Joint-lab workshop Jun. 13-15 2012

Restaurants name, place and time and workshop buses schedule.

This event is supported by INRIA, UIUC, NCSA, ANL, as well as by EDF

Main Topics	Schedule	Speaker	Affiliation	Type of presentation	Title (tentative)	Download
Dinner Before the Workshop	8:00 PM	Registered people only			RESTAURANT « CAFE DE LA PAIX » 1 place de la République - 3500 RENNES (Metro station « République ») Tél : 02 99 79 47 11	
Workshop Day 1	**Wednesday June 13th**					
					TITLES ARE TEMPORARY (except if in bold font)	
Registration	08:30					
Welcome and Introduction	09:00	Marc Snir + Franck Cappello	INRIA&UIUC	Background	**Welcome, Workshop objectives and organization**	Logistic_Workshop_schedule.pdf
	09:15	Bertrand Braunschweig	INRIA	Background	**Welcome to INRIA Rennes**	welcome petascale workshop braunschweig.pdf
	09:30	Thierry Priol	INRIA	Background		HPC_Inria-priol.pdf
Sustained Petascale Chair: Gabriel Antoniu	09:45	Bill Kramer	NCSA	Background	**Blue Waters UPDATE and performance metrics**	Kramer - BW_Update_Joint_Lab_June12.pdf
	10:15	Break				
	10:45	Romain Dolbeau	CAPS Entreprise	Background	**Programming Heterogeneous Many-cores Using Directives**	OpenACC-2012.pdf
	11:15	Marc Snir	ANL	Background	**BlueGene Q: First impression**	Rennes Snir.pdf
	11:45	Robert Ross	ANL	Background	**Big Data and Scientific Computing**	ross_jointlab-data-2012-06-13.pdf
	12:15	Lunch				
Mini Workshop1						
Fault tolerance Chair: Franck Cappello	13:30	Sanjay Kale and Marc Snir	UIUC, ANL	Background	**Fault tolerance needs at NCSA and ANL**	
	14:00	Ana Gainaru	NCSA	Joint Result	**A detailed analysis of fault prediction results and impact for HPC systems**	againaru_jointlab.pdf
	14:30	Amina Guermouche	INRIA	Joint Result	**Unified Model for Assessing Checkpointing Protocols at Extreme-Scale**	AminaGuermouche.pdf
	15:00	Break				
	15:30	Mehdi Diouri	INRIA	Joint Results	**Power and Energy consumption in Fault Tolerance protocols**	MehdiDiouri.pdf
	16:00	Tatiana Martsinkevich	INRIA	in progress	**On distributed recovery for SPMD deterministic HPC applications**	martsinkevich distributed recovey.pdf
	16:30	Sanjay Kale	UIUC	Result	**The recovery and rise of checkpoint/restart**	
	17:00	**Discussions**			**How to address Petascale fault tolerance needs**	
	18:00	Adjourn				
Mini Workshop2						
I/O and BigData Chair: Rob Ross	13:30	Bill Kramer and Rob Ross	UIUC, ANL	Background	**I/O and BIGDATA needs at NCSA and ANL**	
	14:00	Matthieu Dorier	INRIA	Joint Result	**In-Situ Interactive Visualization of HPC Simulations with Damaris**	DORIER-JLPC-June2012.pdf
	14:30	Francieli Zanon Boito	UFRGS /INRIA	Joint Result	**Investigating I/O approaches to improve performance and scalability of the Ocean-Land-Atmosphere Model**	Logistic_Workshop_schedule.pdf
	15:00	Break				
	15:30	Dries Kimpe	ANL	Background	**The Triton Data Model**	dkimpe-asg-model.pdf
	16:00	---	---	---		
	16:30	---	---	---		
	17:00	**Discussions**			**How to address Petascale I/O and Big Data needs**	
	18:00	Adjourn				
Workshop Day 2	**Thursday June 14th**					
Math for HPC Chair: Marc Snir	08:30	Frederic Vivien	INRIA	Joint Result	**Combining Process Replication and Checkpointing for Resilience on Exascale Systems**	2012-06-14_UIUC_FredericVivien.pdf
	09:00	Paul Hovland	ANL	Background	**Computational Foundations of Automatic Differentiation**	Hovland_Foundations2012.pdf
	09:30	Laurent Hascoet	INRIA	Joint Results	**Gradient of MPI-parallel codes**	Hascoet_slidesRennes.pdf
	10:00	Break				

Programming languages and performance modeling Chair:	10:30	Rajeev Thakur	ANL	Background	**Recent Activities in Programming Models and Runtime Systems at ANL**	Rajeev.pdf
	11:00	Sanjay Kale	UIUC	Background	**Charj: compiler supported language with an adaptive runtime**	
	11:30	Torsten Hoefler	NCSA	Background	**The Sustained Petascale Performance Applications on Blue Waters Performance Considerations and Modeling**	hoefler-prac-apps.pdf
	12:00	Lunch				
Mini Workshop3						
Numerical libraries Chair: Paul Hovland	13:30	Paul Hovland and Bill Gropp	UIUC, ANL	Background	Numerical libraries needs at NCSA and ANL	
	14:00	Laura Grigori	INRIA	Joint Result	**Hybrid static/dynamic scheduling for already optimized dense matrix factorization**	HybridSchedCA_IPDPS12-1.pdf
	14:30	François Pelegrini	INRIA	Joint Result	**Introducing PaMPA**	presentation-11.pdf
	15:00	Break				
	15:30	Jocelyne Erhel	INRIA	Background	**Solving linear systems arising from flow simulations in 3D Discrete Fracture Networks**	
	16:00	Daisuke Takahashi	U. Tsukuba	Joint Result	**An Implementation of Parallel 3-D FFT with 1.5-D Decomposition**	takahashi-1.pdf
	16h30	Adrien Remy	INRIA	Joint Result	**Solving general dense linear systems on hybrid multicore-GPU systems.**	Jointlab_REMY.pdf
	17:00	**Discussions**			**How to address Petascale Numerical Libraries needs**	
	17:55	Adjourn				
Mini Workshop4						
Programing Models Chair: Sanjay Kale	13:30	Rajeev Thakur and Sanjay Kale	UIUC, ANL	Background	Programming model needs at NCSA and ANL	
	14:00	Jean-François Mehaut	INRIA	Joint Result	**Load Balancing for Parallel Multi-core Machines with Non-Uniform Communication Costs**	francieliboito_jointlabworkshop.pdf
	14:30	Brice Goglin	INRIA	Background	**Bringing hardware affinity information into MPI communication strategies**	20120614-JLPC-MPIaffinities-1.pdf
	15:00	Break				
	15:30	Thomas Ropars	EPFL	Background	**Towards efficient collective operations on the Intel SCC**	broadcast_scc.pdf
	16:00	Alexandre Duchateau	INRIA	Joint Result	**Hydra : Generation and Tuning of Parallel Solutions for Linear Algebra**	Duchateau_Jointlab_June_2012.pdf
	16:30	**Discussions**			**How to address Petascale programing model needs**	
	17:30	Adjourn				
	18:00	Banquet			@ Saint Malo	
Workshop Day 3	**Friday June 15th**					
Mini Workshop5						
Mapping and Scheduling Chair: Torsten Hoefler	08:30	Blll Kramer and Marc Snir	UIUC, ANL	Background	Mapping and Scheduling needs at NCSA and ANL	BW JLPC-June2012-Breakout-Mapping and Scheduling.pdf
	09:00	François Teyssier	INRIA	Joint Result	**Load balacing and affinities between processes with TreeMatch in Charm++ : preliminary results and prospects**	
	09:30	Sébastien Fourestier	INRIA	Background	**Latest improvements to Scotch and ongoing collaborations**	
	10:00	Torsten Hoefler	NCSA --> ETH	Background	**On-node and off-node Topology Mapping for Petascale Computers**	
	10:30	Joseph Emeras, Olivier Richard, Cristian Ruiz	INRIA	Background	**Jobs Resource Utilization as a Metric for Clusters Comparison and Optimization**	
	11:00	**Discussions**			**How to address Petascale Mapping and Scheduling needs**	
Mini Workshop6						
HPC/Cloud Chair: Kate Keahey and Bogdan Nicolae	08:30	Kate Keahey (main speaker) and Franck Cappello	ANL, INRIA	Background	**HPC Cloud**	HPC Cloud ANL v1.pdf
	09:00	Gabriel Antoniu	INRIA	Joint Result	**A Performance Evaluation of Azure and Nimbus Clouds for Scientific Applications**	
	09:30	Frederic Desprez	INRIA	Background	**Budget Constrained Resource Allocation for Non-Deterministic Workflows on a IaaS Cloud**	
	10:00	Bogdan Nicolae	INRIA	Joint Results	**A Hybrid Local Storage Transfer Scheme for Live Migration of I/O Intensive Workloads**	
	10:30	Derrick Kondo	INRIA	Result	**Characterization and Prediction of Host Load in a Google Data Center**	
	11:00	**Discussions**			**How to address HPC Cloud needs**	
	12:00	Franck Cappello and Marc Snir			Discussion and Closing	
	12:30	Lunch				

Abstracts

Romain Dolbeau: Programming Heterogeneous Many-cores Using Directives

Pushed by the pace of innovation in the GPU and more generally the many-core technology, the processor landscape is moving at high-speed. This fast evolution makes software development more complex. Furthermore, the impact of the programming style on future performance and portability of the application is difficult to forecast. The use of directives to annotate serial languages (e.g. C/C++/Fortran) looks very promising. They abstract low-level parallelism implementation details while preserving code assets from the evolution of processor architectures. In this presentation, we describe how to use HMPP (Heterogeneous Many-core Parallel Programming) as well as OpenACC directives to program heterogeneous compute nodes. In particular, we provide insights on how GPU / CPU can be exploited in a unified manner and how code tuning issues can be minimized. We extend the discussion to the use of libraries that is currently one of the key elements when addressing GPU and many-cores.

Robert Ross: Big Data and Scientific Computing

Big Data is a hot area of research and development across many agencies and application domains, from Internet services to metagenomics. At the same time, it's not clear exactly what Big Data means, or how it relates to the traditional high-performance computing with which we are most familiar. This talk will present some introductory thoughts on this relationship and set the stage for ongoing discussions of Big Data during the workshop.

Paul Hovland: Computational Foundations of Automatic Differentiation

Automatic, or algorithmic, differentiation is a technique for transforming a program or subprogram that computes a mathematical function into one that computes the derivatives of that function. Successful implementation of automatic differentiation tools requires research and development across a broad spectrum of computer science, including graph theory, compilers, parallel algorithms, and numerical analysis. We describe some of the computational foundations of automatic differentiation, including graph-based heuristics for identifying and exploiting common subexpressions, parallel numerical algorithms, and domain-specific dataflow analysis problems. We demonstrate the importance of accurate derivatives to numerical algorithms. We explore in more detail the requirements of the so-called reverse, or adjoint, mode of automatic differentiation.

Laurent Hascoet: Gradient of MPI-parallel codes

Automatic Differentiation (AD) is the primary means of obtaining analytic derivatives from a numerical model given as a computer program. Therefore, it is an essential productivity tool in numerous computational science and engineering domains. Computing gradients with the adjoint mode of AD via source transformation is a particularly beneficial but also challenging use of AD. To date only ad hoc solutions for adjoint differentiation of MPI programs have been available, forcing AD users to reason about parallel communication dataflow and dependencies and manually develop adjoint communication code. In this collaboration between Argonne, RWTH Aachen, and INRIA, we characterize the principal problems of adjoining the most frequently used communication idioms. We propose solutions to cover these idioms and consider the consequences for the MPI implementation, the MPI user and MPI-aware program analysis.

Rajeev Thakur: Recent Activities in Programming Models and Runtime Systems at ANL

Future extreme scale systems will present new challenges in how to program them effectively, particularly in the areas of scalability, energy efficiency, resilience, and programmability. In this talk, I will describe our near-term plans to tackle these challenges. I will also give an update on our other recent activities in communication libraries and runtime systems for accelerators.

Laxmikant (Sanjay) Kale: Charj: compiler supported language with an adaptive runtime (http://charm.cs.uiuc.edu)
How to conquer the complexity of parallel programming is a topic that has been debated since the inception of the field. Various attempts at doing so, including parallelizing compilers, and other higher level languages such as HPF, have not succeeded. With scalable parallel computing subfield, some of the reasons for this failure have to do with the high premium placed on "performance" by practitioners, and the inability of the complex and sophisticated compiler analysis techniques to deliver performance and useful abstraction in a uniform manner. We argue that a relatively simple compilation support, based on well understood techniques in static analysis, can help improve the productivity substantially, if it is linked with a rich and sophisticated (and therefore complex) programming substrate provided by an adaptive runtime system. Specifically, I will present Charj, a compiler-supported language we are designing based on the Charm++ runtime system. I will illustrate the potential and realized productivity benefits using basic analysis techniques, code generation, and support for convenient syntax. The talk is largely based on the PhD thesis of Aaron Becker, who defended his dissertation last week.

Torsten Hoefler: The Sustained Petascale Performance Applications on Blue Waters Performance Considerations and Modeling

The sustained petascale performance of the Blue Waters system will be demonstrated using a suite of several applications representing a wide variety of disciplines important to the scientific community of the US National Science Foundation. The geometric mean of the measured floating point rates for these applications running scientific problems of current interest at scale is used to compute the sustained petascale performance (SPP), which is a key acceptance metric for the system. In this talk, we discuss the performance of these applications on Cray XE hardware. Our elemental modeling methodology splits each of the codes into a small set of performance-relevant kernels (typically around 5-7). We analyze those kernels in detail on the AMD Interlagos architecture to determine the achieved memory bandwidths and latencies, communication bandwidths and latencies, and floating point rates. This allows us to conclude, in a form similar to the Roofline model, how well each kernel performs with regards to each of the parameters. For example, if a kernel is mediocre in peak performance but utilizes 100% of the memory or network bandwidth, we can conclude that the kernel is utilizing the architecture well and is memory- (or communication-) bound. While a low floating point rate and a low bandwidth utilization suggests optimization opportunities. Our analyses should also provide us with insight into application performance on future systems.

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

Frederic Vivien: Combining Process Replication and Checkpointing for Resilience on Exascale Systems

Processor failures in post-petascale settings are common occurrences. The traditional fault-tolerance solution, checkpoint-rollback, severely limits parallel efficiency. One solution is to replicate application processes so that
a processor failure does not necessarily imply an application failure. Process replication, combined with checkpoint-rollback, has been recently advocated by Ferreira et al. We first identify an incorrect analogy made in their work between process replication and the birthday problem, and derive correct values for the Mean Number of Failures To Interruption and Mean Time To Interruption for Exponential failures distributions. We then extend these results to arbitrary failure distributions, including closed-form solutions for Weibull distributions. Finally, we evaluate process replication using both synthetic and real-world failure traces. Our main findings are: (i)~replication is beneficial in fewer scenarios than claimed by Ferreira et al; (ii)~although the choice of the checkpointing period can have a high impact on application execution in the no-replication case, with process replication this choice is no longer critical.

Ana Gainaru: A detailed analysis of fault prediction results and impact for HPC systems

A large percentage of computing capacity in today's large high-performance computing systems is wasted due to failures and recoveries. As a consequence current research is focusing on providing fault tolerance strategies that aim to minimize fault's effects on applications. By far, the most popular and used technique from this field is the checkpoint-restart strategy. A complement to this classical approach of handling errors that cause application crashes in large-scale clusters is failure avoidance, by which the occurrence of a fault is predicted and preventive measures are taken. For this, monitoring systems require a reliable prediction system to give information on what will be generated by the system and at what location. Thus far, research in this field used an ideal predictor that so far did not have any implementation in real HPC systems. In this talk, we present a new method for predicting faults by merging signal analysis concepts with data mining techniques. A large part of this talk is focused on a detailed analysis of the prediction method, by applying it to two large-scale systems and by investigating the characteristics and bottlenecks of each step of the prediction process. Furthermore, we analyze the prediction's precision and recall impact on current checkpointing strategies.

Amina Guermouche: Unified Model for Assessing Checkpointing Protocols at Extreme-Scale

In this talk, we present a unified model for several well-known checkpoint/restart protocols. The proposed model is generic enough to encompass both extremes of the checkpoint/restart space: on one side the coordinated
checkpoint, and on the other extreme, a variety of uncoordinated checkpoint strategies (with message logging). We identify a set of parameters that are crucial to instantiate and compare the expected efficiency of the fault tolerant protocols, for a given application/platform pair. We then propose a detailed analysis of several scenarios, including some of the most powerful currently available HPC platforms, as well as anticipated Exascale designs. This comparison outlines the comparative behaviors of checkpoint strategies at scale, thereby providing insight that is hardly accessible to direct experimentation

Mohammed el Mehdi Diouri: Power and Energy consumption in Fault Tolerance protocols

Exascale supercomputers will gather hundreds millions cores. The first problem that we address is resiliency and fault tolerance to reach application termination on such platforms. The second problem is energy consumption since such systems will consume enormous amount of energy. In our work, we evaluate checkpointing and existing fault tolerance protocols from an energy point of view. We measure on a real testbed the power consumption of the main atomic operations found in these protocols. The first results show that process coordination and RAM consume more power than checkpointing and HDD logging. However, the results we presented in Joules per Bytes for I/O operations, emphasize that checkpointing and HDD logging consume more energy than RAM logging. Finally, we propose to consider energy consumption as a criterion for the choice of fault tolerance protocols. In terms of energy consumption, we should promote message logging for applications exchanging small volumes of data and coordination for applications involving few processes. This preliminary work led us to propose a framework that estimates energy consumption of fault tolerance protocols during HPC executions. Energy estimations are then used to select the best fault tolerance protocol from in terms of energy consumption.

Tatiana Martsinkevich: On distributed recovery for SPMD deterministic HPC applications

Hybrid fault tolerance protocols for HPC applications are the most likely protocols to find their niche in the umcoming exascale era. HydEE, a hierarchical rollback-recovery protocol for message passing applications, is an example of such protocol. Current scheme in HydEE requires a dedicated process to orchestrate the recovery. This centralized solution can slow down the recovery which otherwise could be sped up by the fact that all the messages that process needs to recover to the point of failure are logged and could be accessed immediately. If we find a way for each process to control its recovery by itself, we could fully benefit from message logging. This talk is about finding an approach to implement such distributed recovery.

Laxmikant (Sanjay) Kale: The recovery and rise of checkpoint/restart (http://charm.cs.uiuc.edu)

Checkpoint/restart has often be considered as a protocol that will not scale to large machines. Earlier, the claim was made for petascale machines, and later for exascale machines. But the idea has proved to be more resilient than the claims. Within the HPC community in the US, and especially the DOE, multiple researchers have started defending the scalability of checkpoint/rstart schemes. In this talk, I will show some results in scalable checkpoint/restart, and examine reasons for its scalability. I will describe my viewpoint on where different categories of protocols, such as message-logging, causal, and coordinated checkpoint/restart will work better, and what are the likely scenarios at exascale. The talk is aimed at spurring controversy and discussion. Although we will touch upon soft faults, the main focus of this talk will be fail-stop faults.

L. Pilla (UFRGS/INRIA Grenoble), J-F. Méhaut (UJF/CEA): Load Balancing for Parallel Multi-core Machines with Non-Uniform Communication Costs

Multi-core machines with NUMA design are now common in the assembly of HPC machines. On these machines, in addition to load imbalance coming from the dynamic application, the asymmetry of memory access and network communications costs plays a major role in obtaining high efficiency. Taking both of these criteria into account is a key step to achieve portability of performance on current HPC platforms. In this talk we will explain our portable approach to increase thread/data affinity while reducing core idleness on both sharedmemory and distributed parallel platforms. We will also present the way we implemented it as a Charm++ load balancer that relies on a generic view of the machine topology decorated with benchmarked communication costs. We will end the presentation by showing some of its performance improvements over other state-of-the-art load balancing algorithms on different parallel machines.

Matthieu Dorier: In-Situ Interactive Visualization of HPC Simulations with Damaris

The I/O bottlenecks already present on current petascale systems force to consider new approaches to get insights from running simulations. Trying to bypass storage or drastically reducing the amount of data generated will be of outmost importance for the scales to come and, in particular, for Blue Waters.

This presentation will focus on the specific case of in-situ data analysis collocated with the simulation's code and running on the same resources. We will first present some common visualization and analysis tools, and show the limitations of their in-situ capabilities. We then present how we enriched the Damaris I/O middleware to support analysis and visualization operations. We show that the use of Damaris on top of existing visualization packages allows us to (1) reduce code instrumentation to a minimum in existing simulations, (2) gather the capabilities of several visualization tools to offer adaptability under a unified data management interface, (3) use dedicated cores to hide the run time impact of in-situ visualization and (4) efficiently use memory through an allocation-based communication model.

Francieli Zanon Boito: Investigating I/O approaches to improve performance and scalability of the Ocean-Land-Atmosphere Model

Many HPC applications manipulate large amount of data, with all its processes generating I/O operations. This situation, where a large number of processes do I/O operations, can seriously impair the application's performance, since it causes resource contention and variability in I/O performance. Decreasing the number of nodes involved in I/O, in an application, can reduce such contention and improve overall performance of the code. This can be achieved by coding the application in special modes so that only one process in a node communicates with the file system or by using special APIs to manage I/O. Damaris is a tool that provides such functionality, dedicating cores of a SMP node to perform the I/O operation on behalf of the others cores. In this talk we will present our ongoing work with Damaris to improve the performance of the Ocean-Land-Atmosphere Model (OLAM). OLAM has serious scalability problems that are shown to be caused by its I/O phases, done by all its processes. We will present experimental results showing two I/O-node reduction strategies (Damaris and MPI+OpenMP) and their effects at OLAM's execution time and discuss future directions.

Dries Kimpe: The Triton Data Model

This presentation will present a survey of existing data models provided by contemporary parallel file systems and I/O libraries, followed by an introduction to Triton, Argonne's new storage system, and the revolutionary new data model it provides. Through a set of examples, we show how the new data model can be used as a building block for supporting legacy models, and how its new features can be exploited to optimize distributed I/O patterns.

Bogdan Nicolae: A Hybrid Local Storage Transfer Scheme for Live Migration of I/O Intensive Workloads

Live migration of virtual machines (VMs) is key feature of virtualization that is extensively leveraged in IaaS cloud environments: it is the basic building block of several important features, such as load balancing, pro-active fault tolerance, power management, online maintenance, etc. While most live migration efforts concentrate on how to transfer the memory from source to destination during the migration process, comparatively little attention has been devoted to the transfer of storage. This problem is gaining increasing importance: due to performance reasons, virtual machines that run large-scale, data-intensive HPC applications tend to rely on local storage, which poses a difficult challenge on live migration: it needs to handle storage transfer in addition to memory transfer. This paper proposes a memory-migration independent approach that addresses this challenge. It relies on a hybrid active push / prioritized prefetch strategy, which makes it highly resilient to rapid changes of disk state exhibited by I/O intensive workloads. At the same time, it is minimally intrusive in order to ensure a maximum of portability with a wide range of hypervisors. Large scale experiments that involve multiple simultaneous migrations of both synthetic benchmarks and a real scientific application show improvements of up to 10x faster migration time, 10x less bandwidth consumption and 8x less performance degradation over state-of-art.

François Pellegrini, Cédric Lachat: Introducing PaMPA

PaMPA ("Parallel Mesh Partitioning and Adaptation") is a middleware for the parallel remeshing and the redistribution of distributed unstructured meshes. PaMPA is meant to serve as a basis for the development of numerical solvers implementing compact schemes. PaMPA represents meshes as a set of interconnected entities (elements, faces, edges, nodes, etc.). Since the underlying structure is a graph, elements can be of any kind, and several types of elements can be used within the same mesh. Typed values (scalars, vectors, structured types) can be associated with entities. Value exchange routines allow users to copy values across neighboring processors, and to specify the width of the overlap across processors. Accessors and iterators allow developers of numerical solvers to write their numerical schemes without having to take into account mesh and value distributions. Parallel mesh partitioning and redistribution is now available, partly based on PT-Scotch. Parallel remeshing will soon be available. It will be handled by calling in parallel a user-provided sequential remesher on non-overlapping pieces of the mesh. A full-featured tetrahedron example will be provided before the end of this year, based on the MMG3D sequential remeshing software also developed at Inria.

Jocelyne Erhel: Solving linear systems arising from flow simulations in 3D Discrete Fracture Networks

Underground water is naturally channelled in fractured media, where interconnections can be very intricate. Discrete Fracture Networks are based on a geometrical model, where fractures are 2D domains, for example ellipses, which form a 3D network. We have developed an original numerical model in order to simulate flow in a randomly generated DFN. The spatial discretization leads to a symmetric positive definite linear system, with a large sparse matrix. We have investigated the efficiency of several linear solvers on parallel and distributed computers. Since the network can be easily partitioned into subdomains (the fractures), we have developed a hybrid solver based on domain decomposition. This approach decouples in some sense the flow at the fracture scale from the interactions at the network scale. The Schur complement, which gathers the unknowns at the intersections of the network, is solved with a preconditioned conjugate gradient. We combine Neumann Neumann, defined at the fracture scale, with a global preconditioner, defined at the network scale, based on a deflation formulation. The numerical model and the solver are implemented in the software MPFRAC, which is embedded into the scientific platform H2OLab. Our numerical experiments highlight the efficiency of this Schur solver.

Torsten Hoefler: On-node and off-node Topology Mapping for Petascale Computers

Network topology and the efficient mapping from tasks to physical processing elements is an increasing problem as we march towards larger systems. The last generation of Petascale class systems which comes right before a major switch to optical interconnects is highly affected due to their large low-bisection Torus networks. We will explore opportunities to improve communication performance by avoiding network congestion with automated task remapping. We discuss how we combine different approaches, various well-known heuristics, a heuristic based in RCM, INRIA's SCOTCH, and INRIA's tree-map algorithms for achieving highest mapping performance for on-node as well as off-node mappings. We also investigate a theoretical possibility to reduce energy usage due to minimizing dilation of the mapping. This whole work is done in the context of MPI and can readily be adapted to real production applications.

Adrien Remy: Solving general dense linear systems on hybrid multicore-GPU systems.

We highlight different algorithms for solving general dense linear systems using LU decomposition on multicore systems accelerated with multiple GPUs. We present a set of techniques to achieve high efficiency on these architectures and we compare the resulting algorithms in terms of performance, speed-up and accuracy. Finally we present some ongoing work with the goal to move to a larger scale using clusters of GPUs.

Derrick Kondo: Characterization and Prediction of Host Load in a Google Data Center
[Joint work with Sheng Di, Walfredo Cirne]

Characterization and prediction of system load is essential for optimizing its performance. We characterize and predict host load in a real-world production Google data center, using a detailed trace of over 25 million tasks across over 12,500 hosts. In our characterization, we present valuable statistics and distributions of machine's maximum load, queue state and relative usage levels. Compared to traditional load from Grids and other HPC systems, we ?nd that Google host load exhibits higher variance due to the high frequency of small tasks. Based on this characterization, we develop and evaluate different methods of machine load prediction using techniques such as autoregression, moving averages, probabilistic methods, and noise ?lters. We ?nd that a linear weighted moving average produces accurate predictions with a 80%-95% success rate, outperforming other methods by 5%-20%. Surprisingly, this method outperforms more sophisticated hybrid prediction methods, which are effective for traditional Grid loads but not data center loads due to its more frequent and severe ?uctuations.

Brice Goglin: Bringing hardware affinity information into MPI communication strategies

Understanding the hardware topology and adapting the software accordingly is increasingly difficult. Resource numbering is not portable across machines or operating systems. There are many levels of memory hierarchy. And the access to I/O and memory resource is not flat anymore. We will summarize the work that we put in the Hardware Locality software to provide applications with a portable and easy-to-use abstraction of hardware details. This deep knowledge of hardware affinities let us optimize MPI communication strategies within nodes or between nodes, for both point-to-point and collective operations. We now look at adding quantitative information to the existing qualitative hierarchy description to improve our locality-based criterias.

Thomas Ropars: Towards efficient collective operations on the Intel SCC

Abstract: Many-core chips with more than 1000 cores are expected by the end of the decade. To overcome scalability issues related to cache coherence at such a scale, one of the main research directions is to leverage the message-passing programming model. A many-core chip, such as the Intel Single-Chip Cloud Computer (SCC), integrates a large number of cores connected using a powerful Network-on-Chip. The SCC offers the ability to move data between on-chip Message Passing Buffers using Remote Memory Access (RMA). In this talk, we study how to provide efficient collective operations on the SCC, focusing on the broadcast primitive. We show how RMA operations can be leveraged to dramatically improve the communication performance compared to a solution based on a higher level send/receive interface.

Alexandre Duchateau: Hydra : Generation and Tuning of Parallel Solutions for Linear Algebra.

An auto-tuning system and methodology for algorithm exploration for a class of linear algebra problems.
Starting with a description of equations, the system automatically finds divide and conquer algorithms to solve the equations with the main objective of exposing parallelism. Parallelism is exposed in the form of parallel task graphs that are translated into the StarPU runtime framework for execution on multicore and in the future, on heterogeneous CPU-GPU systems.

Daisuke Takahashi: An Implementation of Parallel 3-D FFT with 1.5-D Decomposition

In this talk, an implementation of a parallel 3-D fast Fourier Transform (FFT) with 1.5-D decomposition is presented. A typical decomposition for performing a parallel 3-D FFT is slab-wise. In this case, for N^3-point FFT, N must be greater than or equal to the number of MPI processes. Our proposed parallel 3-D FFT algorithm with 1.5-D decomposition allows up to N^(3/2) MPI processes for N^3-point FFT. Moreover, this scheme requires only one all-to-all communication for transposed-order output. Performance results of parallel 3-D FFTs on clusters of multi-core processors are reported.

François Tessier: Load balacing and affinities between processes with TreeMatch in Charm++ : preliminary results and prospects

TreeMatch is an algorithm and a tool to perform process placement based on affinities between processes and NUMA topology. We have used this algorithm to design several dynamic load balancers in Charm+. We will present these implementations, the results it provided and the limits. Finally, we will explain the future works we plan to do with the Charm+ team, particularly about a distributed implementation of this affinity-aware load balancer.

Sébastien Fourestier: Latest improvements to Scotch and ongoing collaborations

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

Cristian Ruiz: Jobs Resource Utilization as a Metric for Clusters Comparison and Optimization

In HPC community the System Utilization metric enables to determine if the resources of the cluster are efficiently used by the batch scheduler. This metric considers that all the allocated resources (memory, disk, processors, etc) are full-time utilized. To optimize the system performance, we have to consider the effective physical consumption by jobs regarding the resource allocations. This information gives an insight into whether the cluster resources are optimally used by the jobs.
In this work we propose a new method for the comparison of production clusters based on the Jobs Resource Utilization criteria. The principle is to collect simultaneously traces from the job scheduler (provided by logs) and jobs resource consumptions. The latter has been realized by developing a job monitoring tool, whose impact on the system has been measured as lightweight (0.35% speed-down).
The key point is to statistically analyze both traces to detect and explain underutilization of the resources. This could enable to detect abnormal behavior, bottlenecks in the cluster leading to a poor scalability, and justifying optimizations such as gang scheduling or besteffort scheduling.
This method has been applied to two medium sized production clusters on a period of about a year.

Kate Keahey: HPC Cloud

Infrastructure clouds created ideal conditions for users to outsource their infrastructure needs beyond the boundaries of their institution. A typical infrastructure cloud offers (1) on-demand, short-term access, which allows users to flexibly manage peaks in demand, (2) pay-as-you-go model, which helps save costs for bursty usage patterns (i.e., helps manage "valleys" in demand), (3) access via virtualization technologies which provides a safe and cost-effective way for users to manage and customize their own environments, and (4) sheer convenience, as users and institutions no longer have to have specialized IT departments and can focus on their core mission instead. The flexibility of this approach allows users to also outsource as much or as little of their infrastructure procurement as their needs justify: they can keep a resident amount of infrastructure in-house while outsourcing only at times of increased demand, and they can outsource to a variety of providers choosing the best service levels for the price the market has to offer.

The availability of cloud computing gave rise to an interesting debate on its relationship to high performance computing (HPC). Two significant questions emerged in this context: (1) Can supercomputing workloads be run on a cloud? and (2) Can a supercomputer operate as a cloud? Much investigation has been done on the first issue, most notably and conclusively as part of the Magellan project. The second question, which could provide an interesting solution to challenges defined by the first, has not been investigated nearly as much. This talk will present a state-of-art summary of work in this space, discuss the current open challenges, propose relevant solutions in the area of resource management, as well as outline potential future directions and collaborations.

Gabriel Antoniu: A Performance Evaluation of Azure and Nimbus Clouds for Scientific Applications

As more and more cloud providers and technologies appear, scientists are faced with an increasingly difficult problem of evaluating various options, like public and private clouds, and deciding which model to use for their applications. In this talk, we make a performance evaluation of two public and private cloud platforms for scientific computing workloads. We make evaluations on Azure and Nimbus clouds, considering all primary needs of scientific applications (computation power, storage, data transfers and costs). The evaluation is done using both synthetic benchmarks and a real-life application. Our results show that Nimbus incurs less varaibility and has increased support for data intensive applications, whereas Azure deploys faster and may potentially have a lower cost.

Frédéric Desprez: Budget Constrained Resource Allocation for Non-Deterministic Workflows on a IaaS Cloud
Eddy Caron, Adrian Muresan and Frédéric Suter

Many scientific applications can be described through workflows due to their inherent structure or to the fact that they are built by aggregating other smaller applications and adding flow connections between them. Modern computational architectures are evolving in the direction of parallel computing through multi-core, the generalization of GPUs and compute clusters. This leads to applications being composed of not only sequential tasks, but also parallel ones which leads to more efficient ways of exploiting modern hardware. In the case of workflow applications, a task is called moldable if it can be run on a variable number of resources chosen at scheduling time and the workflow itself is called a Parallel Task Graph (PTG). Although most workflow applications can be fully described by a PTG, there are some that can only be described by a PTG that is augmented with special semantics for exclusive diverging control flows or repetitive flows. The resulting PTG is one that can have non-deterministic transitions and repetitive constructs.
The proliferation of Cloud Computing has led to a new way of managing resources: on-demand as opposed to static allocations. PTG applications can take advantage of this, but the classic approached to resource allocation and scheduling need to be adapted for on-demand resource provisioning. As a starting point, the execution of an application now corresponds to a budget, that translates to a virtual platform size.
The current work presents an approach for determining budget-constrained resource allocations for non-deterministic PTG workflows, on IaaS platforms. We solve this problem by decomposing the initial workflow into a set of deterministic sub-workflows, for which we find allocations by adapting a well-known existing allocation algorith