joint-lab workshop Jun. 9-11 2014

UNDER construction: The agenda below is not the final one

This event is supported by INRIA, UIUC, NCSA, ANL, BSC, PUF NEXTGEN,

Main Topics	Schedule	Speaker	Affiliation	Type of presentation	Title (tentative)	Download
	Sunday June 8th					
Dinner Before the Workshop	7:30 PM	Only people registered for the dinner (included)			Mercure Hotel	
Workshop Day 1	**Monday June 9th**					
					TITLES ARE TEMPORARY (except if in bold font)	
Registration	08:00	At Inria Sophia Antipolis				
Welcome and Introduction **Amphitheatre**	08:30	Franck Cappello + Marc Snir + Yves Robert + Bill Kramer + Jesus Labarta	INRIA&UIUC&ANL&BSC	Background	**Welcome, Workshop objectives and organization**	
Plenary **Amphitheatre** **Chair: Franck Cappello**	09:00	Jesus Labarta	BSC	Background	**Presentation of BSC activities**	
Mini Workshop **Applied Maths.** **Amphitheatre**						
Chair: Paul Hovland	09:30	Bill Gropp	UIUC		**Advancing Toward Exascale: Some Results and Opportunities**	
	10:00	Jed Brown	ANL		**Next-generation multigridding: adaptivity and communication avoidance**	
	10:30	Break				
	11:00	Ian Masliah	Inria		**Automatic generation of dense linear system solvers on CPU/GPU architectures**	
	11:30	Luke Olson	UIUC		**Reducing Complexity in Algebraic Solvers**	
	12:00	Lunch				
Chair: Bill Gropp	13:30	Vincent Baudoui	Inria		**Round-off error propagation in large-scale applications**	
	14:00	Paul Hovland	ANL		**Checkpointing with Multiple Goals**	
	14:30	Stephane Lanteri	Inria		**C2S@Exa: a multi-disciplinary initiative for high performance computing in computational sciences**	
Mini Workshop **I/O and BigData** **Amphitheatre**						
Chair: Rob Ross	15:00	Wolfgang Frings	JSC		**HPC I/O at Large Scale with SIONlib and Spindle**	
	15:30	Break				
	16:00	Jonathan Jenkins	ANL		**Towards Simulating Extreme-scale Distributed Systems**	

	16:30	Matthieu Dorier	Inria		**Omnisc'IO: A Grammar-Based Approach to Spatial and Temporal I/O Patterns Prediction**	
	17:00	Dave Mattson Kenton Guadron McHenry,	NCSA		**The NCSA Image and Spatial Data Analysis Division**	
	17:30	Adjourn				
	18:30	**Bus for dinner** (dinner included)				
Mini Workshop **Runtime** **Room Gilles Kahn**						
Chair: Jesus Labarta	9:30	Pavan Balaji	ANL		**VOCL: A Virtualization Infrastructure for Accelerators**	
	10:00	Augustin Degomme and Arnaud Legrand	Inria		**Status Report on the Simulation of MPI Applications with SMPI/SimGrid**	
	10:30	Break				
	11:00	Ronak Buch	UIUC		**Advanced Techniques in Parallel Performance Analysis**	
	11:30	Victor Lopez	BSC		**DLB: Dynamic Load Balancing Library**	
	12:00	Lunch				
Chair: Rajeev Thakur	13:30	Xin Zhao	ANL		**Programming Runtime Support for Irregular Computations**	
	14:00	Luka Stanisic and Arnaud Legrand	Inria		**Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures**	
	14:30	Pieter Bellens	BSC		**Quantifying the effect of rectangular blocks in the dense QR factorization**	
	15:00	Lucas Nussbaum	Inria		**Evaluating exascale HPC runtimes through emulation with Distem**	
	15:30	Break				
Chair: Sanjay Kale	16:00	Francois Tessier	Inria		**Distributed communication-aware load balancing with TreeMatch in Charm++**	
	16:30	Jean-François Mehaud	Inria		**Saving Energy by Exploiting Residual Imbalance on Iterative Applications**	
	17:00	Juan González	Inria		**Performance Analytics: Understanding Parallel Applications using Cluster Analysis and Sequence Analysis.**	
	17:30	Adjourn				
	18:30	**Bus for dinner** (dinner included)				
Workshop Day 2	**Tuesday June 10th**					
Formal opening **Amphitheatre** **Chair: Bill Kramer**	08:30	Marc Snir + Franck Cappello	INRIA&UIUC&ANL	Background		
	08:40	Claude Kirchner	Inria	Background	**Inria updates and vision of the collaboration**	TBD
	08:50	Marc Snir	ANL	Background	**ANL updates vision of the collaboration**	TBD
Plenary **Amphitheatre**	09:00	Wolfgan Frings	JSC	Background	**JSC activities in HPC**	TBD

Mini Workshop **I/O and Big Data** **Amphitheatre**						
Chair: Gabriel Antoniu	09:30	Rob Ross	ANL		**Understanding and Reproducing I/O Workloads**	
	10:00	Guillaume Aupy	Inria		**Scheduling the I/O of HPC applications under congestion**	
	10:30	Break				
	11:00	Lokman Rahmani	Inria		**Smart In Situ Visualization for Climate Simulations**	
	11:30	Anthony Simonet	Inria		**Using Active Data to Provide Smart Data Surveillance to E-Science Users**	
	12:00	Lunch				
Mini Workshop **Runtime** **Room Gilles Kahn**						
Chair: Jean François Mehaut	09:30	Sanjay Kale	UIUC		**Temperature, Power and Energy: How an Adaptive Runtime can optimize them**	
	10:00	Florentino Sainz	BSC		**DEEP Collective offload**	
	10:30	Break	Inria			
	11:00	Brice Videau	Inria		**Porting HPC applications to the Mont-Blanc prototype using BOAST**	
	11:30	Grigori Fursin	Inria		**Collective Mind: bringing reproducible research to the masses**	
	12:00	Lunch				
Plenary **Amphitheatre** **Chair: Franck Cappello**	13:45	Ed Seidel	UIUC	Background	**NCSA updates and vision of the collaboration**	
Plenary **Amphitheatre** **Chair: Wolfgan Frings**	14:00	Yves Robert	Inria		**Algorithms for coping with silent errors**	
	14:30	Marc Snir	ANL		**Runtime and OS research at DoE**	
	15:00	Break				
Mini Workshop **Resilience** **Amphitheatre**						
Chair: Franck Cappello	15:30	Luc Jaulmes	BSC		**Checkpointless exact recovery techniques for Krylov-based iterative methods**	
	16:00	Ana Gainaru	UIUC		**The road to failure prediction on Blue Waters: latest details and future directions**	
	16:30	Tatiana Martsinkevich	Inria		**Using dedicated resources to alleviate memory limitation for message logging protocols**	
	17:00	Adjourn				

Mini Workshop **Cloud & Cyber-infrastructure** **Room Gilles Kahn**						
Chair: Kate Keahey	15:30	Justin Wozniak	ANL		**Case Studies in Big Data and HPC from X-ray Crystallography**	
	16:00	Shaowen Wang	UIUC		**CyberGIS @ Scale**	
	16:30	Christine Morin	Inria		**Contrail: Interoperability and Dependability in a Cloud Federation**	
	17:00	Adjourn				
	18:30	**Bus for Dinner (dinner included)**				
Workshop Day 3	**Wednesday June 11th**					
Plenary **Amphitheatre** **Chair: Marc Snir**	8:30	Bill Kramer	NCSA		**Blue Waters - A year of results and insights**	
Mini Workshop **Resilience** **Amphitheatre**						
Chair: Yves Robert	9:00	Leonardo Bautista Gomez	ANL		**Fault Tolerance Interface new features and new developments**	
	9:30	Slim Bouguerra	Inria		**Energy-Performance Tradeoffs in Multilevel Checkpoint Strategies**	
	10:00	Break				
	10:30	Martin Quinson	Inria		**Formal verification of unmodified MPI applications with SimGrid**	
Plenary **Amphitheatre**	11:00	**Closing**				
	12:00	Lunch (included)				
Mini Workshop **Cloud & Cyber-infrastructure** **Room Gilles Kahn**						
Chair: Justin Wozniak	09:00	Kate Keahey	ANL			
	09:30	Radu Tudoran	Inria		**JetStream: Enabling High Performance Event Streaming across Cloud Data-Centers**	
	10:00	Break				
	10:30	Timothy Armstrong	ANL		**Towards Dynamic Dataflow Composition for Extreme-Scale Applications with Heterogeneous Tasks**	
Plenary **Amphitheatre**	11:00	**Closing**				
	12:00	Lunch (included)				

Abstract

Matthieu Dorier

Title: Omnisc'IO: A Grammar-Based Approach to Spatial and Temporal I/O Patterns Prediction

The increasing gap between the computation performance of post-petascale machines and the performance of their I/O subsystem has motivated many I/O optimizations including prefetching, caching, and scheduling techniques. To further improve these techniques, modeling and predicting spatial and temporal I/O patterns of HPC applications as they run have become crucial.

This presentation introduces Omnisc'IO, an original approach that aims to make a step forward toward an intelligent I/O management of HPC applications in next-generation post-petascale supercomputers. It builds a grammar-based model of the I/O behavior of any HPC application and uses that model to predict when future I/O operations will occur, as well as where and how much data will be accessed. Omnisc'IO is transparently integrated into the POSIX and MPI I/O stacks and does not require any modification to application sources or to high level I/O libraries. It works without prior knowledge of the application, and converges to accurate predictions within a couple of iterations only. Its implementation is efficient both in computation time and in memory footprint. Omnisc'IO was evaluated with four real HPC applications -- CM1, Nek5000, GTC, and LAMMPS -- using a variety of I/O backends ranging from simple POSIX to Parallel HDF5 on top of MPI I/O. Our experiments show that Omnisc'IO achieves from 79.5% to 100% accuracy in spatial prediction and an average precision of temporal predictions ranging from 0.2 seconds to less than a millisecond.

Sheng Di

Optimization of Multi-level Checkpoint Model with Uncertain Execution Scales

As for future extreme scale systems, there could be different types of failures striking exa-scale applications with different failure scales, from transient uncorrectable memory errors in processes to massive system outages. In this work, a multi-level checkpoint model is proposed by taking into account uncertain execution scales (different numbers of processes/cores). The contribution is three-fold. (1) We provide an in-depth analysis on why it is very tough to derive the optimal checkpoint intervals for different checkpoint levels and optimize the number of cores simultaneously. (2) We devise a novel method which can quickly obtain an optimized solution, which is the first successful attempt in the multi-level checkpoint model with uncertain scales. (3) We perform both large-scale real experiments and extreme-scale numerical simulation to validate the effectiveness of our design. Experiments confirm our optimized solution outperforms other state-of-the-art solutions by 4.3-88% on wall-clock length.

Augustin Degomme/Arnaud Legrand

Status Report on the Simulation of MPI Applications with SMPI/SimGrid

- Virtualisation: The automatic approaches we had for application emulation required to rely on an alternative compiling chain (e.g., using GNU TLS), which is problematic as it could dramatically changes code performance and was not sufficiently generic. We have looked forward alternative approaches and have recently designed a new one based on the OS-like organization of SimGrid that allows us to identify heaps and stacks of virtual MPI process and to mmap them whenever context switching. This new approach enables to *emulate unmodified MPI applications* regardless of the language with which they are written and regardless of the compiling toolchain. Although this has not been evaluated yet, this approach should also allow to use classical profilers at small scale to identify which variables should be aliased and which kernels should be modeled rather than truly executed in simulation.

- Trace replay and interoperability: we have a current effort toward SMPI interoperability. Each simulation tool (BigSim, LogGOPSIM, Dimemas, SimGrid, SST/Macro ...) has its own strength and weaknesses but is often stronly biased toward a given tracing format. Working toward interoperability would allow researchers to seamlessly move to another simulator whenever it is more appropriate rather than trying to fix the one linked to its tracing tool or to its application. Replaying BigSim and scalatrace traces is now possible in SMPI/SimGrid but the validation remains to be done. We have plans to perform similar work with Dimemas and SST/Macro so as to ease the use of SimGrid's fluid models.

- Status report and current effort on network modeling (IB, fat-tree and torus-like topologies).

Luka Stanisic/Arnaud Legrand

Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures

[Joint work between Luka Stanisic, Samuel Thibault, Arnaud Legrand, Brice Videau and Jean-François Méhaut, accepted for publication at Europar'14]

Multi-core architectures comprising several GPUs have become mainstream in the field of High-Performance Computing. However, obtaining the maximum performance of such heterogeneous machines is challenging as it requires to carefully offload computations and manage data movements between the different processing units. The most promising and successful approaches so far rely on task-based runtimes that abstract the machine and rely on opportunistic scheduling algorithms. As a consequence, the problem gets shifted to choosing the task granularity, task graph structure, and optimizing the scheduling strategies. Trying different combinations of these different alternatives is also itself a challenge. Indeed, getting accurate measurements requires reserving the target system for the whole duration of experiments. Furthermore, observations are limited to the few available systems at hand and may be difficult to generalize. In this research report, we show how we crafted a coarse-grain hybrid simulation/emulation of StarPU, a dynamic runtime for hybrid architectures, over SimGrid, a versatile simulator for distributed systems. This approach allows to obtain performance predictions accurate within a few percents on classical dense linear algebra kernels in a matter of seconds, which allows both runtime and application designers to quickly decide which optimization to enable or whether it is worth investing in higher-end GPUs or not.

Guillaume Aupy

Scheduling the I/O of HPC applications under congestion

A significant percentage of the computing capacity of large-scale platforms is wasted due to interferences incurred by multiple applications that access a shared parallel file system concurrently. One solution to handling I/O bursts in large-scale HPC systems is to absorb them at an intermediate storage layer consisting of burst buffers. However, our analysis of the Argonne's Mira system shows that burst buffers cannot prevent congestion at all times. As a consequence, I/O performance is dramatically degraded, showing in some cases a decrease in I/O throughput of 67%. In this paper, we analyze the effects of interference on application I/O bandwidth, and propose several scheduling techniques to mitigate congestion. We show through extensive experiments that our global I/O scheduler is able to reduce the effects of congestion, even on systems where burst buffers are used, and can increase the overall system throughput up to 56%. We also show that it outperforms current Mira I/O schedulers.

Florentino Sainz

DEEP Collective offload

Abstract: We present a new extension of OmpSs programming model which allows users to dynamically offload C/C++ or Fortran code from one or many nodes to a group of remote nodes. Communication between remote nodes executing offloaded code is possible through MPI. It aims to improve programmability of Exascale and nowadays supercomputers which use different type of processors and interconnection networks which have to work together in order to obtain the best performance. We can find a good example of these architectures in the DEEP project, which has two separated clusters (CPUs and Xeon Phis). With our technology, which works in any architecture which fully supports MPI, users will be able to easily offload work from the CPU cluster to the accelerators cluster without the constraint of falling back to the CPU cluster in order to perform MPI communications.

Radu Tudoran

JetStream: Enabling High Performance Event Streaming across Cloud Data-Centers

The easily-accessible computation power offered by cloud infrastructures coupled with the revolution of Big Data are expanding the scale and speed at which data analysis is performed. In their quest for finding the Value in the 3 Vs of Big Data, applications process larger data sets, within and across clouds. Enabling fast data transfers across geographically distributed sites becomes particularly important for applications which manage continuous streams of events in real time. In this paper, we propose a set of strategies for efficient transfers of events between cloud data-centers. Our approach, called, JetStream, is able to self-adapt to the streaming conditions by modeling and monitoring a set of context parameters. It further aggregates the available bandwidth by enabling multi-route streaming across cloud sites. The prototype was validated on tens of nodes from US and Europe data-centers of the Microsoft Azure cloud using synthetic benchmarks and with application code from the context of the Alice experiment at CERN. The results show an increase in transfer rate of 250 times over individual event streaming. Besides, introducing an adaptive transfer strategy brings an additional 25% gain. Finally, the transfer rate can further be tripled thanks to the use of multi-route streaming.

Anthony Simonet

Using Active Data to Provide Smart Data Surveillance to E-Science Users

Modern scientific experiments often involve multiple storage and computing platforms, software tools, and analysis scripts. The resulting heterogeneous environments make data management operations challenging; the significant number of events and the absence of data integration makes it difficult to track data provenance, manage sophisticated analysis processes, and recover from unexpected situations. Current approaches often require costly human intervention and are inherently error prone. The difficulties inherent in managing and manipulating such large and highly distributed datasets also limits automated sharing and collaboration.
We study a real world e-Science application involving terabytes of data, using three different analysis and storage platforms, and a number of applications and analysis processes. We demonstrate that using a specialized data life cycle and programming model---Active Data---we can easily implement global progress monitoring, and sharing; recover from unexpected events; and automate a range of tasks.

Ian Ma

Automatic generation of dense linear system solvers on CPU/GPU architectures
The increasing complexity of new parallel architectures has widened the gap between adaptability and efficiency of the codes. As high performance numerical libraries tend to focus more on performance, we wish to address this issue using a C++ library called NT2. By analyzing the properties of the linear algebra domain that can be extracted from numerical libraries like LAPACK and MAGMA and combining them with architectural features, we developed a generic approach to solve dense linear systems on hybrid architectures. We report performance results that correspond to what state-of-the-art codes achieve while maintaining a generic code that can run either on CPU or GPU.

Dave Mattson and Kenton Guadron McHenry
The NCSA Image and Spatial Data Analysis Division

The Image and Spatial Data Analysis division conducts research and development in general purpose data cyberinfrastructure, addressing specifically the growing need to make use of large collections of non-universally accessible, or individually-managed, data and software (i.e. executable data). We attempt to address these needs through the development of a common suite of internally and externally created open source tools/platforms that provide means of auto and assisted curation for data/software collections. To acquire some of the needed high level metadata not provided with un-curated data we make heavy use of techniques founded in artificial intelligence, machine learning, computer vision, and natural language processing. To close the gap between the state of the art of these fields and current needs, while also providing a sense of oversight many of our domain users desire, we attempt to keep the human in the loop wherever possible by incorporating elements of social curation, crowd sourcing, and error analysis. Given the ever growing urgency to gain benefit from the deluge of un-curated data we push for the adoption of solutions derived from these relatively young fields, highlighting the value of having tools to deal with this data where there would be nothing otherwise. Attempting to follow in the footsteps of the great software cyberinfrastructure successes of NCSA (i.e. mosaic, httpd, and telnet) we attempt to address these scientific and industrial needs in a manner that is also applicable to the general public. By catering toward broad appeal rather than focusing on a niche within the total possible users we aim at stimulating uptake and providing a life for our software solutions beyond funded project deliverables. We will briefly go over a handful of our current projects spanning data integration and visualization, data mining, and the creation of general purpose software tools.

Bill Kramer

Blue Waters - A year of results and insights

This talk will discuss the first year of full service for Blue Waters, including highlights of science and results and well as insights into the use of the systems. The talk will also point to lessons that might be important as we move into the extreme scale era.

Vincent Baudoui

Round-off error propagation in large-scale applications

Round-off errors coming from numerical calculation finite precision can lead to catastrophic losses in significant numbers when they accumulate. They will become more and more overriding in the future as the problem size increases with the refinement of numerical simulations. Existing analytical bounds for round-off errors are known to be poorly scalable and they become quite useless for large problems. That is why the propagation of round-off errors throughout a computation needs to be better understood in order to ensure large-scale application results accuracy. We study here a round-off error estimation method based on first order derivatives computed thanks to algorithmic differentiation techniques. It can help following the error propagation through a computational graph and identifying the sensitive sections of a code. It has been experimented on well known LU decomposition algorithms that are widely used to solve linear systems. We will present some examples as well as challenges that need to be tackled as part of future research work in order to set up a strategy to analyze round-off error propagation in large-scale problems.

Luc Jaulmes

Checkpointless exact recovery techniques for Krylov-based iterative methods

By exploiting inherent redundancy in iterative solvers, especially Krylov-subspace methods, we can recover from non-silent errors in data without reverting to techniques like checkpointing. We implemented this recovery scheme for the Conjugate Gradient (CG) and its Preconditioned variant (PCG) and show near-zero overheads without faults, and fast recoveries that preserve all convergence properties of the solver. Using the asynchronous task-based programming model OmpSs, these overheads are even further minimized.

Lokman Rahmani

Smart In Situ Visualization for Climate Simulations

The increasing gap between computational power and I/O performance in new supercomputers has started to drive a shift from an offline approach to data analysis to an inline approach, termed in situ visualization (ISV). While most visualization software now provides ISV, they typically visualize large dumps of unstructured data, by rendering everything at the highest possible resolution. This often negatively impacts the performance of simulations that support ISV, in particular when ISV is performed interactively, as in situ visualization requires synchronization with the simulation. In this work, we advocate for a smarter method of performing ISV. Our approach is data-driven: it aims to detect potentially interesting regions in the generated dataset in order to feed ISV frameworks with “the interesting” subset of the data produced by the simulation. While this method mitigates the load on ISV frameworks by making them more efficient and more interactive, it also helps scientists focus on the relevant part of their data. We investigate smart ISV in the context of a climate simulation, with a set of generic filters derived from information theory, statistics and image processing, and show the tradeoff between performance and quality of visualization.

Lucas Nussbaum

Evaluating exascale HPC runtimes through emulation with Distem

The Exascale era will require the HPC software stack to face important challenges such as platform hetereogeneity and evolution during execution, or reliability issues. We propose a framework to evaluate key aspects of a central part of this software stack: the HPC runtimes. Starting from Distem, which is a versatile emulator for studying distributed systems, we designed an emulator suitable for the evaluation of HPC runtimes, enabling specifically: (1) emulation of a very large scale platform on top of a regular cluster; (2) introduction of heterogeneity and dynamic imbalance among the computing resources; (3) introduction of failures. Those features provide runtime designers with the ability to experiment their prototypes under a large range of conditions, to discover performance gaps, understand future bottlenecks, and evaluate fault tolerance and load balancing mechanisms. We validate the usefulness of this approach with experiments on two HPC runtimes: Charm++ and OpenMPI.

Sanjay Kale

Temperature, Power and Energy: How an Adaptive Runtime can optimize them.

Jonathan Jenkins

Towards Simulating Extreme-scale Distributed Systems

Simulating future extreme-scale parallel/distributed systems can be an important component in understanding these systems at a scale at which prototyping cannot feasibly reach. For HPC, big-data/cloud, or other computing/analysis platforms, the design decisions for developing systems that scale beyond current-generation systems are multi-dimensional in nature. For example, these decisions encompass distributed storage software/hardware solutions, network topologies within and between computing centers, algorithms for data analysis and compute services in heterogeneous software /hardware environments, etc., each of which can potentially be rich targets for exploring via a simulation-based approach. This talk will examine our ongoing work in developing a simulation model framework using parallel discrete event simulation to examine various design aspects of extreme-scale distributed systems. As an exemplar, simulation of protocols used in distributed storage systems will be examined in detail.

Timothy Armstrong

Towards Dynamic Dataflow Composition for Extreme-Scale Applications with Heterogeneous Tasks
Parallel applications are increasingly built from heterogeneous software components that use diverse programming models, such as message-passing, threads, CUDA, and OpenCL on heterogeneous hardware resources such as CPUs and GPUS. Getting these components to interoperate is a challenge in itself, which is further complicated by complex cross-cutting concerns such as scheduling, overlapping of communication and computation, fault-tolerance, and energy efficiency. Parallel execution models offer the hope of making these challenges more managable for application programmers by unifying heterogeneous components into a more uniform framework. One such model is data-driven task parallelism, in which massive numbers of tasks are dynamically assigned to compute resources and communication and synchronization is based on explicit data dependencies. Swift is a high-level scripting
language that provides a simple yet powerful way of expressing data-driven task parallelism. This talk discusses our current progress and future challenges on a compiler and runtime system that allows Swift to scale to hundreds of thousands of cores.

Juan González

Performance Analytics: Understanding Parallel Applications using Cluster Analysis and Sequence Analysis.
Due to the increasing complexity of HPC systems and applications it is strictly necessary to maximize the insight of the performance data extracted from an application execution. This is the mission of the Performance Analytics field. In this talk we introduce two Performance Analytics techniques. First, we demonstrate how it is possible to capture the computation structure of parallel applications at fine grain by using density-based cluster algorithms. Second, we introduce the use of multiple sequence alignment algorithms to asses the quality of this computation structure."

Jed Brown
Next-generation multigridding: adaptivity and communication avoidance

An alternate interpretation of the Full Approximation Scheme (FAS) multigrid method creates relationships between levels that can be exploited to eliminate communication on fine grids, avoid storage of fine grids, avoid "visiting" fine grids away from active nonlinearities, accelerate recomputation from checkpoints, and use fine-to-coarse compatibility to check for silent data corruption in fine grid state. This talk will present the algorithmic structure, new results with ultra-low-communication parallel multigrid, and directions for future research.

Luke Olson

Reducing Complexity in Algebraic Solvers

Algebraic multigrid solvers can be designed to handle a large range of problem types, yielding high convergence with minimal tuning of parameters. Yet, in many situations these robust methods also yield complexities in the sparse matrix cycling that inhibits performance, particularly in parallel. The multigrid solution cycle is modeled effectively through the structure of the sparse matrices in the multigrid hierarchy. In this talk, we highlight a couple of recent strategies that target reducing the solver complexity (particularly in parallel) while attempting to retain the convergence of the iterative solver.

The coarse-level sparse matrices operations are defined through the Galerkin product, R A P — i.e., restriction, operator, and interpolation. Consequently, we look at two methods that reduce this complexity: an approach that filters P and a method that builds a coarse level through a non-Galerkin construction. To this end we first introduce a root-node based approach to multigrid, which can be viewed as a hybrid of classical and aggregation based multigrid methods. We give an overview and show how the complexity and convergence of the multigrid cycle can be controlled through selective filtering in a root-node setting. In addition, we look at a non-Galerkin algebraic framework where we are able to model the performance and note the performance gains in selectively filtering coarse-level operators.

Vincent Baudoui

Round-off error propagation in large-scale applications

Round-off errors coming from numerical calculation finite precision can lead to catastrophic losses in significant numbers when they accumulate. They will become more and more overriding in the future as the problem size increases with the refinement of numerical simulations. Existing analytical bounds for round-off errors are known to be poorly scalable and they become quite useless for large problems. That is why the propagation of round-off errors throughout a computation needs to be better understood in order to ensure large-scale application results accuracy. We study here a round-off error estimation method based on first order derivatives computed thanks to algorithmic differentiation techniques. It can help following the error propagation through a computational graph and identifying the sensitive sections of a code. It has been experimented on well known LU decomposition algorithms that are widely used to solve linear systems. We will present some examples as well as challenges that need to be tackled as part of future research work in order to set up a strategy to analyze round-off error propagation in large-scale problems.

Paul Hovland
Checkpointing with Multiple Goals

Bill Gropp
Advancing Toward Exascale: Some Results and Opportunities
In this talk, I will discuss some results in addressing problems in extreme scale computing that came about from collaborations within the Joint Laboratory on Petascale Computing. I will follow that with a summary of some of my ongoing research projects and challenges that are addressing some of the problems of extreme scale computing, and close with some suggestions for future collaborations.

Wolfgang Frings

HPC I/O at Large Scale with SIONlib and Spindle
Parallel applications often store data in multiple task-local files, for example, to create checkpoints, to circumvent memory limitations, or to record performance data. When operating at very large processor configurations, such applications often experience scalability limitations when the simultaneous creation of thousands of files causes metadata-server contention or simply when large file counts complicate file management or operations on those files even destabilize the file system.

In the first part of the talk we will cover the design principles of SIONlib, a parallel I/O library, which addresses this problem by transparently mapping a large number of task-local files onto a small number of physical files via internal metadata handling and block alignment to ensure high performance.

Dynamic linking has many advantages for managing large code bases, but dynamically linked applications have not typically scaled well on high performance computing systems at large scale. Launching an executable that depends on many dynamic shared objects (DSOs) causes a flood of file system operations at program start-up, when each process in the parallel application loads its dependencies. At large scales, this operation has an effect similar to a site-wide denial-of-service attack, as even large parallel file systems struggle to service so many simultaneous requests.

In the second part of this talk we will present Spindle, a novel approach to parallel loading, which coordinates, transparently to user applications, simultaneous file system operations with a scalable network of cache server processes.

Pavan Balaji

VOCL: A Virtualization Infrastructure for Accelerators
Abstract: In this talk I'll present a light-weight virtualization infrastructure for accelerators called VOCL (Virtual OpenCL). The VOCL framework provides an implementation of OpenCL-1.1 and internally manages accelerators from different vendors using their native OpenCL implementations. It provides transparent access to both local and remote accelerators internally using MPI communication for data movement. This talk will focus on various capabilities such an infrastructure provides including: (1) automatic load balancing capabilities, (2) automatic global system power management, (3) transparent protection from double-bit errors, and (4) utilization of heterogeneous collections of accelerators.

Ronak Buch

Advanced Techniques in Parallel Performance Analysis

Abstract: Analyzing the performance of HPC applications is difficult and often unintuitive. Techniques from the world of serial programming, such as profilers and wall-clock timers, do not fully reveal the properties of parallel programs. To provide an incisive view into performance, tools must be designed with parallelism in mind. This talk will present some advanced analysis techniques tailored specifically for parallelism. These capabilities, including multirun analysis and processor clustering, will be demonstrated using the Projections performance analysis tool.

Jean-François Mehaut

Saving Energy by Exploiting Residual Imbalance on Iterative Applications

Parallel scientific applications have been influencing the way science is done in the last decades. These applications have ever increasing demands in performance and resources due to their greater complexity and larger datasets. To meet these demands, the performance of supercomputers has been growing exponentially, which leads to an exponential growth in power consumption too. In this context, saving power has become one of the main concerns of current HPC platform designs, as future Exascale systems need to consider power demand and energy consumption constraints. Whereas some scientific applications have regular designs that lead to well balanced load distributions, others are more imbalanced due to the fact that they have tasks with different processing demands, which makes it difficult to provide an efficient use of the available resources at the hardware level. In this case, a challenge lies in reducing the energy consumption of the application while maintaining a similar performance. In our work, we focus on reducing the energy consumption of imbalanced applications through a combination of load balancing and Dynamic Voltage and Frequency Scaling (DVFS). Our strategy employs an Energy Daemon Tool to gather power information and a load balancing module that benefits from the load balancing framework available with the CHARM++ runtime system. Our approach differs from the one proposed by Sarood et al. as we employ DVFS as a way to decrease energy consumption after balancing the load, while the latter uses DVFS to regulate temperature and employs load balancing to correct subsequent imbalance.

Grigori Fursin

Collective Mind: bringing reproducible research to the masses

When trying to make auto-tuning practical using common infrastructure, public repository of knowledge, and machine learning (cTuning.org), we faced a major problem with reproducibility of experimental results collected from multiple users. This was largely due to a lack of information about all software and hardware dependencies as well as a large variation of measured characteristics.
I will present a possible collaborative approach to solve above problems using a new Collective Mind knowledge management system. This modular infrastructure is intended to preserve and share through Internet the whole experimental setups with all related artifacts and their software and hardware dependencies besides just performance data. Researchers can take advantage of shared components and data with extensible meta-description at http://c-mind.org/repo to quickly prototype and validate research techniques particularly on software and hardware optimization and co-design. At the same time, behavior anomalies or model mispredictions can be exposed in a reproducible way to interdisciplinary community for further analysis and improvement. This approach supports our new open publication model in computer engineering where all results and artifacts are continuously shared and validated by the community (c-mind.org/events/trust2014).

Xin Zhao

Programming Runtime Support for Irregular Computations

Irregular computations have become increasingly important in many areas in recent year such as bioinformatics and social network analysis. Traditional data movement approaches for scientific computation are not well suited for such applications. The Active Messages (AM) model is an alternative communication paradigm that is better suited for such applications by allowing computation to be dynamically moved closer to data. Given the wide usage of MPI in scientific computing, enabling an MPI-interoperable AM paradigm would allow traditional applications to incrementally start utilizing AMs in portions of their applications, thus eliminating the programming effort of rewriting entire applications.

In our previous work we proposed a new generalized framework for MPI-interoperable Active Messages that can provide rich semantics to accommodate a wide variety of application computational patterns. Together with a new API, we present a detailed design of the correctness semantics of the functionality, including memory semantics, interoperability, ordering, concurrency, etc. We also proposed techniques for data streaming, buffering management and asynchronous processing to guarantee the correct execution of irregular applications as well as to achieve high performance. In this talk, I will discuss about irregular computations and the effort we made from programming model and runtime to make the computations easier and faster.

Xin Zhao is a fourth-year Ph.D. student from the Department of Computer Science at the University of Illinois at Urbana-Champaign (UIUC), advised by Prof. William Gropp. Her research interests focus on parallel programming models / runtime systems and irregular applications, with an emphasis on communication, resources management and dynamic execution.

Victor Lopez

DLB: Dynamic Load Balancing Library

Distribute equal amounts of work between tasks is not always trivial and usually becomes a negative performance impact in an application. DLB is a dynamic library designed to speed up hybrid applications by improving its load balance with little or none intervention from the user. The idea behind the library is to redistribute the computational resources of the second level of parallelism (OpenMP, OmpSs) to improve the load balance of the outer level of parallelism (MPI). DLB library uses an interposition technique at run time, so it is not necessary to do a previous analysis or modify the application; although finer control is also supported through an API.

We will present also a case study with CESM (Community Earth System Model), a global climate model that provides computer simulations of the Earth climate states. The application already uses a hybrid parallel programming model (MPI+OpenMp), so with few modifications in the source code we have compiled it to use the OmpSs programming model where DLB will benefit from the high malleability of it.

Marc Snir

Runtime and OS research at DoE

Pieter Bellens

Quantifying the effect of rectangular blocks in the dense QR factorization

Blocked, dense QR factorization using Householder reflectors attains minimal communication bounds and creates a fine-grained parallel computation. We consider the effects of rectangular block dimensions for an implementation in OmpSs, hereby unifying the traditional algorithm, that uses panels or block columns, and the square-blocked variants. Communication, computation, potential parallelism and hence the performance are functions of the block dimension. We use hardware counters and the Task Dependence Graph to quantize these properties for different matrix dimensions. Our measurements indicate that, against the grain of traditional practice, performance in dynamically scheduled environments can be improved by resorting to blocks with rectangular dimensions

Francois Tessier

Distributed communication-aware load balancing with TreeMatch in Charm++

Programming multicore or manycore architectures is a hard challenge particularly if one wants to fully take advantage of their computing power. Moreover, a hierarchical topology implies that communication performance is heterogeneous and this characteristic should also be exploited. We developed a parallel and distributed hierarchical load balancer for Charm++ that take into account both aspects. This work is based on our TreeMatch library that computes process placement in order to reduce an application communication cost based on the hardware topology. We show that the proposed load-balancing scheme manages to improve the execution times while being computed fast and in a scalable manner.

Brice Videau

Porting HPC applications to the Mont-Blanc prototype using BOAST

One of the goal of the Mont-Blanc project is to use real HPC application to evaluate the feasibility of exascale architectures using off the shelf hardware commonly used in the embedded world. The porting of those application is thus of paramount importance for the project. But, if getting scientific applications to run on the target platform is not very difficult, obtaining good performance portability is challenging. Indeed, HPC software are often hand tuned for the most frequently encountered architectures, and those optimizations can prove harmful if applied on a very different architecture. One way to alleviate this problem is to use task based runtimes to obtain adaptive application from a load balancing and network point of view. Unfortunately this solves only part of the problem. Individual tasks also have to be optimized and can be very sensitive to many parameters that are often not clearly exposed in the source code. We thus propose BOAST a meta-programming tool aiming at generating parametrized source code. Several output languages are supported and an expressive DSL is defined to help express the optimizations. An integrated compilation and execution framework is also supplied. This allows to directly test the generated kernels inside BOAST. This talk will present BOAST and how we used it to port part of two HPC applications:

- the Debauchies wavelet kernels of BigDFT, a quantum physics software that compute the electronic density around atoms and molecules,

- a port from CUDA to OpenCL of SPECFEM3D_GLOBE, a wave propagation software based on spectral finite element methods.

Performance results will also be presented.

Lokman Rahmani

Smart In Situ Visualization for Climate Simulations

The increasing gap between computational power and I/O performance in new supercomputers has started to drive a shift from an offline approach to data analysis to an inline approach, termed in situ visualization (ISV). While most visualization software now provides ISV, they typically visualize large dumps of unstructured data, by rendering everything at the highest possible resolution. This often negatively impacts the performance of simulations that support ISV, in particular when ISV is performed interactively, as in situ visualization requires synchronization with the simulation. In this work, we advocate for a smarter method of performing ISV. Our approach is data-driven: it aims to detect potentially interesting regions in the generated dataset in order to feed ISV frameworks with “the interesting” subset of the data produced by the simulation. While this method mitigates the load on ISV frameworks by making them more efficient and more interactive, it also helps scientists focus on the relevant part of their data. We investigate smart ISV in the context of a climate simulation, with a set of generic filters derived from information theory, statistics and image processing, and show the tradeoff between performance and quality of visualization.

Justin Wozniak

Case Studies in Big Data and HPC from X-ray Crystallography

Recent advancements in X-ray crystallography methods, including experimental techniques and detector technology, have produced a data explosion (10's of TBs/week) that has outpaced increases in conventional computational and storage capacity, leading to a crisis in computational analysis and data management in X-ray sciences. Typical Big Data solutions do not accommodate the ad hoc nature of the scientific workflow, including opportunistic use of hardware and highly specialized analysis tools. From a computational perspective, existing analysis codes must be quickly scaled up to massively parallel resources. In this presentation, we will describe our recent work applying the Swift programming language to four applications in X-ray sciences, addressing problems in wide-area data movement and management as well as scaling existing applications on large clusters and the Blue Gene/Q.

Christine Morin

Contrail: Interoperability and Dependability in a Cloud Federation

Cloud computing market is in rapid expansion due to the opportunities to dynamically allocate large amount of resources when needed and to pay only for their effective usage. However, many challenges, in terms of interoperability, performance guarantee, and dependability, should still be addressed to make cloud computing the right solution for companies. Contrail integrated project (IP), funded by the European Commission (http://www.contrail-project.eu) developed a comprehensive cloud computing software stack in open source to address these challenges.
In this talk we first discuss the main challenges faced in the open cloud market and then we present components developed in the framework of the Contrail European project to provide solutions to guarantee interoperability in a cloud federation and to deploy distributed applications over a federation of heterogeneous cloud providers. Our solutions allow to negotiate QoS and QoP SLA terms for an application and then map them on the physical resources.

Martin Quinson

Formal verification of unmodified MPI applications with SimGrid

This talk will first recap the approach leveraged in SimGrid to formally assess the correction of MPI applications through model checking. It will be focused on our current status report and future work. We are now able to verify safety properties, but also liveness properties (with some restrictions), on unmodified small to medium MPI applications (few thousands of lines in C, C++ or Fortran). I will conclude with the research leads that we are currently working on, and with the kind of collaboration that could occur within the Joint Lab with the potential users of such tools.

Yves Robert

Algorithms for coping with silent errors
Silent errors have become a major problem for large-scale distributed systems. Detection is hard, and correction is even harder. This talks presents generic algorithms to achieve both detection and correction of silent errors, by coupling verification mechanisms and checkpointing protocols.

Slim Bouguerra

Energy-Performance Tradeoffs in Multilevel Checkpoint Strategies

Increased complexity of computer architectures, consideration of power constraints, and expected failure rates of hardware components make the design and analysis of energy-efficient fault-tolerance schemes an increasingly challenging and important task. We develop run-time and energy models for multilevel checkpoint schemes and characterize when tradeoffs between expected runtime and energy usage exist. Using these models, we study FTI, a recently developed multilevel checkpoint library, on an IBM Blue Gene/Q. We show that FTI has a low energy footprint and that, consequently optimal checkpoint-interval values with respect to time and energy are similar. We also explore the effect of general system-level parameters on run-time and energy tradeoffs.

Tatiana V. Martsinkevich

Using dedicated resources to alleviate memory limitation for message logging protocols

There are different approaches on how to handle memory limitation for a message logging protocol. The simplest is to take a checkpoint once one of the processes runs out of memory or dump logs to the stable storage to free the memory. However this may increase the load on the the I/O subsystem which is not desirable especially for large-scale runs . Another approach is to use the memory of additional dedicated nodes as a log storage: when a process runs out of memory it sends a portion of its log to the memory of a dedicated node. I will present the study on the feasibility of this approach and explore the overheads related to it.

Ana Gainaru

The road to failure prediction on Blue Waters: latest details and future directions

We analyze the characteristics of failures from the Blue Waters system and study their effect on the results given by the online failure prediction. We make a couple of key observations about the difference in behaviour between different failure types and propose specific optimizations for each. A detailed analysis of the prediction results is also given. We present future work direction together with preliminary results.

Leonardo Bautista Gomez

Fault Tolerance Interface new features and new developments

Slim Bouguerra

Energy-Performance Tradeoffs in Multilevel Checkpoint Strategies

Increased complexity of computer architectures, consideration of power constraints, and expected failure rates of hardware components make the design and analysis of energy-efficient fault-tolerance schemes an increasingly challenging and important task. We develop run-time and energy models for multilevel checkpoint schemes and characterize when tradeoffs between expected runtime and energy usage exist. Using these models, we study FTI, a recently developed multilevel checkpoint library, on an IBM Blue Gene/Q. We show that FTI has a low energy footprint and that, consequently optimal checkpoint-interval values with respect to time and energy are similar. We also explore the effect of general system-level parameters on run-time and energy tradeoffs.