, 9am-12:30pm
Parallel sessions: I/O, storage and in situ & Applications
I/O, storage and in situ: 3 project talks, 12 lightning talks
Applications: 2 project talks, 7 lightning talks

I/O, storage and in situ:
9am-10:30am: P1, P2, P3, L1, L2, L3
10:30am-11am: Coffee break
11am-11:30am: L4, L5, L6
11:30am-12:30pm: Discussion

Applications:
9am-10:30am: P1, P2, L1, L2, L3, L4, L5
10:30am-11am: Coffee break
11am-11:20am: L6, L7
11:20am-12:30pm: Discussion

Lunch, 12;30pm-2pm
Location will be given

Afternoon, 2pm-5pm
Parallel sessions: I/O, storage and in situ & Clouds and new architectures
I/O, storage and in situ: cont'd
Clouds and new architectures: 2 project talks, 6 lightning talks

I/O, storage and in situ:
2pm-3pm: L7, L8, L9, L10, L11, L12
3pm-3:30pm: Coffee break
3:30pm-4:40pm: Discussion

Clouds and new architectures
2pm-3pm: P1, P2, L1, L2
3pm-3:30pm: Coffee break
3:30pm-4pm: L3, L4, L5, L6
4:10pm-4:40pm: Discussion

4:45pm-5pm: Closing session

Dinner
Location will be given (need to have registered)

2 APPLICATIONS

2.1 Project talks

2.1.1 Comparison of Meshing and CFD Methods for Accurate Flow Simulations on HPC systems

Andreas Lintermann (JSC)
Keiji Onishi (Riken)

Project: Comparison of Meshing and CFD Methods for Accurate Flow Simulations on HPC systems

Abstract:
The expertise of the collaborators at the two centers AICS and JSC lies in the development of methods for Computational Fluid Dynamics (CFD) simulations on HPC systems. The aim of this project is to compare the accuracy and efficiency of the applied methods in the two CFD simulation codes on the two hardware architectures based on predefined benchmark cases. This project will not only help to further develop an understanding for computational methods for large-scale CFD simulations for the next supercomputer generation, but will also characterize the efficiency of the current codes on different hardware architectures. In this talk, we will present the current implementation of the CFD methods, parallelization, and the associated meshing techniques on both code. Then the collaborative research plan and current progress will be reported.

2.1.2 Dynamic load balancing with Pampa in Alya

Guillaume Houzeaux (BSC)

Project: Dynamic load balancing with Pampa in Alya

Abstract:
In Alya, the transport of particles in a fluid is achieved using two parallel instances of the code. One instance is in charge of solving the flow equation and the other one of transporting the particles. The velocity field is sent from the first to the second one at the end of each time step through MPI. If a classical mesh partitioning based on the balancing of the number of elements is performed, the particles are likely to be located in very few subdomains, which obviously leads to a very poor load balance. In this project we aim at implementing the Pampa library in Alya in order to redistribute the elements among the MPI processes in order to dynamically balance the computation of particle transport.

2.2 Lightning talks

2.2.1 Load balancing of an MPI parallel unstructured CFD code using DLB and OpenMP

Guillaume Houzeaux (BSC)

Abstract:
This presentation addresses the dynamic load balancing of an MPI parallel finite element computational fluid dynamics code for unstructured meshes. In particular, we will focus on the assembly step which, in the finite element context, consists of a loop over the elements of the mesh to compute element matrix and right-hand side, an then to assemble them into global matrix and right-hand side. Load balance of this computation can hardly be achieved a priori, for both software and hardware issues. On the one hand, the mesh partitioning is never perfect. On the other hand, variability of the hardware can also be a random source of imbalance. To alleviate this load imbalance, a dynamic load balance library is applied on the top of OpenMP, which enables the use of the resources of idle MPI tasks by others. Through the solution of a practical CFD problem, we will show the efficiency of the dynamic load balance proposed in this work compared to the classical pure MPI and hybrid MPI+OpenMP approaches.

2.2.2 Lattice QCD with CG and multi-shift CG on Xeon Phi

Yoshifumi Nakamura (RIKEN)

Abstract:
Lattice QCD is an approach to solving the quantum chromodynamics (QCD) We present our optimization method and performance of conjugate gradient (CG) solver and multi-shift CG for lattice QCD on Xeon Phi.

2.2.3 Memory-efficient sparse direct solvers in large-scale frequency-domain geophysical simulations

Samuel Rodriguez (BSC)

Abstract:
Large-scale three-dimensional seismic and electromagnetic geophysical simulations can be formulated as implicit frequency-domain problems. The frequency-domain approach simplifies the correlation of the source and receiver wavefields for inverse problems, but it requires the solution of large sparse linear systems. Locally refined grids with huge material contrasts and high wavelet frequencies, for the electromagnetic and seismic case respectively, leads to extremelly ill-conditioned problems with many millions of unknowns. In this talk we will present how we approach these problems using narrow-banded memory-efficient sparse direct solvers.

2.2.4 Efficient Data Structures for Exascale Astrophysics

Vincent Reverdy (UIUC)

Abstract:
Over the last decade, great advances have been made in the domain of numerical astrophysics. In cosmology, we are now able to simulate the dynamics of galaxy clusters in volumes of significant size when compared to the observable Universe. However, the codes currently running on petascale supercomputers start to show some limitations. One of these limitations is due to the fact that data transfer

between nodes, and data transfer from random access memory to processor caches is becoming more and more a bottleneck when compared to the pure computing cost when everything is in cache. To address these problems, and build the bases of codes that will be able make the most of the next generation of supercomputers, we have explored ways to optimize data structures, and in particular trees. In this talk, I will summarize our approaches, how implicit trees can offer a far better alternative to explicit trees, and how our work has lead to discussions within the C++ standards committee for the next update of the language. I will also show how our work has some application domains that go far beyond the scope of numerical astrophysics.

2.2.5 Eigenspectrum calculation of large sparse non-Herimitian matrices in lattice QCD

Hiroya Suno (RIKEN)

Abstract:
We are exploring mathematical algorithms for computing eigenvalues and eigenvectors of large sparse non-Hermitian matrices arising in Lattice Quantum Chromodynamics (lattice QCD). Lattice QCD is a theory of quarks and gluons, formulated on a grid or lattice of points in space and time. Our goal is to obtain several low-lying eigenvalues of the non-Hermitian O(a)-improved Wilson-Dirac operator. We have been exploring so far the Sakurai-Sugiura (SS) method, a method based on a contour integral, which allows us to compute desired eigenvalues located inside a given contour of the complex plane, as well as the associated eigenvectors. Matrix inversion of shifted matrices is needed to implement the numerical code. Our implementation has been tested for large matrices with the matrix order being up to about one billion, allowing us to compute eigenvalues for several simple cases with a certain accuracy. The process of solving shifted equations is significantly slowed perhaps due to the dense distribution of eigenvalues on the overall complex plane, which also limits the accuracy of the computation. We are now seeking for an effective way to overcome this bottleneck.

2.2.6 Accelerating LBM & LQCD Application Kernels by In-Memory Processing

Thorsten Hater (JSC)

Abstract:
Processing-in-memory architectures promise increased com- puting performance at decreased costs in energy, as the physical proximity of the compute pipelines to the data store eliminates overheads for data transport. We assess the overall performance impact using a recently introduced architecture of that type, called the Active Memory Cube, for two representative scientific applications. Precise performance results for performance critical kernels are obtained using cycle-accurate simulations. We provide an overall performance estimate using performance models.

2.2.7 Using Accelerator Hardware to Improve Subresolution Modeling in Astrophysical Simulations

Paul M. Ricker (UIUC)

Abstract:
Exascale machines are expected to continue the trend toward complex, unbalanced architectures with multiple levels of parallelism. Communication between levels often introduces significant latency. This forces us to be more creative in mapping problems of interest onto the hardware. To address this problem, I describe the development of a Subresolution Accelerator Framework (SAF) designed to exploit the scale separation implicit in the use of subresolution models for accretion flows, star formation, and turbulence on systems in which a large fraction of the peak performance comes from attached GPUs or other types of accelerator hardware.

3 CLOUDS & NEW ARCHITECTURES

3.1 Project talks

3.1.1 OFPGA: OpenMP for FPGA

Franck Cappello (ANL)

Project: OpenMP for FPGA

Abstract:
The end of the Moore's law poses a significant challenge for scientific computing: from the mid 2020s, performance will not improve any more from the CMOS technology progress. Reconfigurable computing presents the unique opportunity of allowing performance progress by customization while still serving a large variety of applications, offering a true co-design vehicle. However, its adoption in scientific computing still faces the lack of high-level parallel programming abstraction and the extreme difficulty of achieving high performance with existing compilation stacks. The OFPGA project explores solutions for these two obstacles. We focus on node-level parallelism; our objective is to demonstrate, for a set of applications parallelized with OpenMP 4, significant performance/watt improvement (factor of 2 to 5) and similar performance on average compared with CPUs and GPUs of the comparable technology. To cover a large spectrum of applications, we chose applications that have different properties in terms of memory-compute bounding, memory access regularity, data format, and Berkeley motifs. We follow a co-design approach. We first provide a path to compile OpenMP 4 applications for high-end field-programmable gate arrays (FPGAs). Second, we explore, design, and develop new mechanisms to optimize the FPGA performance of scientific computing. Our codesign approach synergistically affects the programming interface and the hardware configuration. The project will produce a deep understanding on how to reach better performance per watt on scientific applications with FPGAs than with CPUs and GPUs. We will identify the cases in which code transformations and additional directives (beyond OpenMP 4) are required in order to reach this goal.

3.1.2 On-Demand Data Analytics and Storage for Extreme-Scale Simulations and Experiments

Salman Habib (ANL)
Katrin Heitmann (ANL)

Project: On-Demand Data Analytics and Storage for Extreme-Scale Simulations and Experiments

Abstract:
The science requirements in cosmology demand running very large simulations, such as N-body runs with trillions of particles. The resulting data products are scientifically very rich and of interest to many research groups. It is therefore very desirable that the data be made broadly available. However, as a fiducial example, a trillion-particle simulation with the HACC code generates 20 PB of raw data (40 TB per snapshot and 500 snapshots), which is is more than petascale systems such as Mira and Blue Waters can store for a single run in their file systems. An interesting point is that while one version of HACC is optimized for Mira and can scale to multi-millions of cores, Blue Waters offers exceptional data analytics capabilities with its thousands of GPUs. This suggests a combined infrastructure based on using Mira for the simulation, Blue Waters for a first-level data analysis, and a separate, possibly distributed, data center to store the distilled results. Users from other universities and labs would then pull the data from the data center and run further data analytics locally following their particular scientific interests. We propose an elastic virtual infrastructure connecting several data production sites (comprising both simulation and experiment), data centers for storage, and data analysis centers. The infrastructure set-up is not permanent; it is dynamically instantiated on demand for transient needs (even if the data is stored on a long-term basis in the data centers). The same resources (data production side, data centers, and analytics centers) could be used or shared by other scientific communities instantiating infrastructure components, potentially aggregating other resources as well. Our concept relies on technologies needed for the convergence of HPC and big data, such as virtualized resources, resource reservation, software deployment, user group management, policy management, etc. It combines some grid principles by aggregating geographically distributed resources owned by different institutions, as well as some cloud principles: Infrastructure as a Service, elasticity, and virtualization. In order to demonstrate the feasibility of this concept, a team of about 15 researchers and staff from Argonne, UIUC, UIC, DDN, and including SCinet leaders, is preparing an experiment which will be demonstrated at SC16. This experiment and demonstration covers several critical elements of on-demand data analytics and storage.

3.2 Lightning talks

3.2.1 Incorporating Probabilistic Optimizations for Resource Provisioning of Cloud Workflow Processing

Amelie Chi Zhou (INRIA)

Abstract:
Resource provisioning of data processing workflows in the cloud has recently attracted many research efforts. Still, most previous studies fail to consider cloud dynamics, including the I/O and network performance dynamics and price dynamics of the cloud, which greatly affect the effectiveness of resource provisioning for workflows in the cloud environment. In this talk, we propose to take probability distributions of cloud dynamics as optimization input, and to incorporate probabilistic optimizations for resource provisioning of workflows in the cloud. While probabilistic optimizations can improve the effectiveness of resource provisioning, they can introduce prohibitively high optimization overhead. In this talk, we propose several initial directions for reducing the overhead of probabilistic optimizations, and call for more discussions and collaborations.

3.2.2 Exploring elastic scaling on Chameleon Cloud

Luis Pineda (INRIA)

Abstract:
One of the challenges of applications dealing with dynamic data streams is that the computation requirements and the streams themselves are volatile (highly variable). Cloud computing is a promising platform to cope with such volatility because it enables to allocate computational resources on demand, for short periods of time, and at an acceptable cost. Currently, we tackle the challenge of how and where schedule resources for this type of applications, using Chameleon Cloud in its different flavors of infrastructures.

3.2.3 Adaptation of a HPC system to FPGA

Georgios Christodoulis (INRIA)

Abstract:
Multicore architecture development appeared to tackle the unsustainable power consumption growth of single core CPUs. The next step towards this direction is the use of accelerators for the execution of tasks with certain characteristics (e.g. GPUs -OpenCL/CUDA kernels). In the scope of HEAVEN project we attempt the development of a heterogeneous system that will enable task acceleration using FPGAs, exploiting their outstanding energy efficiency. In our approach application programming would be feasible using OpenMP-the standard programming environment for shared memory architectures, hiding low level hardware specific mechanisms from the application developer (e.g. memory transfers between the host and the device). The HLS tool we are using to generate the corresponding VHDL for the configuration of the FPGA, is AUGH, because of its ability to provide very quickly an RTL description of the design under resources constrains. We also extend StarPU to support the new device, a runtime system that provides mechanisms for heterogeneous scheduling, data transfers, and intranode communication.

3.2.4 OmpSs FPGA support

Carlos Alvarez (BSC)

Abstract:
In this talk, we will show our ongoing work dealing with OmpSs and FPGAs. The main goal of the first presented project is to use OmpSs to simplify the programming of FPGAs as program accelerators. In this sense FPGAs are used as heterogeneous accelerators in the same way as GPUs. Our Mercurium compiler outlines the code annotated for the FPGA onto separate functions to be executed as tasks, and our Nanos++ runtime takes care of all the small details that are cumbersome to programmers: task scheduling, data transfers, data consistency, etc. The talk will highlight the performance results obtained when comparing our system to other state-of-the-art alternatives. The second project uses the FPGA directly as an accelerator for Nanos++ runtime and in fact can be used to accelerate any dataflow task-based programming model. It demonstrates the usefulness of hardware support to alleviate the overheads of runtime dependence management in many-core environments.

3.2.5 ASAP On CAPI: Intro to IBM CAPI and Computational Genomics Case Study

Carl Pearson (UIUC)

Abstract:
CAPI (Coherent Accelerator Processor Interface) is a framework for connecting IBM POWER8/9 systems to GPUs, FPGAs, ASICs, and other accelerators, and will be a key enabling component for the Summit and Sierra next-generation supercomputers. CAPI removes a significant performance and productivity bottleneck by sharing the CPU address space with attached accelerators. This talk presents a brief introduction to CAPI and preliminary experience from integrating it with a high-performance computational genomics short-read-alignment accelerator on an FPGA. Joint work with Simon Garcia De Gonzalo.

3.2.6 Executing COMPSs applications with Docker

Jorge Ejarque (BSC)

Abstract:
One of the current alternatives of virtualization to deploy isolated distributed applications in data centers is Docker. It provides an efficient image management which reduces the overhead introduced by current hypervisors by keeping most of their capabilities. In this talk, we present how developers can use COMPSs to implement distributed applications and deploy them transparently in container engines like Docker.

4 I/O, STORAGE, IN-VISU

4.1 Project talks

4.1.1 Evaluation of Topology-Aware Broadcast Algorithms for Dragonfly Networks

Matthieu Dorier (ANL):

Project: Mitigating I/O interference in concurrent HPC applications

Abstract:
Two-tiered direct network topologies such as Dragonflies have been proposed for future post-petascale and exascale machines, since they provide a high-radix, low-diameter, fast interconnection network. Such topologies call for redesigning MPI collective communication algorithms in order to attain the best performance. Yet as increasingly more applications share a machine, it is not clear how these topology-aware algorithms will react to interference with concurrent jobs accessing the same network. In this talk, we present a study of three topology-aware broadcast algorithms. We evaluate their performance through event-driven simulation for small- and large-sized broadcasts (both in terms of data size and number of processes). We study the effect of different routing mechanisms on the topology-aware collective algorithms, as well as their sensitivity to network contention with other jobs. Our results show that while topology-aware algorithms drastically reduce link utilization, their advantage in terms of latency is more limited.

4.1.2 From File Systems to Services: Changing the Data Management Model in HPC

Rob Ross (ANL)

Project: Towards Interference-aware scheduling in HPC systems

Abstract:
HPC applications are composed from software components that provide only the communication, concurrency, and synchronization needed for the task at hand. In contrast, parallel file system are kernel resident, fully consistent services with semantic obligations developed on single core machines 50 years ago; parallel file systems are old-fashioned system services forced to scale as fast as the HPC system. Rather than the monolithic storage services seen today, we envision an ecosystem of services being composed to meet the specific needs of science activities at extreme scale. In fact, a nascent ecosystem of services is present today. In this talk we will discuss drivers leading to this development, some examples in existence today, and work we are undertaking to accelerate the rate at which these services are developed and mature to meet application needs.

4.1.3 Decaf : Building in situ applications without caffeine

Matthieu Dreher (ANL)

Project: Extreme-Scale Workflow Tools: Swift, Decaf, Damaris, FlowVR

Abstract:
In situ applications are a promising solution to tackle the problem of imbalance between computational capabilities and I/O bandwidth in leadership supercomputers. Initially designed to focus on I/Os, in situ applications now include a wide range of domains such as visualization, machine learning, filtering or feature tracking. We present the first results of Decaf, an in situ middleware enabling the user to describe an in situ application as a graph compatible with leadership supercomputers. Decaf focuses on the link between a parallel producer and parallel consumer and enhanced this link by providing a staging area to transform data, implement resilience mechanism or buffering. In situ infrastructures have to deal with complex heterogeneous codes using different data structures. Decaf relies on Bredala, a data model library, to exchange data between parallel codes. Bredala ensures that the semantic integrity of data is preserved during split/merge operations. We propose an evaluation of our data model library and describe the current state of Decaf. We also discuss the differences of Decaf compare with other in situ middlewares within the JLESC such as FlowVR and Damaris.

4.1.4 Modeling and avoiding execution interferences

Raphael Bleuse (INRIA)

Project: Modeling and avoiding execution interferences

Abstract:
The trend in supercomputers is to integrate a unique and multi-purpose interconnection network. Such a network carries both IO traffic and internal communications of jobs. Using convexity for allocation as a mean to mitigate inter-jobs interaction, how can we take into account IO traffic? In this talk, we will present preliminary complexity results for this problem. Joint work with Giorgio Lucarelli and Denis Trystram.

4.2 Lightning talks

4.2.1 Týr: Blob Storage Systems Meet Built-In Transactions

Pierre Matri (INRIA)

Abstract:
Concurrent Big Data applications often require high-performance storage, as well as ACID (Atomicity, Consistency, Isolation, Durability) transaction support. Blobs (binary large objects) are an increasingly popular low-level model for addressing the storage needs of such applications, providing a solid base for developing higher-level storage solutions, such as object stores or distributed file systems. However, today's blob storage systems typically offer no transaction semantics. This demands users to coordinate access to data carefully in order to avoid race conditions, inconsistent writes, overwrites and other problems that cause erratic behavior. We argue there is a gap between existing storage

solutions and application requirements, which limits the design of transaction-oriented applications. In this talk, we briefly introduce Týr, the first blob storage system to provide built-in, multiblob transactions, while retaining sequential consistency and high throughput under heavy access concurrency.

4.2.2 Spark versus Flink: Understanding Performance in Big Data Analytics Frameworks

Gabriel Antoniu (INRIA)

Abstract:
Big Data analytics has recently gained increasing popularity as a tool to process large amounts of data on-demand. Spark and Flink are two Apache-based data analytics frameworks that facilitate the development of multi-step data pipelines using directly acyclic graph patterns. Making the most out of these frameworks is challenging because efficient executions strongly rely on complex parameter configurations and on an in-depth understanding of the underlying architectural choices. Although extensive research has been devoted to improving and evaluating the performance of such analytics frameworks, most of them benchmark them against Hadoop, as a baseline, a rather unfair comparison considering the fundamentally different design principles. This work aims to bring some justice in this respect, by directly comparing the performance of Spark and Flink. Our goal is to identify and explain the impact of the different architectural choices and the parameter configurations on the perceived end-to-end performance. To this end, we develop a methodology for correlating the parameter settings and the operators execution plan with the resource usage. We use this methodology to dissect the performance of Spark and Flink with several representative batch and iterative workloads on up to 100 nodes. We highlight how performance correlates to operators, to resource usage and to the specifics of the internal framework design.

4.2.3 Interactive Visualization and Analysis of Performance Anomalies: Crossing the Hardware, Run-Time and Application Layer for OpenStream and OpenMP

Andi Drebes (INRIA & U. Manchester)

Abstract:
In this talk, we present Aftermath, a tool for the interactive visualization and post-mortem analysis of execution traces generated by task-parallel OpenStream programs and loop-parallel OpenMP applications. We focus on the detection of performance anomalies inaccessible to state-of-the-art performance analysis techniques, such as anomalies deriving from the interaction of multiple levels of software abstractions, anomalies associated with the hardware, and anomalies resulting from interferences between optimizations in the application and run-time system. We show how Aftermath can be used to visualize and analyze anomalies involving multiple layers and components in the system using its mechanisms for filtering, aggregation and joint visualization of key metrics and performance indicators. We further illustrate Aftermath's capability to takes advantage of explicit memory regions and dependence information in de-

pendent task models to precisely capture long-distance and inter-core effects on machines with non-uniform memory access (NUMA). Finally, we present Aftermath-OpenMP, a ready-to-use state-of-the-art OpenMP run-time generating traces for Aftermath.

4.2.4 Analysis and Visualization of Dynamic Runtime Traces for the Scheduling Researcher

Arnaud Legrand (INRIA)

Abstract:
Hybrid (multi-core and multi-GPU) architectures are now common-place and exploiting them directly through OpenMP, CUDA, or OpenCL is quite burdensome for application developers. In such context, portability and performance optimization only comes at a prohibitive cost. There is thus a recent and general trend in using a modular approach where numerical algorithms are written at a high level independently of the hardware architecture as Directed Acyclic Graphs (DAG) of tasks. A task-based runtime system (StarPU, StarSs, QUARK, DAGuE, ...) then dynamically schedules the resulting DAG on the different computing resources, automatically taking care of data movement and taking into account the possible speed heterogeneity and variability. Such runtime implement involved scheduling algorithms that strive for optimal executions and have to dynamically manage the trade-off between efficiently using resources and progressing on the critical-path of the application. The resulting executions are thus stochastic with fuzzy synchronizations, which makes the analysis of execution traces very difficult. We will present how we have built semi-interactive trace visualizations that exploit the application DAG structure and allow to quickly filter information and identify whether further performance improvement can be expected or not and, if so, what the mistakes of the scheduler have been.

4.2.5 Combining On-Demand Availability and Batch Scheduling in HPC Datacenters

Kate Keahey (ANL)

Abstract:
Introducing on-demand availability to HPC datacenter faces a utilization challenge: in order to provide on-demand functionality, a proportion of resources typically needs to be kept idle to satisfy an incoming request — but this leads to low utilization which makes it hard to amortize the resource. Given the increasing demand for on-demand availability, we propose a model in which an on-demand framework and batch scheduled framework share a resource negotiating access to resources dynamically depending on workflow using a component we call Balancer. We analyzed our approach for feasibility using submission traces from the Advanced Photon Source at Argonne (which requires on-demand access to support ongoing experiments) and the batch cluster at Laboratory Computing Resource Center. We then followed up with an implementation of the Balancer which has been tested using OpenStack to represent the on-demand framework, and Torque to represent batch. In this talk, we will describe the

system and present our experiences to date.

4.2.6 Lowering the barriers to In-Situ Visualization

Jens Henrik Göbbert(JSC)

Abstract:
Extracting scientific insight from large simulations is of crucial importance for science. Scientific advances are made only once the data produced by the simulations is processed into a meaningful analysis. But as simulations increase in size for more detail, post-processing is becoming the bottleneck on the way to the desired scientific insight. In situ techniques, like in situ visualization, are known to play an important part in tackling this problem. Sparing some supercomputing time to process, structure, reduce and visualize the data in real-time during the simulation offers several benefits. In particular, when data reduction becomes inevitable, only during the simulation all relevant data about the simulated fields and any embedded geometry is readily available at the highest resolution and fidelity for critical decision making. For the integration of in situ visualization into multiple large scale simulation codes a light-weighted, flexible and easy-to-use coupling library has been developed. It covers the complexity and numerous options of in situ visualization and lowers the barriers to integrate visualization techniques into an existing simulation code. It allows to use the in situ functionality of VisIt as well as ParaView without disturbing the well-established work-flows in simulation code development. This library has been successfully integrated into highly scalable simulation codes running at Juelich Supercomputing Centre beginning with CIAO and psOpen of the Institute for Combustion Technology written in Fortran90 and ZFS of the Institute of Aerodynamics Aachen written in C++, which are all members of the High-Q Club for codes scaling to the complete JUQUEEN.

4.2.7 A Data + Visual Analytics Approach to Understanding Parallel I/O

Rob Sisneros (UIUC)

Abstract:
The process of optimizing parallel I/O can quite easily become daunting. By the nature of its implementation there are many highly sensitive, tunable parameters and a subtle change to any of these may have drastic or even completely counterintuitive results. There are many factors affecting performance: complex hardware configurations, significant yet unpredictable system loads, and system level implementations that perform tasks in unexpected ways. A final compounding issue is that an optimization is very likely specific to only a single application. The state of the art then is usually a fuzzy mixture of expertise and trial-and-error testing. In this talk I will explore possible next steps in following up recent work providing a characterization of application I/O based on a combination of job-level and filesystem-level aggregation.

4.2.8 NetCDF-based I/O Middleware for Supporting Direct Data Transfer

Tatiana Martsinkevich (Riken)

Abstract:
On the verge of the convergence between high performance computing (HPC) and Big Data processing, it has become increasingly prevalent to deploy large-scale data analysis workloads on high-end supercomputers. Such applications often come in the form of complex workflows with various components, often developed by different teams of programmers. For example, as part of the next generation flagship (post-K) supercomputer project in Japan, RIKEN is investigating the feasibility of a highly accurate weather forecasting system that would provide a real-time prediction for severe guerrilla rainstorms. One of the main performance bottlenecks of this application is the lack of efficient communication among its components, which currently takes place over the parallel file system. In my talk, I will present a direct communication framework that eliminates file I/O among components of such complex applications. Such I/O arbitrator will provide direct parallel data transfer among job components that rely on the netCDF interface for performing I/O operations. To use our framework, user will have to do minimal modifications to the program. I will present the design and an early evaluation of the framework on the K Computer using up to 4800 nodes running RIKEN's experimental weather forecasting workflow as a case study.

4.2.9 Improving lossy compression for scientific data

Franck Cappello (ANL)

Abstract:
In every domain where the infrastructure cannot communicate and/or store the generated raw data directly, data compression is a critical data transformation that contributes to satisfying the end-user needs. For example, data compression is already widely used for image and signal compression in many consumer products. Compression is also needed in large-scale data centers (Yahoo compresses emails), and lossy compression is an active research topic for medical imaging and genomic applications. Compression is also needed for scientific datasets, for example, the RAVEN project proposes to use the Argonne Advanced Photon Source for x-ray tomography of integrated circuits (ICs) that will produce 32 TB of data for each IC. Storing or communicating these raw datasets without significant data reduction is impossible. Often, users need to reduce the data by a factor 10 to 100 to obtain reasonable communication and storage times. The approach of data omission or decimation (i.e., storing only 1 data point of 10 produced, 1 snapshot for 10 produced), often used for simulations, is not satisfactory because it impairs the accuracy of the analytics performed from the simulation. Although compression is critical to evolve many scientific domains to the next step, the technology of scientific data compression and the understanding on how to use it are still in their infancy. The first evidence is the lack of results in this domain: over the 26 years of the prestigious IEEE Data Compression Conferences, only 12 papers identify an aspect of sci-

entific data in their title (floating-point data, data from simulation, numerical data, scientific data). The second evidence is the poor performance on hard to compress datasets. Considering that exascale execution and extreme-scale experiments will produce even more data than current simulations and experiments, new lossy compressor technology for hard-to-compress scientific datasets is urgently needed in order to support the communication, storage, and analysis of this data. In this talk we will review the techniques used by the best in class compressor and discuss its limitations. We also discuss potential application of lossy compression in scientific simulations.

4.2.10 Combining On-Demand Availability and Batch Scheduling in HPC Datacenters

Kate Keahey (ANL)

Abstract:
Introducing on-demand availability to HPC datacenter faces a utilization challenge: in order to provide on-demand functionality, a proportion of resources typically needs to be kept idle to satisfy an incoming request — but this leads to low utilization which makes it hard to amortize the resource. Given the increasing demand for on-demand availability, we propose a model in which an on-demand framework and batch scheduled framework share a resource negotiating access to resources dynamically depending on workflow using a component we call Balancer. We analyzed our approach for feasibility using submission traces from the Advanced Photon Source at Argonne (which requires on-demand access to support ongoing experiments) and the batch cluster at Laboratory Computing Resource Center. We then followed up with an implementation of the Balancer which has been tested using OpenStack to represent the on-demand framework, and Torque to represent batch. In this talk, we will describe the system and present our experiences to date.

4.2.11 Characterising I/O behavior of a large set of applications using file system performance counters

Salem El Sayed (JSC)

Abstract:
As high-end high-performance computing is advancing from petascale to exascale, there are various challenges which apply in particular to the capabilities of the I/O sub-system. A good understanding of characteristics of today's utilisation of such I/O sub-systems can help to address this challenge. In this talk we present results from an analysis of server-side performance counters that had been collected for multiple years on a parallel file system attached to a petascale Blue Gene/P system. To aid in the analysis we, developed a set of general performance characterisation metrics, which we applied to this large dataset. As the collection of server-side performance counters continues on new HPC systems with varying I/O sub-systems, the interest in improving the analysis methods grows. It is therefore of value to communicate the current methods of analysis and the results with the community. This will aid in improving the analysis and guiding it towards needed results from a cross system large scale

application I/O analysis using file system performance counters.

4.2.12 Locality-aware Task Scheduling in a Message Passing and MapReduce Hybrid Model

Shinichiro Takizawa (RIKEN)
Abstract:
Many workflows of scientific applications include ensemble simulations that can be expressed by MapReduce programming model, for example, by running simulation codes in Map phase and an ensemble mean calculation codes in Reduce phase. However, existing MapReduce frameworks do not help improving performance by exploiting data locality as each simulation runs on multiple nodes using MPI and the systems do not aware of structures of simulation data and their decomposition between nodes. We propose an MPI and MapReduce hybrid programming model that can allocate MPI parallel tasks to achieve high data locality. In our model data passed between tasks are modeled as multi-dimensional arrays and they can be split into blocks by users' view of data at runtime, each of which is mapped by a user function. Users can also specify how to decompose the data between nodes at the same time to collect associated data local. We present the model, its prototype implementation and evaluations.

5 NUMERICAL METHODS

5.1 Project talks

5.1.1 HPC libraries for solving dense symmetric eigenvalue problems

Toshiyuki Imamura (RIKEN)

Project: HPC libraries for solving dense symmetric eigenvalue problems

Abstract:
Many applications for example in Density Functional Theory (DFT) used in physics, chemistry, and materials science have to compute eigenvalues and eigenvectors of dense symmetric matrices. In the project, we port several dense eigenvalue libraries onto available computational resources and evaluate performance, accuracy and reproducibility. We currently plan to do that with all combination of libraries (ELPA, Elemental, EigenExa) x systems (K, JUQUEEN, JURECA). On the meeting at Lyon, the current status of the project will be addressed.

5.1.2 Implementing a task parallel FMM on top of Argobots

David Haensel (JSC)

Project: Scalability Enhancements to FMM for Molecular Dynamics Simulations

Abstract:
This project focuses on increasing the scalability of the JSC-developed Fast

Multipole Method (FMM) library used in molecular dynamics simulations. Due to mainly small problem sizes of only a few million particles and the availability of large scale supercomputers we are interested in the strong-scaling up to few particles per core. In this talk we will present current results from our task-based intra-node parallelization. First, we present our extended FMM implementation supporting asynchronous task execution and task-dependency resolving. Based on this high level of abstraction in C++ we implemented a data-driven and task-parallel version with the help of Argobots. Hereby, we focused on fine-grained and flexible tasks. Together with fast context switching from Argobots user-level threads, we will be able to hide inter-node communication done with MPI in the future. We will also show a performance analysis as well as an outlook towards the full hybrid parallelization.

5.1.3 Application of the ChASE eigensolver to excitonic Hamiltonians

Edoardo Di Napoli (JSC)

Project: Optimizing ChASE eigensolver for Bethe-Salpeter computations on multi-GPUs

Abstract:
Numerically solving the Bethe-Salpeter equation for the optical polarization function is a very successful approach for describing excitonic effects in first-principles simulations of materials. Converged results for optical spectra and exciton binding energies are directly comparable to experiment and are of predictive quality, thus allowing for computational materials design. However, these accurate results come at high computational cost: For modern complex materials this approach leads to large, dense matrices with sizes reaching up to n 400k. Since the experimentally most relevant exciton binding energies require only the lowest eigenpairs of these matrices, iterative schemes are a feasible alternative to prohibitively expensive direct diagonalization. The Chebyshev Accelerated Subspace iteration Eigensolver (ChASE), which is developed at JSC, is an ideal solver for solving such large dense eigenvalue problems. ChASE leverages on the preponderant use of BLAS 3 subroutines to achieve close-to-peak performance. Moreover, the code is parallelized for many- and multi-core platforms. In the initial phase of the project we are conducting feasibility tests comparing the shared memory parallelization of ChASE with the state-of-the-art direct eigensolver on problems ranging from n 20k up to n 60k. The long-term objective is to develop a distributed CPU/GPU parallelization of ChASE in order to solve larger eigenproblems by effectively exploiting heterogeneous multi-GPU architectures.

5.1.4 Advancement report on the interfacing of OpenAD and Tapenade

Laurent Hascoet (INRIA)

Project: Interfacing OpenAD and Tapenade

Abstract:
Development of a capable algorithmic differentiation (AD) tool requires large developer effort to provide the various flavors of derivatives, to experiment with the many AD model variants, and to apply them to the candidate application languages. Considering the relatively small size of the academic teams that develop AD tools, collaboration between them is a natural idea. This collaboration can exist at the level of research ideas as well as tool development. We will describe the current ongoing effort to provide interoperation of the two AD tools OpenAD and Tapenade. These tools have a close enough AD model, which allows for interoperation. Still, they rely on different data-flow analysis and implement specific optimizations. The aim of such interoperability is to ensure the robustness and stability of the AD tools. The redundancy between some components of either tool would offer more flexibility to the end-user. A weakness in one component may be compensated by choosing another route in the components graph. In the same order of ideas, a long-term objective is "a‘ la carte" AD, where one may combine powerful capabilities from either tools for instance, the preaccumulation capacities of OpenAD with the accurate data-flow model of Tapenade for activity, adjoint liveness, and TBR analysis. Additionally, not relying on any one component such as the OpenAD front-end compiler developed externally, allows the AD tool to persist beyond the lifetime of that front end compiler. Even further, we can analyze the strengthes and weaknesses of each tool's AD model. In the present state of development, one example of interoperative pipeline uses the parsing and source analysis capabilities of Tapenade with the transformation algorithms of OpenAD.

5.1.5 Overview of Task-based Sparse and Data-sparse Solvers on Top of Runtime Systems and potential JLESC collaborations

Emmanuel Agullo (INRIA)

Project: Iterative and direct parallel linear solvers in a hybrid MPI/OpenMP high performance computational engineering simulations

Abstract:
The complexity of the hardware architectures of modern supercomputers led the community of developers of scientific libraries to adopt new parallel programming paradigms. Among them, task-based programming has certainly become one of the most popular as it allows for high productivity while ensuring high performance and portability by delegating tasks management to a runtime system. In this talk, we will present an overview of sparse solvers that have been designed in the context of the Matrices Over Runtime Systems @ Exascale (MORSE) and Solvers for Heterogeneous Architectures (SOLHAR) projects. We will present the design of new direct solvers implementing supernodal (PaStiX) and multifrontal (qr mumps) methods, new Krylov solvers ensuring pipelining both at a numerical and software level, new sparse hybrid methods (MaPHyS) as well as data sparse libraries implementing fast multipole methods (ScalFMM) and hierarchical matrices (hmat, in collaboration with Airbus Group Innovations). For all these methods, we will highlight the challenges we have faced in terms of expressivity, granularity, scheduling and scalability and illustrate their performance on large academic and industrial test problems. We

will also present future challenges that will need to be faced in following years and that could benefit from the JLESC context to be jointly tackled.

5.1.6 Reducing Communication in Sparse Iterative and Direct Solvers

Amanda Bienz (UIUC)

Project: Reducing Communication in Sparse Iterative and Direct Solvers

Abstract:
The sparse matrix-vector multiply (SpMV) is a dominant operation in iterative methods, such as conjugate gradient, GMRES, and the solve phase of algebraic multigrid. The performance of many solvers can be improved by reducing the cost of each SpMV. In parallel, the SpMV often has poor scalability due to large costs associated with communication. Performance modeling can illustrate the dominant source of communication costs, and can highlight where future research should concentrate on reducing costs. This talk focuses on using topology-aware methods, such as the mapping of ranks to physical nodes, to improve the standard alpha-beta performance model for communication and highlight the source of communication costs associated with various SpMVs.

5.2 Lightning talks

5.2.1 Program Verification for Extreme-Scale Applications (ProVESA): motivation, overview, and math issues

Paul Hovland (ANL)

Abstract:
The ProVESA project involves development of new software verification technologies focused on the numerical and mathematical aspects of scientific software. The aim is to facilitate the migration of scientific software from sequential or bulk-synchronous execution on homogeneous architectures to nondeterministic, asynchronous execution on complex, hierarchical, and heterogeneous architectures. We explain the motivation for the project, provide an overview of the research directions, and describe some of the mathematical and numerical challenges due to roundoff errors and the differences between floating point and real arithmetic.

5.2.2 Using the fully coupled TerrSysMP in quasi-operational forecast mode on JSC/JURECA

Fabian Gasper (JSC)

Abstract:
The multi-physics, massively parallel MPMD coupled Terrestrial Systems Modelling Platform (TerrSysMP) is used on the JSC/JURECA cluster in a fully automatic, parallel pre-processing, modelling and visualisation framework for daily real time weather and geoscience simulations at very high resolution over a European and regional model domain for a comprehensive estimation of water cycle states and fluxes; results are then automatically staged on the HPSC

TerrSys YouTube Channel. This presentation points out technical aspects as well as setup details of the forecasting system. It furthermore puts a special focus on ongoing challenges around ensemble simulations and data assimilation.

6 PERFORMANCE TOOLS

6.1 Project talks

6.1.1 Data distribution approaches for heterogeneous memory systems

Antonio J. Peña (BSC)
Lena Oden (ANL)

Project: Use of the Folding profiler to assist on data distribution for heterogeneous memory systems

Abstract:
In this talk we will present the big picture of this project along with our latest progress. In particular, we will talk about the profiling tool we have been developing and we will present the analysis of a pair of use cases. We will also describe how this profiling tool is used to identify the critical path sections and memory objects. We will show how runtime systems may use this information to improve performance, e.g., to perform asynchronous prefetching of regular and irregular access patterns.

6.1.2 Developer tools for porting and tuning parallel applications on extreme-scale parallel systems

Hitoshi Murai (RIKEN)

Project: Developer tools for porting and tuning parallel applications on extreme-scale parallel systems

Abstract:
Since the previous JLESC meeting in Bonn, we jointly organised a second three-day hands-on VI-HPS Tuning Workshop at RIKEN AICS in Kobe, in which BSC (Paraver/Extrae/Dimemas) and JSC (Scalasca/Score-P/CUBE) performance tools where used to analyse and tune parallel applications on K computer and a local Fujitsu FX10 system. This included the RIKEN FIBER mini-app suite (particularly NTChem) and the NEST neuronal network simulation tool. Beyond tools training, we continue to investigate how to exploit our tools' capabilities in an integrated workflow. Basic integration already allows Scalasca measurement of ompSs and XMP applications, however, we'd like to improve and extend this particularly to exploit recent developments such as lightweight threading and tasking with Argobots.

6.2 Lightning talks

6.2.1 Simplified sustained performance benchmarks

Miwako Tsuji (RIKEN)

Abstract:
The SSP (Sustained System Performance) metric is used to measure the performance of existing and future supercomputer systems. The SSP metric takes into account the performance of various scientific applications and input data sets, which represent some part of the sites' workload. Here, we propose SSSP (Simplified Sustained System Performance) metric that makes performance projection using a set of simple existing benchmarks to the SPP metric for real applications. The benchmarks used as the "simple" may be existing simple benchmarks such as HPCC benchmarks. Although it is important to meet the requirements of various real-applications, not benchmarks, simple benchmarks are easy to port, optimize, execute and estimate their performance on various kinds of systems.

6.2.2 Semi-Automatic Performance Optimization of HPC Kernels

Brice Videau (INRIA)

Abstract:
Porting and tuning HPC applications to new platforms is of paramount importance but tedious and costly in terms of human resources. In the Mont-Blanc and Mont-Blanc 2 European projects we proposed the BOAST metaprogramming framework, which is aimed at generating optimized HPC computation kernels. Using BOAST, the programmer can express optimizations of a computing kernel and thoroughly search this optimization space, finding the best candidate on the targeted architecture. This yields performance portability that would otherwise be difficult to achieve. But a brute force search of the optimization space is obviously impractical as the optimization space grows exponentially with the number of parameters tested. Using generic meta-heuristics (genetic algorithms, tabu search, ...) is then often seen as an alternative but assessing their efficiency is quite difficult. In particular, once a solution is obtained in a given time budget, it is hard to know whether such solution can be further improved and if so how. We thus propose a semi-automatic methodology to alleviate this problem. We first characterize the optimization space using sampling techniques and model the objective function. Using this model and minimization techniques we can prune the search space to focus on the most interesting areas. This process greatly minimizes the number of measurements required to find a suitable candidate. We will evaluate this methodology studying the optimization of a Laplacian computing kernel.
Joint work with Steven Quinito Masnada, Arnaud Legrand, Frédéric Desprez.

6.2.3 Runtime and Hardware Assisted Performance Analysis

Ronak Buch(UIUC)

Abstract:
Modern runtimes and hardware allow the measurement of many different properties of programs. In this talk, we will discuss techniques to use some of these properties to analyze the performance of parallel programs in detail, specifically, source level cache latency and network performance.

6.2.4 Moya: A JIT Compiler for HPC

Tarun Prabhu (UIUC)

Abstract:
When developing real-world HPC applications, one often finds that the compiler either cannot or will not optimize code that seems fairly easy to optimize. Part of the reason for this is that compilers often have to make optimization decisions with incomplete information about the code. For instance, a compiler cannot vectorize a loop nest unless it can prove that the arrays being accessed in the nest do not alias. Since there is a strict requirement that a compiler always produce correct code, in situations where it cannot prove the safety of an optimization, the compiler simply gives up and produces unoptimized code. Often, the safety of these same transformations can be much more easily determined with runtime information. For instance, at runtime, we know the actual addresses and sizes of the arrays being accessed in a loop nest and we can easily determine whether the accesses may alias. By using JIT compilation i.e. re-compiling all/part of the code at runtime, the compiler can use this runtime information to make better decisions regarding both the safety and profitability of optimizations. Moya is an annotation-driven JIT compiler for C, C++ and Fortran built on LLVM. It uses a combination of a small number of programmer-annotations and aggressive static analysis to perform dynamic optimizations. In this talk, we will show how Moya can be used with minimal programmer effort to annotate existing code bases and improve performance.

6.2.5 Analysing some Fiber Miniapps

Judit Gimenez (BSC)

Abstract:
This talk will present the first results analysing some of the Fiber miniapps developed by Riken using the BSC performance tools. The traces have been collected on both Riken and BSC platforms.

6.2.6 Striving for extreme scalability and big data capability using a novel flexible development, benchmarking and runcontrol framework

Wendy Sharples (JSC)

Abstract:
State of the art geoscience simulations are tending towards ever increasing model complexity, due to the incorporation of multi-scale physics, fully coupled model systems and higher spatial resolutions, which in turn require longer run times

and produce larger volumes of output. In order to minimize run times and output data produced, the use of state of the art hardware; such as accelerators and improved supercomputer architectures, in addition to state of the art software; such as the latest Linear Algebra Solver libraries and parallel I/O libraries, is essential. In many cases, to take advantage of the latest developments in HPSC hardware and software, to ensure optimal scalability across HPSC architectures, to ensure the challenge of big data is not insurmountable, and to minimize wasted resources, established geoscience codes need to be continually assessed for potential bottlenecks or areas for improvement. Following this assessment, a strategic solution can then be implemented. So that we can assess the software, in this case ParFlow, effectively, we have developed a set of profiling scripts using JUBE, a benchmarking environment which provides a framework to easily create benchmark sets, run those sets on different computer systems and evaluate the results. Performance analysis tools such as Score-P, Scalasca, Allinea, Darshan, Extrae and Paraver have been incorporated into our benchmarking suite, so that we can analyze ParFlow accurately, efficiently and methodically. Even though ParFlow is one of the few parallel watershed flow model codes capable of modeling continental scale 3D variably saturated flow, there is still room for improvement. To that end, we have identified a few bottlenecks and are currently implementing solutions which address these, namely, parallel I/O using netCDF4 to reduce the volume of input and output, incorporation of the GPU linear algebra library, PETSc, to speed up run times, adaptive mesh refinement to minimize processor load imbalance and in-situ visualization to minimize the time taken to post process the output produced. In addition, we have developed a complete set of run control scripts using JUBE for the whole model pipeline: input data pre-processing, model run and output data post-processing, which will eventually be used as a standard way to run ParFlow models and run regression and unit tests. Joint work with Klaus Goergen, Stefan Kollet, Lukas Poorthuis, Ilya Zukov, Jose Fonseca, Sebastian Luers, Jessica Keune.

6.2.7 The BatSim simulator

Olivier Richard (INRIA)

Abstract:
BatSim is a Resource and Job Management System (RJMS) framework simulator based on SimGrid. It aims at taking into account platform's hardware capabilities and impacts in simulations. Also, schedulers parts are plugable through a comprehensive API and they are seen as external component of the framework.

7 PROGRAMMING MODELS & RUNTIME

7.1 Project talks

7.1.1 Improving hybrid MPI and OmpSs applications using Argobots

Sergi Mateo Bellido (BSC)

Project: Enhancing Asynchronous Parallelism in OmpSs with Argobots

Abstract:
This talk will present the progress done on the "Enhancing Asynchronous Parallelism in OmpSs with Argobots" project, where we have implemented the OmpSs tasking model on top of the Argobots threading library. In this talk we will explain our latest experiments that show how the integration between the Argobots threading library and the MPICH MPI library can improve both programmability and performance of hybrid MPI and OmpSs applications. Moreover, we also propose and compare this close integration of Argobots and MPICH with an alternative method based on MPI call interception.

7.1.2 Overview of Charm++ and collaboration opportunities

Sanjay Kale (UIUC)

Project: Energy Efficiency and Load Balancing

Abstract:
Charm++ is an established parallel programming system, with an adaptive runtime system (RTS) as its signature strength. It is a C++ based system. Parallel computations are specified as an interacting collections of objects, called chares. Each collection can be indexed using a separate index structure (such as 1D, 6D-sparse, bit-vector, etc.). The interactions are specified as asynchronous method invocations, leading to a message driven execution model. This has automatic benefits for adaptive overlap of communication and computation. Placement of chares to processors is controlled by the adaptive runtime system, which allows it to do dynamic load balancing. The RTS is also capable of tolerating faults and optimizing for energy, power and temperature. The RTS collaborates with whole-machine job scheduling and power optimizations, via its ability to shrink and expand the sets of processors used by a job, for example. We are seeking application collaborations, as well as collaborations for runtime strategies. Within-node scheduling is a relatively new challenge, and we hope to work with many of you in this area.

7.2 Lightning talks

7.2.1 The Multi-Stencil Language: orchestrating stencils with a mesh-agnostic DSL

Hélène Coullon (INRIA)

Abstract:
As the computation power of modern high performance architectures increases, their heterogeneity and complexity also become more important. One of the big challenges of exascale is to get programming models which gives access to high performance computing (HPC) to many scientists and not only to a few HPC specialists. One relevant solution to ease parallel programming for scientists is Domain Specific Language (DSL). However, one problem to avoid with DSLs is to not design a new DSL each time a new domain or a new problem has to be solved. This phenomenon happens for stencil-based numerical simulations, for which a large number of languages has been proposed without code reuse between them. The Multi-Stencil Language (MSL) is a language common to any kind of mesh used into a stencil-based numerical simulation. It is said that MSL is mesh-agnostic. Thus, MSL, by finding a common language for different kinds of stencil-based simulation, facilitates code reuse. MSL is evaluated on a real case simulation which solves shallow-water equations.

7.2.2 Lightweight Kernel-Assisted Communication Progression

Alexandre Denis (INRIA)

Abstract:
The advances in processor architecture has brought computing into the multicore era. A typical cluster nowadays is made of nodes comprising tens of cores. To program such machines, we are facing multiple challenges regarding communications and jitter. At large scale, the cost of communication becomes the prominent overhead in HPC applications; to amortize this cost, a common solution consists in overlapping communications with computations. With a large number of nodes and threads that interact and synchronize, jitter may jeopardize the speedup with slowest threads slowing down the whole application; to overcome this risk, a solution consists in using low jitter light-weight kernels (LWK) dedicated to HPC. However, at first sight these solutions may look contradictory since usually communication asynchronous progression involves threads that bring non-determinism and jitter. In this project, we will propose mechanisms to obtain asynchronous progression of communications in a LWK that do not disturb application thread scheduling. This solution combines IHK/McKernel (RIKEN, Tokyo) as LWK and MadMPI+Pioman (INRIA, Bordeaux) for communications. Our plan is to port MadMPI+Pioman on IHK/McKernel so as to get communication progression on this platform, using hooks in McKernel to trigger Pioman activity.

7.2.3 A New Parallel Execution Model for Many-Core Architectures

Atsushi Hori (RIKEN)

Abstract:
A new parallel execution model, named Partitioned Virtual Address Space (PVAS) is proposed. In this talk, a brief introduction of PVAS followed by ongoing several research collaborations and possible research collaboration targets will be addressed.

7.2.4 AutoMOMML: Automatic Multi-Objective Modeling with Machine Learning

Prasanna Balaprakash (ANL)

Abstract:
In recent years, automatic data-driven modeling with machine learning (ML) has received considerable attention as an alternative to analytical modeling for many modeling tasks. While ad hoc adoption of ML approaches has obtained success, the real potential for automation in data-driven modeling has yet to be achieved. In this talk, we will describe AutoMOMML, an end-to-end, ML-based framework to build predictive models for objectives such as performance, and power. The framework adopts statistical approaches to reduce the modeling complexity and automatically identifies and configures the most suitable learning algorithm to model the required objectives based on hardware and application signatures. The experimental results using hardware counters as application signatures show that the median prediction error of performance, processor power, and DRAM power models are 13%, 2.3%, and 8%, respectively.

7.2.5 Leveraging the OmpSs + Charm++ Programming Model for Stencil Computations

Marc Casas (BSC)

Abstract:
Numerical kernels based on stencil computations can be easily implemented with the Charm++ programming model since it lets to easily break down the physical domain into several subdomains and process them in parallel. In this scenario, each subdomain can be assigned to a chare and data are exchanged between chares that process neighbors. However, it is hard to overlap these data transfers with computation and, even more, to combine computations that belong to different iterations with purely Charm++ codes. OmpSs can be then used to carry out such overlaps and thus improve the scalability of stencil computations. In this talk we will show some relevant examples to illustrate the benefits of having OmpSs and Charm++ cooperating.

7.2.6 Domain Specific Languages on HPC systems

Sergi Mateo Bellido (BSC)

Abstract:
The talk will present our experiences developing a framework to build Domain Specific Languages for HPC platforms. This multi-layer framework is based on LMS for embedding DSLs and OpenMP and MPI for running on parallel and distributed environments. Our framework has been successfully used to implement a DSL for solving problems that can be modeled as a set of Partial Differential Equations (PDEs). This specific DSL will be used to describe our software architecture and the different challenges and opportunities we identified to effectively develop DSLs for HPC systems.

7.2.7 BOLT: OpenMP over Lightweight Threads

Sangmin Seo (ANL)

Abstract:
In this presentation, we will introduce BOLT, which targets a high-performing OpenMP implementation, especially specialized for nested or fine-grained parallelism. Unlike other OpenMP implementations, BOLT utilizes a lightweight threading model for its underlying threading mechanism. It currently adopts Argobots, a new holistic, low-level threading and tasking runtime, in order to overcome shortcomings of conventional OS-level threads. Its runtime and compiler are based on the OpenMP runtime and Clang in LLVM, respectively. This talk will present the design and implementation of BOLT as well as preliminary experimental results with some applications.

7.2.8 Adaptive MPI: Overview & Potential Collaborations

Samuel T. White (UIUC)

Abstract:
Adaptive MPI (AMPI) is an implementation of the MPI standard written on top of Charm++. AMPI provides the adaptive runtime features of Charm++, such as overdecomposition, dynamic load balancing, and online fault tolerance, to MPI programmers. This talk gives a high-level overview of AMPI, its features, recent improvements, and application use cases, while also suggesting grounds for collaborations in its use and research.

7.2.9 Energy-Aware Autotuning for HPC Kernels

Luís Felipe Garlet Millani (INRIA)

Abstract:
Energy consumption is a growing concern in HPC. Increasingly complex platforms with several cores and accelerators are difficult to optimize for, both in GFLOPS and GFLOPS/Watt. Autotuners are often used to broaden the search space and obtain portable performance across these different platforms. It's interesting for autotuners to also consider the energy efficiency of the solutions. BOAST is a framework for comparing the time efficiency of different implementations of a computing kernel. We extend the BOAST framework to also consider the energy efficiency of the different implementations. This gives the user a better understanding of the time-energy trade-offs. In this work we present the time-energy tradeoffs of different optimizations found semi-automatically by BOAST for a few HPC kernels. Joint work with Brice Videau, François Broquedis, Lucas Mello Schnorr and Jean-François Méhaut.

8 RESILIENCE

8.1 Project talks

8.1.1 New Techniques to Design Silent Data Corruption Detectors

Leonardo Bautista Gomez (BSC)

Project: New Techniques to Design Silent Data Corruption Detectors

Abstract:
In this talk, we will present our ongoing work on the exploration on different techniques for silent error detectors for HPC applications. We will present our focus on Machine Learning algorithms but also some results that show that not all Machine Learning algorithms can be applied efficiently on-line. In addition, we will present our current work on the broader problem of adapting or choosing a best-fit silent corruption detector based on the application data and the properties of the detectors.

8.1.2 When Amdahl Meets Young/Daly

Aurélien Cavelan (INRIA)

Project: Optimization of Fault-Tolerance Strategies for Workflow Applications

Abstract:
This paper investigates the optimal number of processors to execute a parallel job, whose speedup profile obeys Amdahl's law, on a large-scale platform subject to fail-stop and silent errors. We combine the traditional checkpointing and rollback recovery strategies with verification mechanisms to cope with both error sources. We provide an exact formula to express the execution overhead incurred by a periodic checkpointing pattern of length T and with P processors, and we give first-order approximations for the optimal values T^* and P^* as a function of the individual processor failure rate λ. A striking result is that P^* is of the order $\lambda^{-1/4}$ if the checkpointing cost grows linearly with the number of processors, and of the order $\lambda^{-1/3}$ if the checkpointing cost stays bounded for any P. We conduct an extensive set of simulations to support the theoretical study. The results confirm the accuracy of first-order approximation under a wide range of parameter settings.

8.1.3 Lossy Compression for HPC Checkpoint Restart: Mathematical Guidance for Error Tolerance Selection

Jon Calhoun (UIUC)

Project: Checkpoint/Restart of/from lossy state

Abstract:
Long running HPC applications depend on checkpoint restart to recover from failures and utilize multiple time allocations. Memory bandwidth and in particular file system bandwidth continues to be limiters on application performance

and scalability. Compression techniques can be used to reduce data size limiting its impact. Lossless compression fails to generate high compression factors, but lossy compression generates noticeably higher compression factors at the expense of adding a small but controllable amount of error into the simulation. In this talk, we discuss how knowledge of the numerical method used to advance the simulation and the problem's spatial discretiazations can be utilized in the selection of lossycompression error tolerances. Error tolerances can be selected allowing only mathematically insignificant error into the simulation, or assigned to indicatie the accuracy of a checkpoint relative to an lower-ordered numerical methods. We demonstrate this selection methodology on two production HPC applications PlasComCM and Nek5000 and show simulation accuracy is preserved. Finally, we highlight how physical properties and selection of boundary conditions can be employed to remove any non-significant error.

8.2 Lightning talks

8.2.1 Optimal Multi-Level Checkpointing

Hongyang Sun (INRIA)

Abstract:
We present an optimal first-order approximation of a multi-level checkpointing protocol based on periodic computing patterns, giving explicit formulas on the checkpointing interval, the number of checkpoints for each level, as well as the execution overhead.
New Techniques to Design Silent Data Corruption Detectors

8.2.2 Detecting Silent Data Corruptions with Error Estimations in Numerical Integration Solvers

Pierre-Louis Guhur (ANL)

Abstract:
Numerical integration solvers are particularly sensitive to corruptions, because 1) as a step-by-step method, they propagate a corruption all along the resolution, 2) when unstable, the solutions could even diverge. However, they have the advantage that their approximation error can be computed at a low cost. We used these error estimates for improving detection performance in HPC applications. First, we proposed a new lightweight detector for solvers with a fixed integration size. We mathematically showed that all corruptions affecting the accuracy of a simulation are detected by our method. Secondly, we showed that the inherent rejection mechanism of solvers with a variable integration size is not reliable enough. We designed a mechanism to improve it. Experiments were done in PETSc on the Blues cluster with 4096 cores. I will also present early results on reducing IO bandwidth of checkpoint-restart mechanism. Thanks to error estimates, we can control the error bound of lossy compressors.

8.2.3 A different re-execution speed can help

Valentin Le Fèvre (INRIA)

Abstract:
We consider divisible load scientific applications executing on large-scale platforms subject to silent errors. While the goal is usually to complete the execution as fast as possible in expectation, another major concern is energy consumption. The use of dynamic voltage and frequency scaling (DVFS) can help save energy, but at the price of performance degradation. Consider the execution model where a set of K different speeds is given, and whenever a failure occurs, a different re-execution speed may be used. Can this help? We address the following bi-criteria problem: how to compute the optimal checkpointing period to minimize energy consumption while bounding the degradation in performance. We solve this bi-criteria problem by providing a closed-form solution for the checkpointing period, and demonstrate via a comprehensive set of simulations that a different re-execution speed can indeed help.

8.2.4 The new Buddy-Checkpointing Feature of SIONlib for Task-Local Parallel I/O Support

Wolfgang Frings (JSC)

Abstract:
In this talk we will present the new buddy-checkpointing feature of SIONlib, which is designed to support checkpointing of simulation data on node-local storage for application that uses a task-local I/O pattern. With this new feature, SIONlib is able to store data in a virtual shared file container on local storage of the compute nodes, whereas local data is automatically and transparently mirrored to local storage of a set of buddy nodes. The buddy-checkpointing feature of SIONlib has been developed within the EU-project DEEP-ER to support task-local I/O on hierarchical I/O infrastructures. Additionally, SIONlib can work together with the multi-level checkpoint library SCR to support buddy-checkpointing for SIONlib file container.

8.2.5 Computing on Unprotected Memory : Opportunity for a world-wide collaboration

Leonardo Bautista Gomez (BSC)

Abstract:
A very recent study carried on a machine with low-power unprotected memory devices has shed new light over the character of errors in memory. The new findings show that a certain number of correlations exist and could be exploited to detect or avoid silent data corruption. In order to substantially increase our knowledge about errors in memory I will propose an idea to develop a world-wide memory vulnerability detector.

8.2.6 Programmer-directed Partial Replication for Fault-tolerant HPC Applications

Omer Subasi (BSC)

Abstract:
We investigate partial task replication and checkpointing for task-parallel HPC

applications to mitigate silent data corruption (SDC) errors. As the complete replication of all application tasks can be prohibitive due to resource costs, we explore programmer-directed selective replication mechanism to provide fault-tolerance while decreasing costs.