joint-lab workshop Jun. 12-14 2013

UNDER construction: The agenda below is not the final one

This event is supported by INRIA, UIUC, NCSA, ANL

Main Topics	Schedule	Speaker	Affiliation	Type of presentation	Title (tentative)	Download
Dinner Before the Workshop	7:30 PM	Only people registered for the dinner			Valpré hotel	
Workshop Day 1	**Wednesday June 12th**					
					TITLES ARE TEMPORARY (except if in bold font)	
Registration	08:00					
Welcome and Introduction **Amphitheatre**	08:30	Marc Snir + Franck Cappello	INRIA&UIUC&ANL	Background	**Welcome, Workshop objectives and organization**	Opening-9th-Workhsop.ppt
	08:45	Bill Kramer	UIUC	Background	**NCSA updates and vision of the collaboration**	Kramer-Joint Lab Workshop - 20130612-v3.pdf
	09:00	Marc Snir	ANL	Background	**ANL updates vision of the collaboration**	intro for Lyon.pdf
	09:15	Frederic Desprez	Inria	Background	**INRIA updates and vision of the collaboration**	Desprez-HPC@Inria-JLPC-0613.pdf
Big systems Chair: Christian Perez	9:30	Bill Kramer	UIUC	Background	**Update on BlueWaters**	BW Overview - Inria-Illinois Joint Workshop June 2013-v1.pdf
	10:00	Break				
	10:30	Mitsuhisa Sato	U. Tsukuba & AICS	Background	**AICS and the K computer**	aics-130612.pptx
CANCELED	11:00	Paul Gibbon	Juelich	Background	**Meeting the Exascale Challenge at the Juelich Supercomputing Centre**.	
Resilience&fault tolerance and simulation Chair: Franck Cappello	11:00	Marc Snir	ANL&UIUC	Report	**ICIS report on Resilience**	UWM resilience.pdf
	11:30	Vincent Baudoui	Total & ANL	Joint-Results	**Round-off error and silent soft error propagation in exascale applications**	Lyon_12_juin_2013_Error_propagation_in_exascale_applications_Vincent_Baudoui.pdf
	12:00	Lunch				
Numerical Algorithms Chair: Frederic Desprez	13:30	Bill Gropp	UIUC	Background	**Topics for Collaboration in Numerical Libraries**	libraries-gropp-final.pdf
	14:00	Paul Hoveland	ANL	Background	**Argonne strategic plan in applied math**	
	14:30	Marc Baboulin	INRIA	Background	**Using condition numbers to assess numerical quality in high-performance computing applications**	baboulin.pdf
	15:00	Luke Olson	UIUC	Background	**Opportunities in developing a more robust and scalable multigrid solver**	201306_JointLab.pdf

	15:30	Break				
	16:00	Frederic Nataf	INRIA& P6	Background	**Toward black-box adaptive domain decomposition methods**	talkJLPC20130612.pdf
Resilience&fault tolerance and simulation Chair: Franck Cappello	16:30	Bogdan Nicolae	IBM	Joint Result	**AI-Ckpt: Leveraging Memory Access Patterns for Adaptive Asynchronous Incremental Checkpointing**	AICkpt-9thJLPC.pdf
	17:00	Martin Quison	INRIA	Result	**Improving Simulations of MPI Applications Using A Hybrid Network Model with Topology and Contention Support**	JLPC-simgrid-smpi.pdf
	17:30	Adjourn				
	18:45	**Bus for Diner**				
Workshop Day 2	**Thursday June 13th**					
Programming Models Chair: Frederic Desprez	08:30	Jean-François Mehaut	INRIA	Result	**Progresses in the European FP7 Mont-Blanc 1 project and objectives of its follow up: Mont-Blanc 2**	
	09:00	Rajeev Thakur	ANL	Background	**Update on MPI and OS/R Activities at Argonne**	Rajeev.pdf
	09:30	Andra Ecaterina Hugo	INRIA	Results	**Composing multiple StarPU applications over heterogeneous machines: a supervised approach**	ahugo_Composability_StarPU.pdf
	10:00	Celso Mendes	UIUC	Background	**Dynamic Load Balancing for Weather Models via AMPI**	AMPI-BRAMS-JointLab2013.pdf
	10:30	Break				
Big Data, I/O, Visualization Chair: Kate Keahey	11:00	Dries Kimpe	ANL	Results	**Triton: Exascale Storage**	
	11:30	Gilles Fedak	INRIA	Result	**Active Data: A Programming Model to Manage Data Life Cycle Across Heterogeneous Systems and Infrastructures**	active-data-fedak.pdf
	12:00	Matthieu Dorier	INRIA	Joint Result	**Data Analysis of Ensemble Simulations: an In Situ Approach using Damaris**	DORIER-JLPC-06-2013.pdf
	12:30	Ian Foster	ANL	Background	**Compiler optimization for distributed dynamic data flow programs**	
	13:00	Lunch				
Mini Workshop1 **Amphitheatre**						
Resilience Chair: Marc Snir	14:00	Ana Gainaru	UIUC	Results	**Challenges in predicting failures on the Blue Waters system.**	againaru (1).pdf
	14:30	Xiang Ni	UIUC	Results	**ACR: Automatic Checkpoint/Restart for Soft and Hard Error Protection.**	JLPC_workshop_Xiang.pdf
	15:00	Tatiana Martsinkevich	INRIA & ANL	Result	**On the feasibility of message logging in hybrid hierarchical FT protocols**	Martsinkevich message logging.pdf
	15:30	Mohamed Slim Bouguerra	INRIA & ANL	Result	**Investigating the probability distribution of false negative failure alerts in HPC systems**	Slim_jointlab_icpp_presentation_v0.pdf
	16:00	Break				
	16:30	Amina Guermouche	UVSQ	Result	**Multi-criteria Checkpointing Strategies: Response-time versus Resource Utilization**	AminaGuermouche.pdf
	17:00	Thomas Ropars	EPFL	Result	**Towards efficient replication of HPC applications to deal with crash failures**	Limited access
	17h30	Mehdi Diouri	INRIA	Result	**ECOFIT: A Framework to Estimate Energy Consumption of Fault Tolerance Protocols for HPC Applications**	MehdiDiouri.pdf

	18:00	Adjourn				
Mini Workshop2 **Room: Saint Maur**						
Numerical Algorithms and Libraries Chair: Bill Gropp	14:00	Jean Utke	ANL	Result	**Designing and implementing a tool-indedendent, adjoinable MPI wrapper library**	JointLabLyon.pdf
	14:30	Laurent Hascoet	INRIA	Result	**The adjoint of MPI one-sided communications**	
	15:00	Stefan Wild,	ANL	Result	**Loud computations? Noise in iterative solvers**	wild (1).pdf
	15:30	Jed Brown	ANL	Result	**Vectorization, communication aggregation, and reu se in stochastic and temporal dimensions**	20130613-JointLab.pdf
	16:00	Break				
	16:30	Yushan Wang	INRIA P11	Result	**Accelerating incompressible fluid flows simulations using SIMD or GPU computing**	Jointlab_Lyon.pdf
	17:00	Frederic Hecht	INRIA /P6	Result	**FreeFem++, a user language to solve PDE.**	ff-lyon-2013.pdf
	18:00	Adjourn				
	18:45	**Bus for diner**			Lyon	
Workshop Day 3	**Friday June 14th**					
Mini Workshop1 (cont.) **Room: Les essarts**						
Resilience Chair: Franck Cappello.	08:30	Di Sheng	INRIA	Result	**Optimization of Google Cloud Task Processing with Checkpoint-Restart Mechanism**	Lyon-workshop-sdi.ppt
	09:00	Guillaume Aupy	INRIA	Result	**On the Combination of Silent Error Detection and Checkpointing**	G-aupy-silent-errors.pdf
Mini Workshop3	09:30	Guillaume Mercier	INRIA	Result	**Topology Management and MPI Implementations Improvements**	JointLab9.pdf
	10:00	Break				
Programming and Scheduling Chair: Rajeev Thakur	10:30	Vincent Lanore	INRIA	Result	**Static 2D FFT adaptation through a component model based on Charm++**	vlanore_jointlab.pdf
	11:00	Anne Benoit	INRIA	Result	**Energy-efficient scheduling**	BenoitAnne.pdf
	11:30	François Tessi er	INRIA	Result	**Communication-aware load balancing with TreeMatch in Charm++**	Tessier_JLPC13.pdf
	12:00	**Closing**				
	12:30	Lunch				
Mini Workshop2 (cont.) **Room: Saint Maur**						
Numerical Algorithms and Libraries Chair: Paul Hovland	08:30	François Pellegrini	INRIA	Result	**Shared memory parallel algorithms in Scotch 6**	inria-uiuc_20130614.pdf
	09:00	Abdou Guermouche	INRIA	Result	**Towards resilient parallel linear Krylov solvers**	

Mini Workshop4	09:30	Kate Keahey	ANL	Result	**Research Topics and Collaboration Opportunities in the Nimbus Team**	
Clouds Chair: Frederic desprez	10:00	Break				
	10:30	Jonathan Rouzaud-Cornabas	CNRS &INRIA	Result	**SimGrid Cloud Broker: Simulation of Public and Private Clouds**	sgcb_pres.pdf
	11:00	Christian Perez	INRIA	Result	**On Component Models to Deploy Application on Clouds**	130614_Cloud_Component (1).pdf
	11:30	Eddy Caron	INRIA	Result	**Seed4C: Secured Embedded Element and Data privacy for Cloud Federation**	
	12:00	**Closing**				
	12:30	Lunch				

Abstracts

Paul Gibbon

Meeting the Exascale Challenge at the Juelich Supercomputing Centre.

This talk will address recent developments in the field of supercomputing research at JSC, beginning with an overview of petascale hardware installed since 2009 together with our present user support infrastructure. Over the coming 5 years the JSC roadmap for exascale computing will leverage the work performed in three `Exascale Centres' - the Exascale Innovation Lab (with IBM), Exa-Cluster Lab (Intel, Partec) and NVIDIA Lab, jointly staffed with the respective industrial partners. Software support will continue to revolve around our `Simulation Laboratories' and Cross-Sectional Teams, providing high-level algorithmic expertise in a number of disciplines such as climate research, energy materials and life sciences, all strongly represented at FZ-Jülich. These and other selected research activities will be briefly reviewed.

Martin Quison

Improving Simulations of MPI Applications Using A Hybrid Network Model with Topology and Contention Support

Proper modeling of collective communications is essential for understanding the behavior of medium-to-large scale parallel applications, and even minor deviations in implementation can adversely affect the prediction of real-world performance. We propose a hybrid network model extending LogP based approaches to account for topology and contention in high-speed TCP networks. This model is validated within SMPI, an MPI implementation provided by the SimGrid simulation
toolkit. With SMPI, standard MPI applications can be compiled and run in a simulated network environment, and traces can be captured without incurring errors from tracing overheads or poor clock synchronization as in physical experiments. SMPI provides features for simulating applications that require large amounts of time or resources, including selective execution, ram folding, and off-line replay of execution traces. We validate our model by comparing traces produced by SMPI with those from other simulation platforms, as well as real world environments.

Frederic Nataf

Toward black-box adaptive domain decomposition methods

Domain decomposition methods address in a natural and powerful way modern parallel architectures. In order to be scalable, these methods involve coarse spaces. These coarse spaces are specifically designed for the two-level methods to be scalable and robust with respect to the coefficients in the equation and the choice of the decomposition. We achieve this in an automatic way by solving generalized eigenvalue problems on the interfaces between subdomains to identify the modes which slow down convergence.This construction allows for a black-box implementation. Theoretical bounds for the condition numbers of the preconditioned operators which depend only on a chosen threshold and the maximal number of neighbours of a subdomain are presented and proved. Scalable implementations on HPC platforms make it possible to solve problems with several billions of unknowns in three dimensions using FreeFem++ DSL for finite element simulations.

Marc Baboulin

Using condition numbers to assess numerical quality in high-performance computing applications

We explain how condition numbers of problems can be used to assess the quality of a computed solution. We illustrate our approach by considering the example of overdetermined linear least squares (linear systems being a special case of the latter). Our method is based on deriving exact values or estimates for the condition number of these problems. We describe algorithms and software to compute these quantities using standard parallel libraries. We present numerical experiments in a physical application and we propose performance results using new routines on top of the multicore-GPU library MAGMA.

Jean François Mehaut

Progresses in the European FP7 Mont-Blanc 1 project and objectives of its follow up: Mont-Blanc 2

Amina Guermouche

Multi-criteria Checkpointing Strategies: Response-time versus Resource Utilization

Failures are increasingly threatening the efficiency of HPC systems, and current projections of Exascale platforms indicate that rollback recovery, the most convenient method for providing fault tolerance to general-purpose applications, reaches its own limits at such scales. One of the reasons explaining this unnerving situation comes from the focus that has been given to per-application completion time, rather than to platform efficiency. In this talk, we discuss the case of uncoordinated rollback recovery where the idle time spent waiting recovering processors is used to progress a different, independent application from the system batch queue. We then propose an extended model of uncoordinated checkpointing that can discriminate between idle time and wasted computation. We instantiate this model in a simulator to demonstrate that, with this strategy, uncoordinated checkpointing per application completion time is unchanged, while it delivers near-perfect platform efficiency.

Anne Benoit

Energy-efficient scheduling

In this talk, I will survey recent works on energy-efficient scheduling. The goal is to minimize the energy consumption of a schedule, given some performance constraints, for instance a bound on the total execution time. I will first revisit the greedy algorithm for independent tasks in this context. Then I will present problems accounting for the reliability of a schedule: if a failure may occur, then replication or checkpoint is used to achieve a reliable schedule. The goal remains the same, i.e., minimize the energy consumption under performance constraints

Jean Utke

Designing and implementing a tool-indedendent, adjoinable MPI wrapper library

The efficient computation of gradients by the "adjoint-mode" of algorithmic differentiation (AD) entails the inversion of MPI communication graphs. The logic to be implemented for adjoining non-blocking communication patterns is sufficiently complex to warrant a design of components that is independent o f the algorithmic differentiation tool that provides the context in which the adjoint communication is to take place. We discuss (i) how we account for the different data models implied by the AD tool as well as the target language, (ii) the implementation choices among the possible adjoint communications, and (iii) the currently known limitations of our approach. We hope for feedback from the community regarding this design particularly with respect to performance and current developments in the MPI standard.

Laurent Hascoet

The adjoint of MPI one-sided communications
Computing gradients of numerical models by the adjoint mode of algorithmic differentiation is a crucial ingredient for model optimization, sensitivity analysis, and uncertainty quantification of many large-scale science and engineering applications. The adjoint mode implies a reversal of the data dependencies and consequently a reversal of communications in parallelized models. Building on previous studies regarding the adjoining of MPI two-sided communications, we investigate the construction of adjoints for certain one-sided MPI communications

Mehdi Diouri

ECOFIT: A Framework to Estimate Energy Consumption of Fault Tolerance Protocols for HPC Applications

Energy consumption and fault tolerance are two interrelated issues to address for designing future exascale systems. Fault tolerance protocols used for checkpointing have different energy consumption depending on parameters like application features, number of processes in the execution and platform characteristics. Currently, the only way to select a protocol for a given execution is to run the application and monitor the energy consumption of different fault tolerance protocols. This is needed for any variation of the execution setting. To avoid this time and energy consuming process, we propose an energy estimation framework. It relies on an energy calibration of the considered platform and a user description of the execution setting. We evaluate the accuracy of our estimations with real applications running on a real platform with energy consumption monitoring. Results show that our estimations are highly accurate and allow selecting the best fault tolerant protocol without pre-executing the application.

Matthieu Dorier

Data Analysis of Ensemble Simulations: an In Situ Approach using Damaris
As we approach exascale, simulations running on ever more cores on supercomputers produce ever larger data that has to be stored for subsequent analysis. With unmatched storage and computation performance, in situ analysis has been proposed as a way to run analysis tasks along with the running simulation. While this reduces the need to store massive amounts of raw data and lets scientists get a direct insight into their simulation, it does not allow to compare multiple runs of the same simulation (ensemble simulations), as these runs are not performed at the same moment. Thus in situ approaches remain limited and ensemble simulations still requires to store raw data. We present a complete framework for comparing data produced by different runs of the same simulation. This framework uses the Damaris I/O middleware to re-load data from previous experiments inside a running instance of the simulation, allowing a direct in situ comparison of data between older and current runs.

Gille Fedak

Active Data: A Programming Model to Manage Data Life Cycle Across Heterogeneous Systems and Infrastructures
The Big Data challenge consists in managing, storing, analyzing and visualizing these huge and ever growing data sets to extract sense and knowledge. As the volume of data grows exponentially, the management of these data becomes more complex in proportion. A key point is to handle the complexity of the data life cycle, i.e. the various operations performed on data: transfer, archiving, replication, deletion, etc. To alleviate the complexity of the data life cycle, we propose Active Data, a programming model to automate and improve the expressiveness of data management applications. We first introduce the concept of data life cycle and define a formal model that allow to expose data life cycle across heterogeneous systems and infrastructures. The Active Data programming model allows code execution at each stage of the data life cycle: routines provided by programmers are executed when a set of events (creation, replication, transfer, deletion) happen to any data. We implement and evaluate the model with four use cases: a storage cache to Amazon-S3, a cooperative sensor network, an incremental implementation of the MapReduce programming model and automated data provenance tracking across heterogeneous systems. Altogether, these scenarios illustrate the adequateness of the model to program applications that manage distributed and dynamic data sets. We also show that applications that do not leverage on data life cycle can benefit from Active Data to improve their performances.

Francois Pellegrini

Shared memory parallel algorithms in Scotch 6

The Scotch software package comprises two libraries: the Scotch sequential library, and the PT-Scotch parallel library. The latter is based on a distributed memory paradigm, and uses MPI to exchange data between processes. The advent of many-core, shared memory, machines imposes to reconsider this approach. The complexity of graph partitioning algorithms is low compared to factorization. A first solution is to reduce communication overhead by running graph partitioning only on a limited number of nodes. A second solution is to make graph partitioning algorithms more efficient, by reducing communication overhead and resorting to shared memory parallelism. This talk will present our first experiments in this direction.

Vincent Baudoui

Round-off error and silent soft error propagation in exascale applications

Future exascale computers will open up new perspectives in numerical simulation, but they will also experience more errors because of their massive scale. We will focus here on round-off errors and on silent soft errors, of which propagation needs to be studied in order to ensure results accuracy. Round-off errors come from numerical calculation finite precision and can lead to catastrophic losses in significant numbers when they accumulate. We will discuss the limits of existing error bounds when facing large scale problems. Soft hardware errors can also perturb computations by randomly flipping memory bits. Some of these errors are automatically corrected but others can propagate silently through the calculations. We will present some strategies to determine the sensitive sections of an application as part of future research work.

Bogdan Nicolae

AI-Ckpt: Leveraging Memory Access Patterns for Adaptive Asynchronous Incremental Checkpointing

With increasing scale and complexity of supercomputing and cloud computing architectures, faults are becoming a frequent occurrence, which makes reliability a difficult challenge. Although for some applications it is enough to restart failed tasks, there is a large class of applications where tasks run for a long time or are tightly coupled, thus making a restart from scratch unfeasible. Checkpoint-Restart (CR), the main method to survive failures for such applications faces additional challenges in this context: not only does it need to minimize the performance overhead on the application due to checkpointing, but it also needs to operate with scarce resources. To this end, this paper contributes with a novel approach that leverages both the current and past memory access pattern in order to optimize the order in which memory pages are flushed to stable storage during asynchronous checkpointing. Large scale experiments show up to 60% improvement when compared to state-of-art checkpointing approaches, all this achievable with an extra memory requirement of less than 5% of the total application memory.

Bill Gropp

Topics for Collaboration in Numerical Libraries

This talk will discuss some open problems in numerical libraries for extreme scale systems, including issues currently facing some of the application teams that are currently using the Blue Waters sustained petascale system.

Luke Olson

Opportunities in developing a more robust and scalable multigrid solver

Multigrid methods have increased in robustness in recent years due to new algorithmic advances and new theoretical developments. The result is a more robust multilevel framework leading to improved convergence for a wider range of non-elliptic problems. Yet, many of these developments have not been adapted at scale despite their intended use while many of the optimizations could be strengthened by considering the high-perfromance computing architectures more directly. In this talk, we discuss a particular example of these recent optimizations in multigrid, to define optimal interpolation, that moves toward a more general framework, and highlight some focused directions for collaboration in this respect. In addition, recent trends in highthrouput computing have motivated algorithmic changes in the multigrid design. In this talk, we will also highlight some directions to futher advance multigrid solvers at scale based on this work with collaborion through the Joint Lab.

Paul Hovland

Argonne strategic plan in applied math

Jed Brown

Vectorization, communication aggregation, and reuse in stochastic and temporal dimensions

Transformative computing in science and engineering involves problems posed in more than just the spatial domain: temporal, stochastic, and parameter spaces also play a role. Current methods for solving such problems are predominantly based on the concept that the fundamental building block is the solution of a deterministic PDE model, or perhaps one time step of a transient model. This is practical: it permits comfortable partitioning of mathematical analysis and relatively unintrusive software interfaces, but it eagerly chooses which dimensions are treated sequentially, which are distributed in parallel, etc. These imposed choices leave developers of the PDE models banging their heads against the familiar challenges of efficiently utilizing increasingly precious memory bandwidth, hiding and reducing synchronization costs, and obtaining vectorization. Meanwhile, the stochastic and temporal dimensions provide structure that is ideally suited to extreme-scale architectures, if only they could be promoted to first-class citizens, alongside the spatial dimensions, in algorithmic analysis and in software. Exploiting this structure in ``full-space'' methods will require crosscutting development: improved convergence theory, efficient hardware-adapted algorithms, high-quality software libraries, and programming tools and run-time systems to facilitate the development of libraries and applications. In this talk, I present several examples and propose a guideline for reasoning about efficient mappings of full-space analysis onto parallel computers.

Celso Mende

Dynamic Load Balancing for Weather Models via AMPI

Load imbalances can severely limit the scalability of a parallel application. Typically, the solution adopted to overcome this problem is to change the application code as an attempt to distribute the load more uniformly across the available processors. This solution, however, requires deep knowledge of the application, and needs to be redone as new sources of imbalance arise. In this presentation, we show how an intelligent, adaptive runtime system can help in addressing this problem. Using Adaptime-MPI, an implementation of the MPI standard based on the Charm++ runtime system, we demonstrate how to achieve a better balance without requiring major changes or much knowledge about the application. As a case-study, we show an application of this approach with weather forecasting models, which can suffer from severe imbalances due to several sources, including dynamic variations in the atmosphere. Besides presenting recent results, we also point to some remaining challenges, which make opportunities for further work in this area.

Xiang Ni

ACR: Automatic Checkpoint/Restart for Soft and Hard Error Protection.

As the scale of machines increase, the HPC community has seen a steady decrease in reliability of the systems, and hence an increase in the down time. Moreover, soft errors such as bit flips do not prevent execution but generate incorrect results. Checkpoint/restart is by far the most commonly used fault tolerance method for hard errors, and its efficiency and scalability has been improved with recent research. In this talk, we will discuss a holistic methodology for automatically detecting and recovering from soft or hard faults with minimal application intervention. This is demonstrated by ACR: an automatic checkpoint/restart framework that performs application replication and automatically adapts the checkpoint period using online information about the current failure rate. ACR performs an application- and user-oblivious recovery. We empirically test ACR by injecting failures that follow different distributions for five applications and show low overhead when scaled to 131,072 cores. We also analyze the interaction between soft and hard errors and propose three recovery schemes that explore the trade-off between performance and reliability requirements.

Thomas Ropars,

Towards efficient replication of HPC applications to deal with crash failures

Ana Gainaru

Challenges in predicting failures on the Blue Waters system.

As the size of supercomputers increases, so does the probability of a single component failure within a time frame. With the growing operation cost of extreme scale supercomputers like Blue Waters, the act of predicting failures to prevent the loss of computation hours becomes cumbersome and presents a couple of challenges not encountered for smaller systems. The talk will focus on presenting online failure prediction and analyzing the Blue Water system. We show to what extent online failure prediction is a possibility at petascale and what are the challenges in achieving an effective fault prevention mechanism for Blue Waters.

Mohamed Slim Bouguerra

Investigating the probability distribution of false negative failure alerts in HPC systems

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 its able to discover around 50\% of all failures in a system. However large part of failures are not predicted and considered as false negative alerts. Therefore, developing efficient fault tolerance strategies to tolerate failures requires a good perception and understanding of failure prediction properties and characteristics. In order to study and understand the properties and characteristics of the false negative alerts, we conduct in this paper a statistical analysis to discover the probability distribution of such alerts and their impact on fault tolerance techniques. To this end we study failures logs from different HPC production systems. We show that: (i) surprisingly 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 describes the inter arrival time between false negative alerts and so current fault tolerance can be applied on these systems; (iii) the current failures traces contain a high amount of correlation between the failure inter arrival time that can be used to improve the failure prediction mechanism. Another important result is that checkpoint intervals can still be computed from existing first order formula when failure distribution is purely random.

Rajeev Thakur,

Update on MPI and OS/R Activities at Argonne

This talk will give an update on MPI and OS/R activities at Argonne, including a big new project that is about to start in the area of exascale operating systems and runtime.

Andra Hugo

Composing multiple StarPU applications over heterogeneous machines: a supervised approach

Enabling HPC applications to perform efficiently when invoking multiple parallel libraries simultaneously is a great challenge. Even if a single runtime system is used underneath, scheduling tasks or threads coming from different libraries over the same set of hardware resources introduces many issues, such as resource oversubscription, undesirable cache flushes or memory bus contention. We present an extension of StarPU, a runtime system specifically designed for heterogeneous architectures, that allows multiple parallel codes to run concurrently with minimal interference. Such parallel codes run within scheduling contexts that provide confined execution environments which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries. We introduce a hypervisor that automatically expands or shrinks contexts using feedback from the runtime system (e.g. resource utilization). We demonstrate the relevance of our approach using benchmarks invoking multiple high performance linear algebra kernels simultaneously on top of heterogeneous multicore machines. We show that our mechanism can dramatically improve the overall application run time (-34%), most notably by reducing the average cache miss ratio (-50%).

Vincent Lanore

Static 2D FFT adaptation through a component model based on Charm++

Adaptation algorithms for HPC applications can improve performance but their implementation is often costly in terms of development and maintenance. Component models such as *Gluon++,* which is built on top of Charm++, propose to separate the business code, encapsulated in*components*, and the application structure, expressed through a *component assembly*. Adaptation of component-based HPC applications can be achieved through the optimization of the assembly. We have studied such an approach with the adaptation to network topology and data size of a gluon++ 2D FFT application. In this talk, we present our work thus far and comment preliminary experimental results on the Grid'5000 platform.

Stefan Wild

Loud computations? Noise in iterative solvers

Roundoff errors, discretizations, numerical solutions to systems of equations, and adaptive techniques can destroy the smoothness of processes underlying computations at scale. Such computational noise complicates optimization, sensitivity analysis, and other applications that depend on the simulation output. We present a method for analyzing computational noise and illustrate the insights it enables on a collection of problems based on Krylov solvers.

Guillaume Aupy

On the Combination of Silent Error Detection and Checkpointing

In this talk, we revisit traditional checkpointing and rollback recovery strategies, with a focus on silent data corruption errors. Contrarily to fail-stop failures, such latent errors cannot be detected immediately, and a mechanism to detect them must be provided. We consider two models: (i) errors are detected after some delays following a probability distribution (typically, an Exponential distribution); (ii) errors are detected through some verification mechanism. In both cases, we compute the optimal period in order to minimize the waste, i.e., the fraction of time where nodes do not perform useful computations. In practice, only a fixed number of checkpoints can be kept in memory, and the first model may lead to an irrecoverable failure. In this case, we compute the minimum period required for an acceptable risk. For the second model, there is no risk of irrecoverable failure, owing to the verification mechanism, but the corresponding overhead is included in the waste. Finally, both models are instantiated using realistic scenarios and application/architecture parameters.

Dries Kimpe

Triton: Exascale Storage

In this talk, I will present a status update of our work on Triton, a newly designed exascale era storage system. In addition to Triton specific information, the presentation will also include a brief discussion about the tools and techniques that help us in implementing and designing Triton. One such tool is the use of discrete event simulation to quickly evaluate algorithms at scale before implementing them in Triton.

Tatiana Martsinkevich

On the feasibility of message logging in hybrid hierarchical FT protocols

abstract: Hybrid hierarchical fault tolerance protocols are a promising solution for providing fault tolerance on large scale. A hybrid hierarchical protocol combines coordinated checkpointing and message logging. A lot of work has been done on more efficient implementation of checkpointing protocols, however there are some questions that stay not fully studied with regards to message logging. Message logging requires some portion of the process memory and logged data is flushed to a safe storage together with the checkpoint. There are several possible strategies to take in case where there is not enough memory to log messages between two checkpoints. Each of the strategies will be discussed.

Di Sheng

Optimization of Google Cloud Task Processing with Checkpoint-Restart Mechanism

In this paper, we aim at optimizing fault-tolerance techniques based on a checkpointing/restart mechanism, in the context of cloud computing. Our contribution is three-fold. (1) We derive a fresh formula to compute the optimal number of checkpoints for cloud jobs with varied distributions of failure events. Our analysis is not only generic with no assumption on failure probability distribution, but attractively simple to apply in practice. (2) We design an adaptive algorithm to optimize the checkpointing effect regarding various costs like checkpointing/restart overhead. (3) We evaluate our optimized solution in a real cluster environment with hundreds of virtual machines and Berkeley Lab Checkpoint/Restart tool. Task failure events are emulated via a production trace produced on a large-scale Google data center. Experiments confirm that our solution is fairly suitable for Google systems. Our optimized formula outperforms Young's formula by 3-10 percent, reducing wall-clock lengths by 50-200 seconds per job on average.

Yushan Wang

Accelerating incompressible fluid flows simulations using SIMD or GPU computing

We present a parallel solver for the 3-D Navier-Stokes (NS) equations of incompressible unsteady flows with constant coefficients, discretized by the finite difference method. We apply a prediction-projection method which transforms the Navier-Stokes equations into three Helmholtz equations and one Poisson equation. For each Helmholtz system, we apply the Alternating Direction Implicit (ADI) method resulting in three tridiagonal systems. The Poisson equation is solved using partial diagonalization which transforms the Laplacian operator into a tridiagonal one. In this talk we describe how we can take advantage of SIMD extensions in the solution of the resulting tridiagonal systems. We also present preliminary results for a GPU version of our NS solver.

Francois Tessier

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 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 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 for the two classes of parallel applications.

Guillaume Mercier

Topology Management and MPI Implementations Improvements

Modern hardware architectures featuring multicores and a complex memory hierarchy raise challenges that need to be addressed by parallel applications programmers. It is therefore tempting to adapt an application communication pattern to the characteristics of the underlying hardware. The MPI standard features several functions that allow the ranks of MPI processes to be reordered according to a graph attached to a newly created communicator. In this talk, we explain how the MPI implementation of the MPI_Dist_graph_create function was modified to reorder the MPI process ranks to create a match between the application communication pattern and the hardware topology. The experimental results on a multicore cluster show that improvements can be achieved as long as the application communication pattern is expressed by a relevant metric. We also show several areas in MPI implementations where similar techniques can be beneficial.

Eddy Caron

Seed4C: Secured Embedded Element and Data privacy for Cloud Federation

In this talk we introduce the design of a secure federated cloud from end to end. We discussed the core of this platform based on a High Performance Computing middleware that uses federated clouds and other virtual resources as well classic HPC resources. We propose an architecture to ensure a high level of security from personal devices to the targeted virtual machine. The Seed4C platform improved security in each layer. With DIET Cloud, we are able to deploy a large-scale, distributed and secure HPC platform that spans across a large pool of resources aggregated from different providers through a secure way

Jonathan Rouzaud-Cornabas

SimGrid Cloud Broker: Simulation of Public and Private Clouds

Before migrating an application to public Clouds, it is required to evaluate its performance. Doing so on a real Cloud has a time and money cost. By using simulation, it is possible to evaluate the migration without money cost and with a reduced time cost. Furthermore, it ables to test different resource reservation and application allocation algorithms without paying for these resources. The same is true when using private and/or hybrid Clouds. SimGrid Cloud Broker (SGCB) ables to easily simulate a whole Cloud public and/or private to evaluate an application. Moreover, due to high modularity, SGCB can also be used to evaluate the inner working of a Cloud middleware. Accordingly, it is possible to evaluate the impact of new VM to PM placement algorithms and virtual machine image deployment policies. To conclude, SGCB is a general purpose simulator for evaluating applications that run on public and private Clouds and their compositions.

Frederic Hecht

FreeFem++, a user language to solve PDE.

I will make a small overview of the capability of FreeFem++. and I will focus on four computer science problems:

-the design of the language:

--- from store mathematical formulation (the weak form of PDE) to Data Structure (DS),

--- from DS to matrix and right hand side.

-the way to use lots of third party software : like : MUMPS, IPOPT, TETGEN, MKL, ...

-the use of mesh adapted

-the parallelization with MPI.

Ian Foster

Compiler optimization for distributed dynamic data flow programs

Distributed, dynamic data flow is an execution model well-suited for many large-scale parallel applications, particularly scientific simulations and analysis pipelines running on large, distributed-memory clusters. In this paper we describe compiler optimization techniques and an intermediate representation for distributed dynamic data flow programs. These techniques are applied to Swift/T, a high-level declarative language that allows flexible data flow composition of functions written in other programming languages such a C or Fortran. We show that compiler optimization can reduce communication overhead by 70-93% on distributed memory systems, making the high-level language competitive with hand-coded coordination logic for certain common application styles

Christian Perez

On Component Models to Deploy Application on Clouds

Clouds have become a complex ecosystem, providing many kinds of virtual machines (with different capabilities), of usage (on demand, spot instances, reservation), of data storage, etc. Moreover, some clouds provides worldwide "regions", enabling large scale distributed applications. Users also have very different requirements, potentially from execution to another such as minimizing execution time, respecting budget constraints, etc. Therefore, automatically and efficiently deciding how to map an application to a set of VM is a difficult challenge. This talk will discuss how the European PaaSage project as well as the French ANR MapReduce are using component models to describe and map an application structure, independently of anycloud, to an actual cloud

Kate Keahey

Research Topics and Collaboration Opportunities in the Nimbus Team

The advent of IaaS cloud computing promises acquisition and management of customized on-demand resources. What is the best way to leverage those resources? What new applications are emerging in this context? How will they change our work patterns? What new technical approaches need to be developed to support them? What new opportunities will they lead to? In this talk, I will describe tools the Nimbus team is developing, among others, in the context of the Ocean Observatory Initiative project, that focus on answering these questions. I will describe our approach and tools, the problems we are trying to address, as well as the interaction patterns associated with scientific applications currently driving our approach.

Abdou Guermouche

Towards resilient parallel linear Krylov solvers

The advent of exascale machines will require the use of parallel resources at an unprecedented scale, probably leading to a high rate of hardware faults. High Performance Computing (HPC) applications that aim at exploiting all these resources will thus need to be resilient, i.e., being able to still compute a correct solution even in presence of faults. In this work, we investigate possible remedies in the framework of the solution of large sparse linear systems that is often the inner most numerical kernel in many scientific and engineering applications and also one of the most time consuming part. More precisely, we present recovery followed by restarting strategies in the framework of Krylov subspace solvers where lost entries of the iterate are interpolated to define a new initial guess before restarting. In particular, we consider two interpolation policies that preserve key numerical properties of well-known solvers. We assess the impact of the recovery method, the fault rate and the number of processors on the robustness of the resulting linear solvers. We consider experiments with CG, GMRES and Bi-CGStab.