joint-lab workshop Nov. 25-27 2013

The agenda below is close to the final one

This event is supported by INRIA, UIUC, NCSA, ANL and French Ministry of Foreign Affairs

Main Topics	Schedule	Speaker	Affiliation	Type of presentation	Title (tentative)	Download
Sunday Nov. 24th Dinner Before the Workshop	7:00 PM **(Departure from Hampton Inn at 6:45PM) with mini buses**	Only people registered for the dinner			**Restaurant:** **Silvercreek:** **Address: 402 N Race St, Urbana, IL 61801** **Phone:(217) 328-3402**	
Workshop Day 1	**Monday Nov. 25th**					
					TITLES ARE TEMPORARY (except if in bold font)	
Registration	08:00					
Welcome and Introduction **Auditorium 1122** **Chair: Franck Cappello**	08:30	Marc Snir + Franck Cappello **Co-directors of the joint-lab**		Background	**Welcome, Workshop objectives and organization**	Opening-10th-Workshop.pdf
	08:45	Ed. Seidel **Incoming NCSA director**	UIUC	Background	**NCSA update and vision of the collaboration** (This address has been inverted with the next one due to schedule constraints)	
	09:00	Peter Schiffer **UIUC Vice Chancellor for Research**	UIUC	Background	**Welcome from UIUC Vice Chancellor for Research**	
	09:15	Michel Cosnard **Inria CEO and President**	Inria	Background	**INRIA updates and vision of the collaboration**	HPC@Inria-UIUC-nov13-v2.pptx
	09:30	Marc Snir **Director of Argonne/ MCS and co-director of the joint-lab**	ANL	Background	**Argonne updates and vision of the collaboration**	jlpc 11-13 snir.pdf
	09:45	Marc Daumas **Attaché for Science and Technology**	Embassy of France	Background	**France-USA collaboration program updates**	http://prezi.com/hsggz_30xlqt/2013-jlpc-workshop-ncsa-uiuc-il/
	9h55	Franck Cappello **Co-director of the Joint-lab**	ANL	Background	**Joint-Lab, PUF, New Joint-Lab, organization**	Joint-Lab-JLESC-PUF.pdf
	10:15	Break				
Extreme Scale Systems and infrastructures **Auditorium 1122** **Chair: Pavan Balaji**	10:45	Pete Beckman	ANL		Extreme Scale Computing & Co-design Challenges	
	11:15	John Towns	UIUC		Applications Challenges in the XSEDE Environment	XSEDE-Apps-Challenges-for-Joint-Lab.pdf

	11:45	Gabriel Antoniu	Inria		A-Brain and Z-CloudFlow: Scalable Data Processing on Azure Clouds - Lessons Learned in Three Years and Future Directions	2013-11-25-JLPC-Azure-final.pdf
	12:15	Lunch				
Chair: Yves Robert	13:45	Bill Kramer	UIUC	Blue Waters	Is Petascale Completely Done? What Should We Do Now?	Kramer JLPC November Workshop - v1.pdf
	14:15	Torsten Hoefler	ETH	IEEE/ACM SC13 Best Paper	Enabling Highly-Scalable Remote Memory Access Programming with MPI-3 One Sided	hoefler-mpi3rma-slides.pdf
	14:45	Rob Ross	ANL		Thinking Past POSIX: Persistent Storage in Extreme Scale Systems	ross_uiuc-storage-20131125.pdf
	15:15	Break				
Chair: Bill Gropp	15:45	François Pellegrini	Inria		Parallel repartitioning and remeshing : results and prospects	pellegrini_scotch.pdf pellegrini_pampa.pdf
	16:15	Pavan Balaji	ANL		Message Passing in Massively Multithreaded Environments	2013-11-25-jlpc-threads-pavanbalaji.pptx
	16:45	Wen Mei Hwu	UIUC		A New, Portable Algorithm Framework for Parallel Linear Recurrence Problems	UIUC_INRIA__Tangram_GPU_2013_Hwu.pdf
	17:15	Adjourn				
Diner	**(Departure from Hampton Inn at 6:45PM) with mini buses)**				**Restaurant: Kamakura:** **Address: 715 S Neil St, Champaign, IL 61820** **Phone:(217) 351-9898**	
Workshop Day 2	**Tuesday Nov. 26**					
Applications, I/O, Visualization, Big data **Auditorium 1122** **Chair: Rob Ross**	08:30	Greg Bauer	UIUC		Applications and their challenges on Blue Waters	GBAUER-INRIA-NCSA-BW-2013.pdf
	09:00	Matthieu Dorier	Inria	Joint-result, submitted	CALCioM: Mitigating I/O Interferences in HPC Systems through Cross-Application Coordination	DORIER-JLPC-November2013.pdf
	09:30	Dries Kimpe	ANL		Mercury: Enabling Remote Procedure Call for High-Performance Computing	dkimpe-mercury.pdf
	10:00	Break				
Chair: Gabriel Antoniu	10:30	Venkat Vishwanath	ANL		Addressing I/O Bottlenecks and Simulation-Time Analytics at Extreme Scales	VISHWANATH_INRIA_JLPC_DIST.pdf
	11:00	Babak Behzad	UIUC	ACM/IEEE SC13	Taming Parallel I/O Complexity with Auto-Tuning	Babak_Slides.pdf
	11:30	McHenry, Kenton Guadron	UIUC		NSF CIF21 DIBBs: Brown Dog	
	12:00	Lunch				
Mini Workshop1 **Resilience** **Room 1030** **Chair: Frederic Vivien**						
	13:30	Wesley Bland	ANL		Fault Tolerant Runtime Research at ANL	bland-jlpc.pdf
	14:00	Tatiana Martsinkevich	Inria	Joint-result	On the feasibility of message logging in hybrid hierarchical FT protocols	martsinkevich jlpc workshop in ncsa.pdf

	14:30	Mohamed Slim Bouguera	Inria	Joint-result, submitted	Failure prediction: what to do with unpredicted failures ?	jointlab_ipdps_presentation_v0.pdf
	15:00	Ana Gainaru	UIUC	Joint-result, submitted	Topology and behaviour aware failure prediction for Blue Waters.	jlpc13_againaru.pdf
	15:30	Break				
Chair: Franck	16:00	Sheng Di	Inria	Joint-result, submitted	Optimization of Multi-level Checkpoint Model for Large Scale HPC Applications	10th-Joint-workshop-UIUC-sdi.ppt
	16:30	Yves Robert	Inria	Joint-result,	Assessing the impact of ABFT & Checkpoint composite strategies	joint-lab2013.pdf
	17h00	Leonardo Bautista Gomez	ANL	Joint-result ACM PPoPP 2014	Detecting Silent Data Corruption through Data Dynamic Monitoring for Scientific Applications	jlpc10leo.pdf
	17H30	Adjourn				
Diner	**(Departure from Hampton Inn at 7PM) with mini buses)**				**Restaurant:** **Ko-Fusion:** **Address: 1 Main St #104, Champaign, IL 61820** **Phone:(217) 531-1166**	
Mini Workshop2 **Numerical Agorithms** **Room 1040** **Chair: Stefan Wild**						
	13:30	Luke Olson	UIUC		Toward a more robust sparse solver with some ideas on resilience and scalability	2013_JointLab_NCSA_Olson.pdf
	14:00	Prasanna Balaprakash	ANL		Active-Learning-based Surrogate Models for Empirical Performance Tuning	Balaprakash.pdf
	14:30	Yushan Wang	Inria		Solving 3D incompressible Navier-Stokes equations on hybrid CPU/GPU systems.	JointLab-Urbana.pdf
	15:00	Jed Brown	ANL		Fast solvers for implicit Runge-Kutta systems	20131126-JointLabRungeKutta.pdf
	15:30	Break				
Chair: Luke Olson	16:00	Pierre Jolivet	Inria	Best Paper finalist, IEEE, ACM SC13	Scalable Domain Decomposition Preconditioners For Heterogeneous Elliptic Problems	jolivet-ddm.pdf
	16:30	Vincent Baudoui	Total &ANL	Joint-result	Round-off error propagation and non-determinism in parallel applications	baudoui-roundoff_errors.pdf
	17:00	*Torsten Hoefler*	*ETH*		*Using Automated Performance Modeling to Find Scalability Bugs in Complex Codes*	htor.pdf
	17:30	Adjourn				
Diner	**(Departure from Hampton Inn at 7PM) with mini buses)**				**Restaurant:** **Ko-Fusion:** **Address: 1 Main St #104, Champaign, IL 61820** **Phone:(217) 531-1166**	
Workshop Day 3	**Wednesday Nov. 27**					
Mini Workshop3						

Programming models, compilation and runtime. **Room 1030** **Chair: Marc Snir**	08:30	Grigori Fursin	Inria		Collective Mind: making auto-tuning practical using crowdsourcing and predictive modeling	Fursin_Slides.pdf
	09:00	Maria Garzaran	UIUC		Optimization by Run-time Specialization for Sparse Matrix-Vector Multiplication	garzaranNCSA-INRIA.pdf
	09:30	Jean-François Mehaut	Inria		From Multicores to Manycores Processors: Challenging Programming Issues with the MPPA /KALRAY	slides_JFM.pdf
	10:00	Break				
	10:30	Rafael Tesser	Inria	Joint result PDP 2013	Using AMPI to improve the performance of the Ondes3D seismic wave simulator through dynamic load balancing	RafaelTessser-WSJLPC-Nov2013.pdf
	11:00	Emmanuel Jeannot	Inria	Joint-result, IEEE Cluster2013	Communication and Topology-aware Load Balancing in Charm++ with TreeMatch	cluster_slide.pdf
Auditorium 1122	11:30	**Closing**				
	12:00	Lunch				
Diner	**(Departure from Hampton Inn at 5:45 PM) with mini buses)**				**Restaurant:** **Ribeye:** **Address: 1701 S Neil St, Champaign, IL 61820** **Phone:(217) 351-9115**	
Mini Workshop4 **Large scale systems and their simulators** **Room 1040** **Chair: Bill Kramer**						
	08:30	Eric Bohm	UIUC		A Multi-resolution Emulation + Simulation Methodology for Exascale	JLPC_Bigsim-201311.pdf
	09:00	Arnault Legrand	Inria		SMPI: Toward Better Simulation of MPI Applications	smpi_jlpc_13.pdf
	09:30	Frederic Vivien	Inria		Scheduling tree-shaped task graphs to minimize memory and makespan	
	10:00	Break				
	10:30	Kate Keahey	ANL		Evaluating Streaming Strategies for Event Processing across Infrastructure Clouds	jointlab-ncsa.pdf
	11:00	Jeremy Enos	UIUC		Application Runtime Consistency and Performance Challenges on a shared 3D torus.	smpi_jlpc_13.pdf
Auditorium 1122	11:30	**Closing**				
	12:00	Lunch				
Diner	**(Departure from Hampton Inn at 5:45 PM) with mini buses)**				**Restaurant:** **Ribeye:** **Address: 1701 S Neil St, Champaign, IL 61820** **Phone:(217) 351-9115**	

Abstracts

Kenton McHenry

NSF CIF21 DIBBs: Brown Dog

The objective of this project is to construct a service that will allow for past and present un-curated data to be utilized by science while simultaneously demonstrating the novel science that can be conducted from such data. The proposed effort will focus on the large distributed and heterogeneous bodies of past and present un-curated data, what is often referred to in the scientific community as long-tail data, data that would have great value to science if its contents were readily accessible. The proposed framework will be made up of two re-purposable cyberinfrastructure building blocks referred to as a Data Access Proxy (DAP) and Data Tilling Service (DTS). These building blocks will be developed and tested in the context of three use cases that will advance science in geoscience, biology, engineering, and social science. The DAP will aim to enable a new era of applications that are agnostic to file formats through the use of a tool called a Software Server which itself will serve as a workflow tool to access functionality within 3rd party applications. By chaining together open/save operations within arbitrary software the DAP will provide a consistent means of gaining access to content stored across the large numbers of file formats that plague long tail data. The DTS will utilize the DAP to access data contents and will serve to index unstructured data sources (i.e. instrument data or data without text metadata). Building off of the Versus content based comparison framework and the Medici extraction services for auto-curation the DTS will assign content specific identifiers to untagged data allowing one to search collections of such data. The intellectual merit of this work lies in the proposed solution which does not attempt to construct a single piece of software that magically understands all data, but instead aims at utilizing every possible source of automatable help already in existence in a robust and provenance preserving manner to create a service that can deal with as much of this data as possible. This proverbial “super mutt” of software, or Brown Dog, will serve as a low level data infrastructure to interface with digital data contents and through its capabilities enable a new era of science and applications at large. The broader impact of this work is in its potential to serve not just the scientific community but the general public, as a DNS for data, moving civilization towards an era where a user’s access to data is not limited by a file’s format or un-curated collections.

Emmanuel Jeannot, Esteban Meneses-Rojas, Guillaume Mercier, François Tessier and Gengbin Zheng

Communication and Topology-aware Load Balancing in Charm++ with TreeMatch

Abstract—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 two load balancers for Charm++ that take into account both aspects depending on the fact that the application is compute-bound or communication-bound. This work is based on our TREEMATCH library that compute 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 for the two classes of parallel applications.

Matthieu Dorier

CALCioM: Mitigating I/O Interferences in HPC Systems through Cross-Application Coordination

Unmatched computation and storage performance in new HPC systems have led to a plethora of I/O optimizations ranging from application-side collective I/O to network and disk-level request scheduling on the file system side. As we deal with ever larger machines, the interference produced by multiple applications accessing a shared parallel file system in a concurrent manner become a major problem. Interference often breaks single-application I/O optimizations, dramatically degrading application I/O performance and, as a result, lowering machine wide efficiency.

This talk will focuse on CALCioM, a framework that aims to mitigate I/O interference through the dynamic selection of appropriate scheduling policies. CALCioM allows several applications running on a supercomputer to communicate and coordinate their I/O strategy in order to avoid interfering with one another. In this work, we examine four I/O strategies that can be accommodated in this framework: serializing, interrupting, interfering and coordinating. Experiments on Argonne’s BG/P Surveyor machine and on several clusters of the French Grid’5000 show how CALCioM can be used to efficiently and transparently improve the scheduling strategy between two otherwise interfering applications, given specified metrics of machine wide efficiency.

Babak Behzad

Taming Parallel I/O Complexity with Auto-Tuning

We present an auto-tuning system for optimizing I/O performance of HDF5 applications and demonstrate its value across platforms, applications, and at scale. The system uses genetic algorithms to search a large space of tunable parameters and to identify effective settings at all layers of the parallel I/O stack. The parameter settings are applied transparently by the auto-tuning system via dynamically intercepted HDF5 calls. To validate our auto-tuning system, we applied it to three I/O benchmarks (VPIC, VORPAL, and GCRM) that replicate the I/O activity of their respective applications. We tested the system with different weak-scaling configurations (128, 2048, and 4096 CPU cores) that generate 30 GB to 1 TB of data, and executed these configurations on diverse HPC platforms (Cray XE6, IBM BG/P, and Dell Cluster). In all cases, the auto-tuning framework identified tunable parameters that substantially improved write performance over default system settings. We consistently demonstrate I/O write speedups between 2x and 100x for test configurations.

Yves Robert, ENS Lyon, INRIA & Univ. Tenn. Knoxville

Assessing the impact of ABFT & Checkpoint composite strategies

Algorithm-specific fault tolerant approaches promise unparalleled scalability and performance in failure-prone environments. With the advances in the theoretical and practical understanding of algorithmic traits enabling such approaches, a growing number of frequently used algorithms (including all widely used factorization kernels) have been proven capable of such properties. These algorithms provide a temporal section of the execution when the data is protected by it's own intrinsic properties, and can be algorithmically recomputed without the need of checkpoints. However, while typical scientific applications spend a significant fraction of their execution time in library calls that can be ABFT-protected, they interleave sections that are difficult or even impossible to protect with ABFT. As a consequence, the only fault-tolerance approach that is currently used for these applications is checkpoint/restart. In this talk, we propose a model and a simulator to investigate the behavior of a composite protocol, that alternates between ABFT and checkpoint/restart protection for effective protection of each phase of an iterative application composed of ABFT-aware and ABFT-unaware sections. We highlight this approach drastically increases the performance delivered by the system, especially at scale, by providing means to rarefy the checkpoints while simultaneously decreasing the volume of data needed to be saved in the checkpoints.

Prasanna Balaprakash

Active-Learning-based Surrogate Models for Empirical Performance Tuning

Performance models have profound impact on hardware-software co-design, architectural explorations, and performance tuning of scientific applications. Developing algebraic performance models is becoming an increasingly challenging task. In such situations, a statistical surrogate-based performance model, fitted to a small number of input-output points obtained from empirical evaluation on the target machine, provides a range of benefits. Accurate surrogates can emulate the output of the expensive empirical evaluation at new inputs and therefore can be used to test and/or aid search, compiler, and autotuning algorithms. We present an iterative parallel algorithm that builds surrogate performance models for scientific kernels and work-loads on single-core and multicore and multinode architectures. We tailor to our unique parallel environment an active learning heuristic popular in the literature on the sequential design of computer experiments in order to identify the code variants whose evaluations have the best potential to improve the surrogate. We use the proposed approach in a number of case studies to illustrate its effectiveness.

Greg Bauer

Applications and their challenges on Blue Waters
The leadership class Blue Waters system is providing petascale level computational and I/O capabilities to its partners. To date there are approximately 32 teams using Blue Waters to pursue their science and engineering on 22,640 Cray XE CPU compute nodes and 4,224 Cray XK GPU nodes with a 26 PB, 1 TB/s filesystem. The challenges encountered by the teams are as varied as the applications running on Blue Waters. This talk will provide an overview of the Blue Waters system, its recent upgrade in GPU computing capability and network dimension, and a discussion of the applications and their challenges computing at scale on Blue Waters.

Yushan Wang
Solving 3D incompressible Navier-Stokes equations on hybrid CPU/GPU systems.

The Navier-Stokes equations are the fundamental bases of many computational fluid dynamics problems. In this presentation, we will talk about a hybrid multicore/GPU solver for the incompressible Navier-Stokes equations with constant coefficients, discretized by the finite difference method. We use the prediction-projection method which transforms the Navier-Stokes problem into Helmholtz-like and Poisson problems. Efficient solvers for the two subproblems will be presented with implementations which take advantages of GPU accelerators. We will also provide numerical experiments on a current hybrid machine.

Arnaud Legrand
SMPI: Toward Better Simulation of MPI Applications
We will present our last result on the SMPI/SimGrid framework. SMPI now implements all the collective algorithms and selection logics of both OpenMPI and MPICH and even a few other collective algorithms from Star MPI. Together with a flexible network model and topology description mechanisme, this allowed us to obtain almost perfect prediction of NASPB and BigDFT on Ethernet/TCP based clusters. We are currently working on extending this work to other kind of networks as well as on mixing the emulation capability of SMPI with the trace replay mechanism. We are also working on improving the replay mechanism so that it handles seamlessly classical trace formats.
Wesley Bland
Fault Tolerant Runtime Research at ANL
Fault tolerance has been presented as an emerging problem for decades, with researchers often claiming that the next generation of hardware will introduce new levels of failure rates that will destroy productivity and cause applications to become unusable. While it is true that as machines have scaled, resilience has become more and more of a concern, there are issues already affecting applications at current scales. Process failure remains a concern, though primarily for applications that can run at the largest scales or on very unstable hardware. For smaller applications however, there are other concerns, such as soft errors, performance loss, etc. This talk will cover some of the research being performed in the Programming Models and Runtime Systems group at Argonne National Laboratory to study these phenomena.

Jed Brown and Debojyoti Ghosh

Fast solvers for implicit Runge-Kutta systems
Implicit Runge-Kutta methods offer very high order accuracy, excellent stability properties, and optional symplecticity at the expense of needing to solve a coupled system of equations. In the past, this has been seen as a detractor and implicit RK methods have received little attention in the large-scale computing world, apart from recent interest in Spectral Deferred Correction (SDC) methods which are a particular iterative method for solving implicit RK systems, but the work scales quadratically in the number of stages and SDC is rarely more efficient than conventional sequential time stepping. Implicit RK systems have tensor product structure $$ S \otimes I + I \otimes J $$ where $S = (h A)^{-1}$ comes from the $s\times s$ Butcher table A, and J is the (typically sparse) Jacobian of the spatial discretization. Diagonalization of S was proposed independently by Butcher (1976) and Bickert (1977) as a solution method, leading to s decoupled sparse systems, each with a different (complex-valued) diagonal shift, and quickly became the standard approach in the ODE community. Instead of distributing the stages, we permute the multivector and solve all stages at once using preconditioned iterative methods that achieve much higher machine utilization due to a computational structure similar to solving a single linear system with multiple right hand sides.

Mohamed Slim Bougerra

Failure prediction: what to do with unpredicted failures ?

As large parallel systems increase in size and complexity, failures are inevitable and exhibit complex space and time dynamics. Several key results have demonstrated that recent advances in event log analysis can provide precise failure prediction. The state of the art in failure prediction provides a ratio of correctly identified failures to the number of all predicted failures of over 90\% and able to discover around 50\% of all failures in a system. However, large parts of failures are not predicted and are considered as false negative alerts. Therefore, developing efficient fault tolerance strategies to tolerate failures requires a good perception and understanding of failure prediction characteristics. To understand the properties of false negative alerts, we conducted a statistical analysis of the probability distribution of such alerts and their impact on fault tolerance techniques. Specifically we studied failures logs from different HPC production systems. We show that (i) the false negative distribution has the same nature as the failure distribution (ii) After adding failure prediction, we were able to infer statistical models that describe the inter-arrival time between false negative alerts and hence current fault tolerance can be applied to these systems. Moreover, we show that the current failures traces have a high correlation between the failure inter-arrival time that can be used to improve the failure prediction mechanism. Another important result is that checkpoint intervals for unpredicted failures can be computed from the existing high-order Daly's formula. We show how we can apply the proposed statistical-model to combine proactive migration and preventive checkpoints. Trace based simulations show that the proposed combination leads to an improvement of the execution useful work by more than 13\% with only 45\% of recall.

Dries Kimpe

Mercury: Enabling Remote Procedure Call for High-Performance Computing

Remote procedure call (RPC) is a technique that has been largely adopted by distributed services. This technique, now more and more used in the context of high-performance computing (HPC), allows the execution of routines to be delegated to remote nodes, which can be set aside and dedicated to specific tasks. However, existing RPC frameworks assume a socket-based network interface (usually on top of TCP/IP), which is not appropriate for HPC systems, because this API does not typically map well to the native network transport used on those systems, resulting in lower network performance. In addition, existing RPC frameworks often do not support handling large data arguments, such as those found in read or write calls. We present in this paper an asynchronous RPC interface, called Mercury, specifically designed for use in HPC systems. The interface allows asynchronous transfer of parameters and execution requests and provides direct support of large data arguments. Mercury is generic in order to allow any function call to be shipped. Additionally, the network implementation is abstracted, allowing easy porting to future systems and efficient use of existing native transport mechanisms.

Bill Kramer

Is Petascale Completely Done? What Should We Do Now?

Abstract: As Blue Waters approaches it first anniversary of acceptance, this talk will present the 10 most surprising lessons we learned so far from the worlds first sustained petascale system. The talk will then offer the 10 most surprising areas the HPC community should be addressing for future large scale systems.

John Towns

Applications Challenges in the XSEDE Environment

XSEDE provides access to an evolving portfolio of high end computing resources, among many other resources and services to a large community of researcher. Currently, there are more than 7,000 open individual accounts across all XSEDE systems. In this talk, we will look at the leading platforms in recent times for XSEDE (Kraken and Stampede) and discuss some of the challenges faced in bringing application up on them at scale.

Tatiana Martsinkevich

On the feasibility of message logging in hybrid hierarchical FT protocols

Frederic Vivien

Scheduling tree-shaped task graphs to minimize memory and makespan

This work investigates the execution of tree-shaped task graphs using multiple processors. Each edge of such a tree represents a large data. A task can only be executed if all input and output data fit into memory. Such trees arise in the multifrontal method of sparse matrix factorization. The maximum amount of memory needed depends on the execution order of the tasks. With one processor,
the problem of finding the tree traversal with minimum required memory was well studied and optimal polynomial algorithms have been proposed. Here, we extend the problem by considering multiple processors. With the multiple processors comes the additional objective to minimize the makespan. Not surprisingly, this problem proves to be much harder. We study its computational complexity and provide an inapproximability result even for unit weight trees. Several heuristics are proposed, especially for the realistic problem of minimizing the makespan under a strong memory constraint. They are analyzed in an extensive experimental evaluation using realistic trees.

Maria Garzaran

Optimization by Run-time Specialization for Sparse Matrix-Vector Multiplication

Abstract: Run-time specialization is the process of generating programs based on information available only at run time. This technique has the potential of generating highly efficient codes at the expense of the overheads of the run-time code generation. It is applicable when some input data is used repeatedly while other input data varies. In this talk, I explore the potential for obtaining speedups for sparse matrix dense vector multiplication using runtime specialization, in the case where a single matrix is to be multiplied by many vectors. We experiment with several methods involving run-time specialization, comparing them to methods that do not (including INTEL's MKL library). For this talk, my focus is the evaluation of the speed-ups that can be obtained with run-time specialization without considering the overheads of the code generation. Our experiments use several matrices from the Matrix Market and the University of Florida Sparse Matrix Collection and run on several machines. In most cases, the specialized code runs faster than any version without specialization. The best method depends on the matrix and machine; no method is best for all matrices and machines.

Jean-François Mehaut

From Multicores to Manycores Processors: Challenging Programming Issues with the MPPA/KALRAY
Joint work with M. Castro (UFRGS), E. Francesquini (USP), T. Messi (Yaoundé 1), J-F. Méhaut (UJF-CEA)

The exponential growth in processor performance seems to have reached a turning point. Nowadays, energy efficiency is as important as performance and has become a critical aspect to the development of scalable systems. These strict energy constraints paved the way for the development of multi and manycore processors. In this presentation we analyze a well-known irregular NP-complete problem, the Traveling-Salesman Problem (TSP). This study investigates two aspects of the TSP on multicore, NUMA, and many-core processors. First, we concentrate on the nontrivial task of adapting this application to a manycore, specifically the novel MPPA-256 manycore processor. Then, we analyze its performance and energy consumption on different platforms that comprise general-purpose and low-power multicores, a NUMA machine, and the MPPA-256 manycore. Our results show that applications able to fully use the resources of a manycore can have better performance and may consume 9.8 and 13 times less energy when compared to low-power and general-purpose multicore processors, respectively.

Pierre Jolivet

Scalable Domain Decomposition Preconditioners For Heterogeneous Elliptic Problems
Domain decomposition methods are, alongside multigrid methods, one of the dominant paradigms in contemporary large-scale partial differential equation simulation. I will present a lightweight implementation of a theoretically and numerically scalable preconditioner in the context of overlapping methods. The performance of this work is assessed by numerical simulations executed on thousands of cores, for solving various highly heterogeneous elliptic problems in both 2D and 3D with billions of degrees of freedom. Such problems arise in computational science and engineering,
in solid and fluid mechanics. This framework can also be used for building substructuring preconditioners and for pipelining communication during an iterative process such as a Krylov method.

Vincent Baudoui

Round-off error propagation and non-determinism in parallel applications

Round-off errors coming from numerical calculation finite precision can lead to catastrophic losses in significant numbers when they accumulate. Their propagation throughout a computation needs to be studied in order to ensure results accuracy. We present a round-off error estimation method based on first order derivatives that can help following error propagation in an execution graph and identifying the sensitive sections of a code. It has been experimented on well known LU decomposition algorithms. In a second part, we focus on the effects of non-determinism in parallel applications where messages exchanged between processes are received in random order, possibly leading to different round-off error accumulations and subsequently to different results at each execution. We study the impact of this non-reproducibility on the convergence of stencil computations after a failure and recovery event.

Jeremy Enos

Application Runtime Consistency and Performance Challenges on a shared 3D torus.

Early testing on Blue Waters revealed varied performance for some applications making required walltimes unpredictable. Many potential causes were investigated, ultimately indicating that poor placement on to compute resources within the 3D torus network was a chief aggravating factor. Multiple thrusts of effort were launched to improve both application performance and consistency; a long term topology-aware placement development plan, improved high speed network monitoring, and immediate "stop gap" measures available within already existing tools and methods.

Ana Gainaru

Topology and behaviour aware failure prediction for Blue Waters.
Failure prediction has made substantial progress in the last 5 years and current studies have shown that failure avoidance techniques could give high benefits when combined with classical fault tolerance protocols. Understanding the properties of a prediction module and exploiting them for enhancing fault tolerance approaches and scheduling decisions is crucial for providing scalable solutions to deal with failures on future HPC systems.
Recently, we have presented a novel methodology for truly online failure prediction for the Blue Water system. In this talk we described the main bottlenecks and limitations faced in applying failure prediction on a petascale system and proposed a couple of solutions by using topology-level information.
Moreover, we will show that on a real system, system failures are not very frequently translated into application failures. We will present how this is influencing application level failure prediction and future system performance degradation analysis.

Sheng Di
Optimization of Multi-level Checkpoint Model for Large Scale HPC Applications
HPC community projects that future extreme scale systems will be much less stable than current Petascale systems, thus requiring sophisticated fault tolerance to guarantee the completion of large scale numerical computations. Execution failures may occur due to multiple factors with different scales, from transient uncorrectable memory errors localized in processes to massive system outages. Multi-level checkpoint/restart is a promising model that provides an elastic response to tolerate different types of failures. It stores checkpoints at different levels: e.g., local memory, remote memory, using a software RAID, local SSD, remote file system. This talk will respond to two open questions: 1) how to optimize the selection of checkpoint levels based on failure distributions observed in a system, 2) how to compute the optimal checkpoint intervals for each of these levels. (1) A mathematical model is formulated to fit the multi-level checkpoint/restart mechanism with large scale applications regarding various types of failures. (2) The entire execution performance of each parallel application is theoretically optimized by selecting the best checkpoint level combination and corresponding checkpoint intervals at different levels. (3) The proposed optimal solutions is evaluated using both simulation and real environment with real-world MPI programs running on hundreds of cores. Experiments show that optimized selections of levels associated with optimal checkpoint intervals at each level outperforms other state-of-the-art solutions by 5-50 percent.
Rafael Keller Tesser
Using AMPI to improve the performance of the Ondes3D seismic wave simulator through dynamic load balancing
Ondes3D is a seismic wave propagation model, which is used to analyze the consequences of future earthquakes. This model presents some challenges in terms of load-balancing, especially due to boundary conditions. These conditions are executed on the borders of the simulated domain, to absorb the outgoing energy. So, the amount of computation in the borders of the domain is greater than in its center. Thus, when we divide the domain, to parallelize the execution, we end up with load unbalanced subdomains. In this work, we investigated the use of dynamic-load balancing to deal with this problem. For this purpose, we ported Ondes3D to Adaptive MPI (AMPI). This way, we can take advantage of the load-balancing framework provided by its runtime. We

evaluated the performance of our AMPI version of Ondes3D, using different load-balancers. In our best case, the application ran 23.85% faster than the original MPI implementation. Moreover, the load balancers were able to adapt to the variation in load balancing caused by the propagation of the wave through the simulated region.

Eric Bohm

A Multi-resolution Emulation + Simulation Methodology for Exascale

As we design exascale applications and machines, it becomes important to be able to analyze and experiment with alternate designs of both machines and applications. These experiments have to be done before the machines are built since it will be too expensive to build a large number of alternate designs. One of the challenges in this process is how to represent application behavior in such machines. For analyzing network performance via simulations of dynamic applications,the feedback that occurs naturally in applications must be simulated: if an incoming message is late, the ordering of events may change, and outgoing message injection will also change. To achieve a high fidelity simulation is therefore challenging.

We will discuss one promising method to address this problem, emulation-followed-by-simulation, in which one carries out a full-scale emulation of the application with the correct number of nodes and control threads, facilitated by some overdecomposition based system such as Charm++. The emulation captures dependencies between sequential computations and remote data in traces. Trace data can be further constrained, using a variety of techniques, to capture steady state behavior and phases of interest.

The traces generated by emulation can then be fed to a multi-component simulator, where a variable resolution simulation can be carried out to predict performance and other attributes. We advocate this methodology and elaborate on research challenges involved in following it in exascale design. At exascale, we expect that several components, which are pluggable entities similar to those used in existing frameworks, such as BigSim and SST, will simulate network, resilience support, power management, thermal constraints, operating system overhead and file system performance. In addition, the adaptive runtime system, essential for scalable execution at exascale, needs to be (and can be) simulated in detail, with realistic code and strategies, in order to attain high fidelity. The runtime system itself can use modeling and/or simulation to predict computation and communication patterns to facilitate adaptive runtime control strategies.

Gabriel Antoniu

A-Brain and Z-CloudFlow: Scalable Data Processing on Azure Clouds - Lessons Learned in Three Years and Future Directions

Joint acquisition of neuroimaging and genetic data on large cohorts of subjects is a new approach used to assess and understand the variability that exists between individuals, and that has remained poorly understood so far. As both neuroimaging- and genetic-domain observations represent a huge amount of variables (in the order of millions), performing statistically rigorous analyses on such amounts of data is a major computational challenge that cannot be addressed with conventional computational techniques only. The A-Brain project was started in October 2010 within the Microsoft Research-INRIA Joint Research Center with the goal of addressing the above computational and data processing challenges using MapReduce-related cloud techniques on Microsoft's Azure cloud infrastructure. This talk draws the conclusions of three years of investigation of the benefits of using the cloud for large-scale application experiments such as the genetics-neuroimaging data comparisons. It also gives the main lines of the future work just started within Z-CloudFlow, a follow-up project just started, dedicated to scalable data processing for cloud workflows running across multiple data centers.

Leonardo Bautista Gomez

Detecting Silent Data Corruption through Data Dynamic Monitoring for Scientific Applications

We propose a novel technique to detect silent data corruption based on low-overhead, localized data monitoring. We implemented our technique on a generic library that allows scientific applications to easily self-analyze during runtime. Using this technique, an application an learn the normal dynamics of its datasets, allowing it to quickly spot anomalies. We evaluate our technique with synthetic enchmarks and large scientific datasets of production-level scientific applications simulating real phenomena. We show that our technique can detect up to 50% of injected errors while incurring only negligible overhead on real scientific applications.

Pavan Balaji

Message Passing in Massively Multithreaded Environments

Many-core architectures, such as the IBM Blue Gene/Q and Intel Xeon Phi, provide dozens of cores and hundreds of hardware threads. To utilize such architectures, application programmers are increasingly looking at hybrid programming models (frequently referred to as ``MPI+X'' models), where multiple threads interact with the MPI library. A common mode of operation for hybrid MPI+threads applications is where multiple threads are used to parallelize the computation, and one or more threads also issues MPI operations. While such a model is becoming increasingly popular because of the reducing per-core hardware resources available in modern architectures, it poses several challenges for the efficiency of MPI communication in such environments. In this talk, I’ll describe some of our recent work on optimizing MPI in such environments, either with multiple threads calling MPI operations or a single thread doing so.

Kate Keahey

Evaluating Streaming Strategies for Event Processing across Infrastructure Clouds

Infrastructure clouds revolutionized the way in which we approach resource procurement by providing an easy way to lease compute and storage resources on short notice, for a short amount of time, and on a pay-as-you go basis. This new opportunity however introduces new performance trade-offs. Making the right choices in leveraging different types of storage available in the cloud is particularly important for applications that depend on managing large amounts of data within and across clouds. An increasing number of such applications conform to a pattern where data processing relies on streaming the data to a compute platform where a set of similar operations is repeatedly applied to independent chunks of data. This pattern is evident in virtual observatories such as the Ocean Observatory Initiative, in cases when new data is evaluated against existing features in geospatial computations, or when experimental data is processed as a series of time events. In this presentation, we propose different strategies for efficiently implementing such streaming in the cloud and evaluate them in the context of an Atlas application processing experimental data. Our results show that choosing the right cloud configuration can improve overall application performance by as much as four times.

Grigori Fursin

Collective Mind: making auto-tuning practical using crowdsourcing and predictive modeling

Software and hardware optimization and co-design of computer systems becomes intolerably complex, ad-hoc, time consuming and error prone due to enormous number of available design and optimization choices, complex interactions between all software and hardware components, and ever changing tools and applications. We present our novel long-term holistic and practical solution to address these problems using new plugin-based Collective Mind infrastructure and repository. For the first time, it can preserve the whole experimental setup and all associated artifacts to distribute program analysis and multi-objective optimization among many participants while utilizing any available smart phone, tablet, laptop, cluster or data center, and continuously observing, classifying and modeling realistic their behavior. Any unexpected behavior is analyzed using shared data mining and predictive modeling plugins or exposed to the community at a public portal cTuning.org and repository c-mind.org/repo for collaborative explanation. Gradually increasing public optimization knowledge helps to continuously improve optimization heuristics of any compiler, predict optimizations for new programs or suggest efficient run-time adaptation strategies depending on end-user requirements. We successfully validated this approach and framework in several academic and industrial projects while releasing hundreds of codelets, numerical applications, data sets, models, universal experimental pipelines, and unified tools to start community-driven, systematic and reproducible R&D to build adaptive, self-tuning computer systems, and initiate new publication model where experiments and techniques are continuously validated and improved by the community.

Wen-Mei Hwu

A New, Portable Algorithm Framework for Parallel Linear Recurrence Problems

Linear recurrence solvers are common constructs in a class of important scientific applications. Many parallel algorithms have been proposed to achieve high performance for different problems that are linear recurrence in nature. Through a detailed investigation of the existing parallel implementations, we identify a general, hierarchical parallel linear recurrence algorithm that has the potential to fully utilize a wide variety of hardware. However, this algorithm is complex and requires enormous programming efforts to achieve high performance across different architectures. To achieve single source performance portability, we create a code-generator using auto-tuning for optimizing high-performance, parallel, linear recurrence solvers that are retargetable to specific platforms. The framework is composed of two major components. The first component is an auto-tuned tiling procedure which generates tiling by searching a unified tiling space (UTS). The UTS combines on-chip memory resources to simplify the complexity of tiling decisions. Based on the tiling decision, the second component selects the best communication implementation to minimize the communication overhead. By heuristically reducing the search space, our auto-tuning technique generates optimized programs in a reasonable time. We evaluate our framework using several benchmarks including prefix sum, IIR filter, bidiagonal solver and tridiagonal solver on GPU architectures. The resulting linear recurrence solvers significantly outperforms the previous state-of-the-art, specialized GPU implementations.

François Pellegrini

Parallel repartitioning and remeshing : results and prospects

The purpose of this talk is to expose the current state and the prospects of research and of implementation regarding two software tools that we develop for HPC : PT-Scotch and PaMPA. PT-Scotch is a parallel partitionning and mapping tool that has been recently extended to provide dynamic remapping features. While its algorithms have been developed with scalability in mind, several algorithmic bottelnecks appear, which impose to re-think the way we perform repartitioning. PaMPA is a library for parallel (re)meshing of distributed, unstructured meshes, that delegates (re)partitioning to PT-SCOTCH. After basic mesh handling features were developed, we focused on parallel remeshing itself, allowing us to produce distributed, tetraedral meshes comprising several hundred million elements.

Venkatram Vishwanath

Addressing I/O Bottlenecks and Simulation-Time Analytics at Extreme Scales

We will first present our work in GLEAN - a flexible and extensible framework that takes application, analysis, and system characteristics into account to facilitate simulation-time data analysis and I/O acceleration. The GLEAN infrastructure hides significant details from the end user, while at the same time providing a flexible intterface to the fastest path for their data and analysis needs and, in the end, scientific insight. We describe the efficacy of our approaches in scaling to 768K cores of the Mira BG/Q system, and on the Cray supercomputer. If time permits, we will present our work on Concerted Flows - A parallel data movement infrastructure that takes into account analytical and empirical models of an end-to-end system infrastructure together with mathematical optimization to improve the achievable performance for parallel data flows at various system scales.

Luke Olson

Toward a more robust sparse solver with some ideas on resilience and scalability

In this talk we look at some recent attempts to improve robustness in algebraic multigrid solvers for a wider range of problems. In particular we look at optimality throughout the solver by refining interpolation and the sense of strength in the method. With this we comment on some current directions for improving scalability by thinning the hierarchy and some possibilities for strengthening resilience.

Torsten Hoefler

Using Automated Performance Modeling to Find Scalability Bugs in Complex Codes

Many parallel applications suffer from latent performance limitations that may prevent them from scaling to larger machine sizes. Often, such scalability bugs manifest themselves only when an attempt to scale the code is actually being madeâ€"a point where remediation can be difficult. However, creating analytical performance models that would allow such issues to be pinpointed earlier is so laborious that application developers attempt it at most for a few selected kernels, running the risk of missing harmful bottlenecks. In this paper, we show how both coverage and speed of this scalability analysis can be substantially improved. Generating an empirical performance model automatically for each part of a parallel program, we can easily identify those parts that will reduce performance at larger core counts. Using a climate simulation as an example, we demonstrate that scalability bugs are not confined to those routines usually chosen as kernels.