Joint-lab workshop Nov. 21-23 2011

This event is supported by INRIA, UIUC and NCSA, the French ministry of foreign affairs, as well as by EDF

Main Topics	Schedule	Speaker	Affiliation	Type of presentation	Title (tentative)	Download
	Sunday Nov. 20th	Dinner at Larissa, Victoria and Bill Kramers' house			**Buses will depart from the Hampton Inn hotel at 5:45PM** Address: 1103 Country Lane in Champaign.	
Workshop Day 1	Monday Nov. 21th					
					ALL TITLES ARE TEMPORARY	
Registration	08:00					
Welcome and Introduction	08:30	Marc Snir + Franck Cappello	INRIA &UIUC	Background	Welcome, Workshop objectives and organization	
	08:40	Melanie Loots Associate Vice Chancellor for Research at UIUC	UIUC	Background	Greetings from UIUC	
		Robert Jeansoulin Attaché for science & technology (IT), Embassy of France, Washington	French Ambassy	Background	Greetings from French Ambassy	
		Claude Kirchner Executive Officer for Research and Technology Transfer for Innovation at INRIA	INRIA	Background	Greetings from INRIA	
		Danny Powell *NCSA* Executive Director	NCSA	Background	Welcome at NCSA	
Sustained Petascale Chair: Marc Snir	09:00	Bill Kramer	NCSA	Background	**Blue Waters Redone:** Un super système pour résoudre de super défis	^BlueWatersOverview_Nov11_INRIA_Workshop_a_submitted.pdf
	09:30	Bill Gropp	UIUC	Background	**Application Challenges for Sustained Petascale**	^cs-challenges-Gropp.pptx
	10:00	Break				
From Petascale to Exascale Chair: Franck Cappello	10:30	Marc Snir	ANL & UIUC	Background	**About Argonne MCS, About Exascale**	^snir JLPC 11-11.zip
	11:00	Wen-Mei Hwu	UIUC	Background	**GPUs in Blue Waters and Beyond**	
	11:30	Rajeev Thakur	ANL	Background	**Challenges in Scaling MPI to Exascale**	^Rajeev(1).pdf
	12:00	Lunch				
	13:30	Robert Ross	ANL	Background	**Storage Architectures and Abstractions for Exascale Systems**	^ross_uiuc-inria-11-2011.pdf
	14:00	Paul Hovland	ANL	Background	**An Overview of Applied Math Activities at Argonne.**	^Hovland_UIUC_INRIA_Nov2011.pdf
	14:30	George Bosilca	UTK /ICL	Background	**Enabling Software Fault Tolerance in MPI**	^bosilca.pdf
	15:00	Break				
System software Chair: Yves Robert	15:30	Franck Cappello	INRIA &UIUC	Joint Results	Introduction of the activities in System	^Syst-software-activity.pptx
	15:45	Ana Gainaru	UIUC & NCSA	Joint Results	**Signal Analysis for Modeling the Normal and Faulty Behavior of Large-scale HPC Systems**	^signals.pdf

	16:15	Thomas Ropars	EPFL	Joint Results	**On Distributed Recovery for Send-Deterministic-Aware MPI Applications**	^send_determinism.pdf
	16:45	Leonardo Bautista Gomez	Titech	Joint Results	**Fast checkpoint restart for sustained petascale computing: Opportunities and directions.**	^jlpc-ws6.pdf
	17:15	Olivier Gluck	INRIA	Joint Results	**Reducing energy consumption of fault tolerance algorithms**	^OlivierGluck.pdf
	18:30	Dinner at SilverCreek			Buses will depart from the Hampton Inn at 6:30 PM 402 North Race Street Urbana, IL 61801, (217) 328-3402	
Workshop Day 2	**Tuesday Nov. 22th**					
System Software cont. Chair: Torsten Hoefler	08:30	Michele Buttler and Bill Kramer	NCSA	Background	Storage system issues for sustained petascale systems	^INRINA Nov 2011.ppt
	09:00	Gabriel Antoniu & Matthieu Dorrier	INRIA	Joint Results	**Update on Damaris: How CM1 Scales Linearly up to (Almost) 10K Cores And What Comes Next**	^2011-11-22-JLPCworkshop-Antoniu-Dorier.pdf
Numerical Library Chair: Jean Roman	09:30	Bill Gropp	UIUC	Joint Results	Introduction of the activity in Numerical Algorithms and Libraries	^NA-Summary-Gropp.pptx
	09:40	Luc Giraud	INRIA	Joint Results	**Towards robust numerical linear solvers for large scale simulation**	Robust-numerical-linear-solvers.pdf\
	10:15	Break				
	10:45	Laura Grigori	INRIA	Joint Early Results	**Hybrid scheduling and communication avoiding for CALU**	^HybridSchedCANov11.pdf
	11:20	Sébastien Fourestier, Harshitha Menon	INRIA	Joint Early Results	**Latest improvements to Scotch and ongoing collaborations**	^fourestier_menon_Latest_improvements_to_SCOTCH_and ongoing_collaborations.pdf
	11:55	Yves Robert	INRIA	Background	**Linear algebra kernels on petascale/exascale platforms: scheduling issues**	^YvesRobert.pdf.gz
	12:30	Lunch				
Numerical Lib. Cont. Chair: Bill Gropp	14:00	Marc Baboulin	INRIA	Joint Early Results	**A parallel tiled solver for dense symmetric indefinite systems on multicore architectures**	^baboulin.pdf
	14:30	Daisuke Takahashi & Alex Yee	U. Tsukuba	Joint Results	**A Scalable Parallel Algorithm for 3-D FFT**	^takahashi.pdf
Programming environments Chair: Rajeev Thakur	15:00	Sanjay Kale	UIUC	Joint Early Results	**Some progress highlights for Charm++**	^inriaWorkshopSummaryAndTalk.pptx
	15:30	Julien Bigot / Christian Perez	INRIA	Joint Early Results	**Modularizing an FFT library with Charm++ & HLCM: combining performance and portability**	^jbigot-Charm_HLCM.odp
	16:00	Break				
	16:30	Alexandre Duchateau	UIUC	Joint Early Results	**Generation and Tuning of parallel solutions for linear algebra equations**	^Duchateau_Jointlab_November_2011.pdf
	17:00	Laercio Pilla, Jean François Mehaud	INRIA	Joint Early Results	**Topology-Aware Load Balancing for Parallel Applications on Multi-Core Systems and Beyond**	^topology-aware_lb_pilla.pdf
	17:30	Emmanuel Jeannot	INRIA	Joint Early Results	**Process placement on multicore. Load balancing in Charm++ and comparison of TreeMatch with graph partitioners**	^Opening-6th-Workhsop.ppt
	18:00	Franck Cappello & Marc snir	INRIA &UIUC & ANL		Preparation of the working groups	
	19:00	Dinner at NCSA Lobby			Viz demo and Petascale Facility tour. Appetizer 6PM, Diner 6:30-7:30	

Workshop Day 3	**Wednesday June 23th**					
	8:50	Franck Cappello & Marc snir	Auditorium		Indications for working groups Q&A about collaboration implementation	
Working groups	9:00-10:30	Bill Gropp	NCSA 1030		Numerical libraries (Laura Grigori, Yves Robert, Sebastien Lefourestier + Paul Hovland + Wen-Mei Hwu, Marc Baboulin, Alexandre Duchateau, Daisuke Takahashi, Alex Yee + Torsten Hoefler, etc....)	
	9:00 - 10:30	Marc Snir	NCSA 1040		I/O (Bill Kramer + Gabriel Antoniu + Matthieu Dorrier + Michele Buttler + Brett Bode + Rajeev Thakur + Rob Ross + Pavan Balaji + Franck Cappello, Olivier Gluck...)	
	10:30	Break				
	11:00 - 12:30	Sanjay Kale	NCSA 1030		Programming models (Jean Francois Mehaut, Sebastien Fourestier, Chrsitian Perez, Emmanuel Jeannot, Pavan Balaji + Wen-Mei Hwu, Torsten Hoefler, ...)	
	11:00 - 12:30	Franck Cappello	NCSA 1040		Resilience: resilient algorithms (Bill Gropp, Yves Robert, Laura Grigori+ Marc Baboulin, ...) and resilient systems (Bill Kramer, Marc Snir, Ana Gainaru, Leonardo Bautista, Yves Robert + Rajeev Thakur + Thomas Ropars, Esteban Meneses, Olivier Gluck...)	
	12:30	Adjourn				
	13:00	Lunch				
	14:30 - 18:00				Informal working groups	
	19:00	Dinner at Ribeye			Buses will depart from the Hampton Inn at 6:45 PM 1701 S. Neil St,	

Abstracts

Rajeev Thakur: **Challenges in Scaling MPI to Exascale**

This talk will discuss challenges in using MPI effectively at exascale. I will describe ongoing research at Argonne aimed at addressing these challenges. I will also give an update on recent activities of the MPI Forum and what new features are being considered for inclusion in MPI-3.

Robert Ross: **Storage Architectures and Abstractions for Exascale Systems**

The complexity and scale of upcoming systems, applications, and analysis workflows motivate dramatic change in future HPC storage systems. This talk will outline three areas where R&D activities now can better prepare us for the development and deployment of these future systems: understanding I/O behavior, tools for exploring the HPC storage design space, and abstractions and data model support in the storage system. For each area, related ongoing activities will be discussed.

Paul Hovland: **An Overview of Applied Math Activities at Argonne.**

We provide a survey of applied mathematics research at Argonne National Laboratory. We present the philosophy behind software development and the primary capabilities of our software libraries. We argue for the use of the highest possible abstraction in scientific computation and point toward the use of high level scientific computing libraries.

George Bosilca: **Enabling Software Fault Tolerance in MPI**

The International Exascale Software Project roadmaps predicts, as soon as 2014, billion way parallel machines encompassing not only millions of cores, but also tens of thousands of nodes. Even considering extremely optimistic advances in hardware reliability, probabilistic amplification entails that failures will be unavoidable. Consequently, software fault tolerance is of paramount importance to maintain future scientific productivity. Major problems hinder ubiquitous adoption of fault tolerance techniques: 1) traditional checkpoint based approaches incur a steep overhead on failure free operations and 2) the dominant programming paradigm for parallel applications (the MPI standard and its implementations) offers extremely limited support of software-level fault tolerance approaches. In this talk, I will present and evaluate an approach that relies exclusively on the current MPI standard definition of a high quality implementation and enables algorithmic based recovery to complete the computation despite failures without incurring the overhead of customary periodic checkpointing.

Ana Gainaru: **Signal Analysis for Modeling the Normal and Faulty Behavior of Large-scale HPC Systems**

This talk will present a novel way of characterizing the normal and faulty behavior of the system by using signal analysis concepts. All analysis modules create ELSA (Event Log Signal Analyzer), a toolkit that has the purpose of modeling the normal flow of each state event during a HPC system lifetime, and how it is affected when a failure hits the system. Current event mining approaches do not take into consideration the specific behavior of each type of events and as a consequence, fail to analyze them according to their characteristics. We will show that our models provide an accurate view of the system output, which improves the effectiveness of proactive fault tolerance algorithms. Specifically, we implemented a filtering algorithm and short-term fault prediction methodology based on the extracted model and test it against real failure traces from a large-scale system. We show that by analyzing each event according to its specific behavior, we get a more realistic overview of the entire system.

Thomas Ropars: **On Distributed Recovery for Send-Deterministic-Aware MPI Applications**

The send-deterministic execution model states that in any correct execution of an application, the processes send the same sequence of messages for a given set of input parameters. Many large scale MPI HPC applications comply with this model. Send-determinism allows to design new rollback-recovery protocols that: i) can rely on uncoordinated checkpointing without suffering from the domino effect; ii) can provide failure containment with a limited performance overhead. One major challenge remains: how to make recovery efficient and scalable ?
In this talk, we first give a brief overview of the principles and the performances of HydEE, our hybrid rollback-recovery protocol based on send-determinism. Then we discuss the problems related to performance on recovery, and we show how recovery could be made fully distributed in such a protocol if the application was able to express its send-determinism.

Olivier Gluck: **Reducing energy consumption of fault tolerance algorithms**

Over the past few years, energy consumption of supercomputers has become a major issue. In order to be able to meet the important needs in terms of performance that express scientists in various fields, supercomputers are growing too fast. In fact, they involve more and more computing nodes, which consequently increase both their total energy consumption and their probability to experience a failure. Especially, in order to ensure the transition to the exascale era by 2018 which will involve millions of cores, we need to address these two challenges by providing efficient fault tolerance mechanisms while reducing the total energy consumption.
In this talk, we first present some techniques used to reduce the energy consumptions of large scale distributed systems and particularly in future supercomputers. Then, we present our current research works for reducing energy consumption costs of fault tolerance algorithms in exascale supercomputers.

Yves Robert: **Linear algebra kernels on petascale/exascale platforms: scheduling issues**

Future exascale machines will likely be massively parallel architectures, with 100K to 1000K processors, each processor itself being equipped with 1K to 10Kcores. At the node level, the architecture is a shared-memory machine, running many parallel threads on the cores. At the machine level, the architecture is a distributed-memory machine. This additional level of hierarchy, together with massive parallelism at the node level, dramatically complicates the design of new versions of the standard numerical linear algebra algorithms that are at the heart of many scientific applications. On exascale platforms, resilience is a key challenge. Failures are much more likely to occur during the execution of parallel jobs that enroll increasingly larger numbers of processors. The design of efficient fault-tolerant scheduling strategies will be key to high performance. Such strategies can involve either checkpointing, or task replication, or dynamic task re-execution, or any combination. But they all incur big overheads in terms of performance, and of energy-consumption. The main goal of the talk is to survey the challenges faced to design linear algebra algorithm on exascale architectures, and to provide a few examples of algorithms and scheduling techniques
that constitute a first step to solving these challenges. Joint work with Marin Bougeret, Henri Casanova, Jack Dongarra, Thoma Hérault, Julien Langou, Mathieu Faverge, and Frédéric Vivien.

Gabriel Antoniu and Matthieu Dorier: **Update on Damaris: How CM1 Scales Linearly up to (Almost) 10K Cores And What Comes Next**

With exascale computing on the horizon, the performance variability of I/O systems represents a key challenge in sustaining high performance, as it significantly impacts the overall application performance. In previous work we introduced Damaris, an I/O library which leverages dedicated I/O cores on each multicore SMP node to efficiently perform asynchronous data processing and I/O. We present new results for Damaris through large-scale experiments (up to over 9K cores) performed on the Kraken Cray XT5 supercomputer with the CM1 atmospheric model, one of the target HPC applications for the Blue Waters project. We increase the sustained write throughput by a factor of almost 15 and we provide almost 70 % overall application speedup while fully hiding the I/O costs. We then present possible future directions for achieving efficient in-situ visualization without disturbing the simulation and we discuss the possible benefits of the BlobSeer approach to concurrency-optimized metadata management in this context.

Luc Giraud: **Towards robust numerical linear solvers for large scale simulation**

In this talk we will review numerical schemes that are natrually resilient to core faults. We will show how other might be adapted to enjoy similar properties.

Laura Grigori: **Hybrid scheduling and communication avoiding for CALU**

We present the use of a hybrid static/dynamic scheduling strategy of the task dependency graph for direct methods used in dense numerical linear algebra. This strategy provides a balance of data locality, load balance, and low dequeue overhead. We show that the usage of this scheduling in communication avoiding dense factorization leads to significant performance gains. On a 48 core AMD Opteron NUMA machine, our experiments show that we can achieve up to 64% improvement over a version of CALU that uses fully dynamic scheduling and 30% improvement over the version of CALU that uses fully static scheduling. On a 16-core Intel Xeon machine, our hybrid static/dynamic scheduling approach is up to 8% faster than the version of CALU that uses a fully static scheduling or fully dynamic scheduling. Our algorithm leads to important speedups over the corresponding routines for computing LU factorization in well known libraries. On the 48 core AMD NUMA machine, our best implementation is up to 110% faster than MKL, while on the 16 core Intel Xeon machine, it is 82% faster than MKL. Our approach also shows significant speedups compared with PLASMA on both of these systems.

Sebastien, Fourestier: **Last improvements in Scotch and ongoing collaborations.**

Scotch is a software package for sequential and parallel graph partitioning, static mapping, sparse matrix block ordering, and sequential mesh and hypergraph ordering. As a research project, it is subject to continuous improvement, resulting from several on-going research tasks. Our talk will focus on the last improvements we have done in Scotch and the ongoing collaborations within the joint laboratory. We will also briefly present other ongoing work, in the context of our new roadmap.

Marc Baboulin: **A parallel tiled solver for dense symmetric indefinite systems on multicore architectures**

We present an efficient and innovative parallel tiled algorithm for solving symmetric indefinite systems on multicore architectures. This solver avoids the communication overhead due to pivoting by using symmetric randomization. This randomization is computationally inexpensive and requires very little storage. Following randomization, a tiled LDLT factorization is used that reduces synchronization by using static or dynamic scheduling. We compare Gflop /s performance of our solver with other types of factorizations on a current multicore machine and we provide tests on accuracy using LAPACK test cases.

Daisuke Tekahashi and Alex Yee: **A Portable Approach to the Super-Optimized Hand-Written FFTs AND A Scalable Parallel Algorithm for 3-D FFT**

FFTs typically fall into two categories: Generic Libraries, and Specialized Generic libraries are the libraries that we normally use. They are generic, and portable. Specialized FFTs, are generally hand-written and specialized for their tasks. They are generally not-portable, but are much faster than generic libraries - often by factors of 2, 3, or more. We propose a new FFT implementation that is the "middle ground" between generic and specialized FFTs. Our implementation is faster than FFTW and nearly as comparable to some specialized FFT implementations - while being reasonably portable.

In this talk, a scalable parallel algorithm for 3-D fast Fourier transform (FFT) is presented. A typical decomposition for performing a parallel 3-D FFT is slab-wise. In this case, for N^3-point FFT, N must be greater than or equal to the number of MPI processes. Our proposed parallel 3-D FFT algorithm allows up to N^(3/2) MPI processes for N^3-point FFT. Moreover, this scheme requires only one all-to-all communication for transposed-order output. Performance results of parallel 3-D FFTs on clusters of multi-core processors are reported.

Julien Bigot: **Modularizing an FFT library with Charm++ & HLCM: combining performance and portability**

When designing a High Performance application, one usually has to handle two kinds of decomposition. The first one is dictated by the parallelism of the hardware platform. The second one follows the logical module that form the application. In order to combine high performance with a high level of code re-usability, the code should reflect both. Programming models such as Charm++ offer a good support for parallelism. Charm++ encourages a philosophy of over-decomposition. Applications are decomposed into chares, objects that communicate by exchanging messages. They are executed in parallel on the available processors. Object-oriented languages do however lack intrinsic support for modular decomposition. The paradigm of component based software engineering has been proposed to tackle this problem. Components are pieces of code that can be externally assembled to form the whole application. When combining these two kinds of decomposition, care should be taken as they can interfere. For example, replacing a given component with an implementation relying on a different parallel decomposition can lead to inefficient data redistribution at the interface between components. The HLCM component assembly model has been designed to support the efficient combination of both form of decomposition. It supports user defined interactions that can be optimized for various kind of hardware platforms and is based on a compilation approach to prevent any overhead at runtime. We present an implementation that enables the use of HLCM to assemble Charm++ components. We show how this has been used to modularize an FFT library with minimal modification to the code. We evaluate this by showing that the modularized code behaves similarly to the initial one with respect to performance while easing the replacement of some of its module with code optimized for specific hardware.

Alexandre Duchateau: **Generation and Tuning of parallel solutions for linear algebra equations**

An auto-tuning system and methodology for algorithm exploration for a class of linear algebra problems. Starting with a description of equations, the system automatically finds divide and conquer algorithms to solve the equations with the main objective of exposing parallelism. The same strategy can be used to improve cache locality.

Jean-François Mehaut: **Topology-Aware Load Balancing for Parallel Applications on Multi-Core Systems and Beyond**

The current trend in building high performance parallel machines is to use chips based on multi-core design. These chips feature a complex and hierarchical core topology, cache and memory subsystem. Although this design provides high processing power to parallel machines, it comes with the cost of increased memory access latencies. In order to fully exploit the potential of these machines, it becomes crucial to take into account memory affinities, which are provided by having a complete view of the machine topology.
In this presentation, we talk about our topology-aware approach for load balancing. It combines information about the machine topology and the characteristics of the application. The load balancing algorithm improves application performance by equalizing the load on the available cores while improving the affinity between cores and memory. It is also proved asymptotically optimal. We discuss some of the results obtained with a version of the algorithm implemented for Charm++. We will also talk about an extension of this work to address clusters of multi-core machines.

Emmanuel Jeannot: **Process placement on multicore. Load balancing in Charm++ and comparison of TreeMatch with graph partitioners**

We continue our study of the process placement strategy and its implementation as a load-balancer in Charm++. We also provide a comparison in terms of speed and efficiency of our algorithm (TreeMatch) against standard graph partitionners (e.g. Scotch, Metis, Chaco, etc.)