Activity Report 2016

Team DATAMOVE

Data Aware Large Scale Computing

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Distributed and High Performance Computing

Table of contents

Team DATAMOVE

Creation of the Team: 2016 January 01

The DataMove team is localized in the Imag building on the Campus of Univ. Grenoble Alpes.

Keywords:

Computer Science and Digital Science:

1.1.4. - High performance computing
1.1.5. - Exascale
2.1.10. - Domain-specific languages
2.6.2. - Middleware
7.1. - Parallel and distributed algorithms

Other Research Topics and Application Domains:

1.1.2. - Molecular biology
5.5. - Materials

1. Members

Research Scientist

Bruno Raffin [Team leader, Inria, Senior Researcher, Research Scientist, HDR]

Faculty Members

Yves Denneulin [Grenoble-INP, Faculty Member, Professor]
Pierre Francois Dutot [Univ. Grenoble-Alpes, Faculty Member, Associate Professor]
Gregory Mounie [Grenoble-INP, Faculty Member, Associate Professor]
Olivier Richard [Univ. Grenoble-Alpes, Faculty Member, Associate Professor]
Denis Trystram [Grenoble-INP, Faculty Member, Professor, HDR]
Frederic Wagner [Grenoble-INP, Faculty Member, Associate Professor]

Engineers

Romain Cavagna [Univ. Grenoble-Alpes, until Aug 2016]
Ivan Cores Gonzalez [Inria]
Tristan Ezequel [Inria]
Nicolas Michon [Inria, from Oct 2016]
Pierre Neyron [CNRS]
Baptiste Pichot [Inria]
Theophile Terraz [Inria]

PhD Students

Marcos Amaris Gonzalez [USP, until Oct 2016]
Raphaël Bleuse [Univ. Grenoble-Alpes]
Estelle Dirand [CEA]
Mohammed Khatiri [partly Inria]
Alessandro Kraemer [Federal Technological University of Paraná]
Fernando Machado Mendonca [USP]
Michael Mercier [ATOS/BULL, granted by CIFRE]
Millian Poquet [Univ. Grenoble-Alpes]
Valentin Reis [Grenoble-Alpes]
Marwa Sridi [CEA, until Apr 2016]
Abhinav Srivastav [Univ. Grenoble-Alpes, until Jul 2016]

Julio Toss [UFRGS]
David Glesser [ATOS/BULL, granted by CIFRE,until Oct 2016]

Post-Doctoral Fellow

Giorgio Lucarelli [Inria, until Oct 2017]

Visiting Scientists

Daniel Cordeiro [USP, until Mar 2016]
Sirine Marakchi [ISBS SFAX, Jan 2016]
Wafa Nafti [ESSTT, until May 2016]
Ioannis Milis [Athens UEB, June 2016]
Katrin Scharnowski [University of Stuttgart, from May 2016 until Aug 2016]

Administrative Assistant

Annie Simon [Inria]

Others

Bruno Bzeznik [Univ. Grenoble-Alpes]
Christian Seguy [CNRS]
Luis Omar Alvarez Mures [Inria, Student, from March 2016 until Mai 2016]
Clement Mommessin [Inria, Student, from Feb 2016 until Dec 2016]
Waqas Imtiaz [Inria,Student, from Feb 2016 until Jun 2016]
Matthias Kohl [Inria, Student, from Feb 2016 until Jun 2016]
Thomas Lavocat [Inria, Student, from Feb 2016 until Aug 2016]
Piat Wegener [Inria, Student, from May 2016 until Aug 2016]
Mohamed Dyab [Inria, Student, from Feb 2016 until Sep 2016]
Jordan Ellapin [Inria, Student, from May 2016 until Aug 2016]
Lucas Barallon [Inria, Student, from Feb 2016 until Aug 2016]
Jad Darrous [Inria, Student, from Feb 2016 until June 2016]

2. Overall Objectives

2.1. Overall Objectives

Moving data on large supercomputers is becoming a major performance bottleneck, and the situation is expected to worsen even more at exascale and beyond. Data transfer capabilities are growing at a slower rate than processing power ones. The profusion of flops available will be difficult to use efficiently due to constrained communication capabilities. Moving data is also an important source of power consumption. The DataMove team focuses on **data aware large scale computing**, investigating approaches to reduce data movements on large scale HPC machines. We will investigate data aware scheduling algorithms for job management systems. The growing cost of data movements requires adapted scheduling policies able to take into account the influence of intra-application communications, IOs as well as contention caused by data traffic generated by other concurrent applications. At the same time experimenting new scheduling policies on real platforms is unfeasible. Simulation tools are required to probe novel scheduling policies. Our goal is to investigate how to extract information from actual compute centers traces in order to replay job allocations and executions with new scheduling policies. Schedulers need information about the jobs behavior on the target machine to actually make efficient allocation decisions. We will research approaches relying on learning techniques applied to execution traces to extract data and forecast job behaviors. In addition to traditional computation intensive numerical simulations, HPC platforms also need to execute more and more often data intensive processing tasks like data analysis. In particular, the ever growing amount of data generated by numerical simulation calls for a tighter integration between the simulation and the data analysis. The goal is to reduce the data traffic and to speed-up result analysis by processing results in situ, i.e. as closely as possible to the locus and time of data generation. Our goal is here to investigate how to program and schedule such analysis workflows in the HPC context, requiring the development of adapted resource sharing strategies, data structures and parallel analytics schemes. To tackle these issues, we will intertwine theoretical research and practical developments to elaborate solutions generic and effective enough to be of practical interest. Algorithms with performance guarantees will be designed and experimented on large scale platforms with realistic usage scenarios developed with partner scientists or based on logs of the biggest available computing platforms. Conversely, our strong experimental expertise will enable to feed theoretical models with sound hypotheses, to twist proven algorithms with practical heuristics that could be further retro-feeded into adequate theoretical models.

3. Research Program

3.1. Motivation

Today's largest supercomputers [1] are composed of few millions of cores, with performances almost reaching 100 PetaFlops [2] for the largest machine. Moving data in such large supercomputers is becoming a major performance bottleneck, and the situation is expected to worsen even more at exascale and beyond. The data transfer capabilities are growing at a slower rate than processing power ones. The profusion of available flops will very likely be underused due to constrained communication capabilities. It is commonly admitted that data movements account for 50% to 70% of the global power consumption [3]. Thus, data movements are potentially one of the most important source of savings for enabling supercomputers to stay in the commonly adopted energy barrier of 20 MegaWatts. In the mid to long term, non volatile memory (NVRAM) is expected to deeply change the machine I/Os. Data distribution will shift from disk arrays with an access time often considered as uniform, towards permanent storage capabilities at each node of the machine, making data locality an even more prevalent paradigm.

The DataMove team works on **optimizing data movements for large scale computing** mainly at two related levels:

- Resource allocation
- Integration of numerical simulation and data analysis

The resource and job management system (also called batch scheduler or RJMS) is in charge of allocating resources upon user requests for executing their parallel applications. The growing cost of data movements requires adapted scheduling policies able to take into account the influence of intra-application communications, I/Os as well as contention caused by data traffic generated by other concurrent applications. Modelling the application behavior to anticipate its actual resource usage on such architecture is known to be challenging, but it becomes critical for improving performances (execution time, energy, or any other relevant objective). The job management system also needs to handle new types of workloads: high performance platforms now need to execute more and more often data intensive processing tasks like data analysis in addition to traditional computation intensive numerical simulations. In particular, the ever growing amount of data generated by numerical simulation calls for a tighter integration between the simulation and the data analysis. The challenge here is to reduce data traffic and to speed-up result analysis by performing result processing (compression, indexation, analysis, visualization, etc.) as closely as possible to the locus and time of data generation. This emerging trend called *in situ analytics* requires to revisit the traditional workflow (loop of batch processing followed by postmortem analysis). The application becomes a whole including the simulation, in situ processing and I/Os. This motivates the development of new well-adapted resource sharing strategies, data structures and parallel analytics schemes to efficiently interleave the different components of the application and globally improve the performance.

[1] Top500 Ranking, http://www.top500.org

[2] 10^{15} floating point operations per second

[3] SciDAC Review, 2010, http://www.scidacreview.org/1001/pdf/hardware.pdf

3.2. Strategy

DataMove targets HPC (High Performance Computing) at Exascale. But such machines and the associated applications are expected to be available only in 5 to 10 years. Meanwhile, we expect to see a growing number of petaflop machines to answer the needs for advanced numerical simulations. A sustainable exploitation of these petaflop machines is a real and hard challenge that we address. We may also see in the coming years a convergence between HPC and Big Data, HPC platforms becoming more elastic and supporting Big Data jobs, or HPC applications being more commonly executed on cloud like architectures. This is the second top objective of the 2015 US Strategic Computing Initiative [4]: *Increasing coherence between the technology base used for modelling and simulation and that used for data analytic computing*. We contribute to that convergence at our level, considering more dynamic and versatile target platforms and types of workloads.

Our approaches should entail minimal modifications on the code of numerical simulations. Often large scale numerical simulations are complex domain specific codes with a long life span. We assume these codes as being sufficiently optimized. We influence the behavior of numerical simulations through resource allocation at the job management system level or when interleaving them with analytics code.

To tackle these issues, we propose to intertwine theoretical research and practical developments in an agile mode. Algorithms with performance guarantees are designed and experimented on large scale platforms with realistic usage scenarios developed with partner scientists or based on logs of the biggest available computing platforms (national supercomputers like Curie, or the BlueWaters machine accessible through our collaboration with Argonne National Lab). Conversely, a strong experimental expertise enables to feed theoretical models with sound hypotheses, to twist proven algorithms with practical heuristics that could be further retro-feeded into adequate theoretical models.

A central scientific question is to make the relevant choices for optimizing performance (in a broad sense) in a reasonable time. HPC architectures and applications are increasingly complex systems (heterogeneity, dynamicity, uncertainties), which leads to consider the **optimization of resource allocation based on multiple objectives**, often contradictory (like energy and run-time for instance). Focusing on the optimization of one particular objective usually leads to worsen the others. The historical positioning of some members of the team who are specialists in multi-objective optimization is to generate a (limited) set of trade-off configurations, called *Pareto points*, and choose when required the most suitable trade-off between all the objectives. This methodology differs from the classical approaches, which simplify the problem into a single objective one (focus on a particular objective, combining the various objectives or agglomerate them). The real challenge is thus to combine algorithmic techniques to account for this diversity while guaranteeing a target efficiency for all the various objectives.

The DataMove team aims to elaborate generic and effective solutions of practical interest. We make our new algorithms accessible through the team flagship software tools, **the OAR batch scheduler and the in situ processing framework FlowVR**. We maintain and enforce strong links with teams closely connected with large architecture design and operation, as well as scientists of other disciplines, in particular computational biologists, with whom we elaborate and validate new usage scenarios.

3.3. Research Directions

DataMove targets HPC (High Performance Computing) at Exascale. But such machines and the associated applications are expected to be available only in 5 to 10 years. Meanwhile, we expect to see a growing number of petaflop machines to answer the needs for advanced numerical simulations. A sustainable exploitation of these petaflop machines is a real and hard challenge that we address. We may also see in the coming years a convergence between HPC and Big Data, HPC platforms becoming more elastic and supporting Big Data jobs, or HPC applications being more commonly executed on cloud like architectures. This is the second top objective of the 2015 US Strategic Computing Initiative [5]: *Increasing coherence between the technology*

[4] https://www.whitehouse.gov/the-press-office/2015/07/29/executive-order-creating-national-strategic-computing-initiative

[5] https://www.whitehouse.gov/the-press-office/2015/07/29/executive-order-creating-national-strategic-computing-initiative

base used for modelling and simulation and that used for data analytic computing. We contribute to that convergence at our level, considering more dynamic and versatile target platforms and types of workloads.

Our approaches should entail minimal modifications on the code of numerical simulations. Often large scale numerical simulations are complex domain specific codes with a long life span. We assume these codes as being sufficiently optimized. We influence the behavior of numerical simulations through resource allocation at the job management system level or when interleaving them with analytics code.

To tackle these issues, we propose to intertwine theoretical research and practical developments in an agile mode. Algorithms with performance guarantees are designed and experimented on large scale platforms with realistic usage scenarios developed with partner scientists or based on logs of the biggest available computing platforms (national supercomputers like Curie, or the BlueWaters machine accessible through our collaboration with Argonne National Lab). Conversely, a strong experimental expertise enables to feed theoretical models with sound hypotheses, to twist proven algorithms with practical heuristics that could be further retro-feeded into adequate theoretical models.

A central scientific question is to make the relevant choices for optimizing performance (in a broad sense) in a reasonable time. HPC architectures and applications are increasingly complex systems (heterogeneity, dynamicity, uncertainties), which leads to consider the **optimization of resource allocation based on multiple objectives**, often contradictory (like energy and run-time for instance). Focusing on the optimization of one particular objective usually leads to worsen the others. The historical positioning of some members of the team who are specialists in multi-objective optimization is to generate a (limited) set of trade-off configurations, called *Pareto points*, and choose when required the most suitable trade-off between all the objectives. This methodology differs from the classical approaches, which simplify the problem into a single objective one (focus on a particular objective, combining the various objectives or agglomerate them). The real challenge is thus to combine algorithmic techniques to account for this diversity while guaranteeing a target efficiency for all the various objectives.

The DataMove team aims to elaborate generic and effective solutions of practical interest. We make our new algorithms accessible through the team flagship software tools, **the OAR batch scheduler and the in situ processing framework FlowVR**. We maintain and enforce strong links with teams closely connected with large architecture design and operation, as well as scientists of other disciplines, in particular computational biologists, with whom we elaborate and validate new usage scenarios.

4. Application Domains

4.1. Data Aware Batch Scheduling

Large scale high performance computing platforms are becoming increasingly complex. Determining efficient allocation and scheduling strategies that can adapt to technological evolutions is a strategic and difficult challenge. We are interested in scheduling jobs in hierarchical and heterogeneous large scale platforms. On such platforms, application developers typically submit their jobs in centralized waiting queues. The job management system aims at determining a suitable allocation for the jobs, which all compete against each other for the available computing resources. Performances are measured using different classical metrics like maximum completion time or slowdown. Current systems make use of very simple (but fast) algorithms that however rely on simplistic platform and execution models, and thus, have limited performances.

For all target scheduling problems we aim to provide both theoretical analysis and complementary analysis through simulations. Achieving meaningful results will require strong improvements on existing models (on power for example) and the design of new approximation algorithms with various objectives such as stretch, reliability, throughput or energy consumption, while keeping in focus the need for a low-degree polynomial complexity.

4.1.1. Status of Current Algorithms

The most common batch scheduling policy is to consider the jobs according to the First Come First Served order (FCFS) with backfilling (BF). BF is the most widely used policy due to its easy and robust implementation and known benefits such as high system utilization. It is well-known that this strategy does not optimize any sophisticated function, but it is simple to implement and it guarantees that there is no starvation (i.e. every job will be scheduled at some moment).

More advanced algorithms are seldom used on production platforms due to both the gap between theoretical models and practical systems and speed constraints. When looking at theoretical scheduling problems, the generally accepted goal is to provide polynomial algorithms (in the number of submitted jobs and the number of involved computing units). However, with millions of processing cores where every process and data transfer have to be individually scheduled, polynomial algorithms are prohibitive as soon as the polynomial degree is too large. The model of *parallel tasks* simplifies this problem by bundling many threads and communications into single boxes, either rigid, rectangular or malleable. Especially malleable tasks capture the dynamicity of the execution. Yet these models are ill-adapted to heterogeneous platforms, as the running time depends on more than simply the number of allotted resources, and some of the common underlying assumptions on the speed-up functions (such as monotony or concavity) are most often only partially verified.

In practice, the job execution times depend on their allocation (due to communication interferences and heterogeneity in both computation and communication), while theoretical models of parallel jobs usually consider jobs as black boxes with a fixed (maximum) execution time. Though interesting and powerful, the classical models (namely, synchronous PRAM model, delay, LogP) and their variants (such as hierarchical delay), are not well-suited to large scale parallelism on platforms where the cost of moving data is significant, non uniform and may change over time. Recent studies are still refining such models in order to take into account communication contentions more accurately while remaining tractable enough to provide a useful tool for algorithm design.

Today, all algorithms in use in production systems are oblivious to communications. One of our main goals is to **design a new generation of scheduling algorithms fitting more closely job schedules according to platform topologies**.

4.1.2. Locality Aware Allocations

Recently, we developed modifications of the standard back-filling algorithm taking into account platform topologies. The proposed algorithms take into account locality and contiguity in order to hide communication patterns within parallel tasks. The main result here is to establish good lower bounds and small approximation ratios for policies respecting the locality constraints. The algorithms work in an online fashion, improving the global behavior of the system while still keeping a low running time. These improvements rely mainly on our past experience in designing approximation algorithms. Instead of relying on complex networking models and communication patterns for estimating execution times, the communications are disconnected from the execution time. Then, the scheduling problem leads to a trade-off: optimizing locality of communications on one side and a performance objective (like the makespan or stretch) on the other side.

In the perspective of taking care of locality, other ongoing works include the study of schedulers for platforms whose interconnection network is a static structured topology (like the 3D-torus of the BlueWaters platform we work on in collaboration with the Argonne National Laboratory). One main characteristic of this 3D-torus platform is to provide I/O nodes at specific locations in the topology. Applications generate and access specific data and are thus bounded to specific I/O nodes. Resource allocations are constrained in a strong and unusual way. This problem is close for actual hierarchical platforms. The scheduler needs to compute a schedule such that I/O nodes requirements are filled for each application while at the same time avoiding communication interferences. Moreover, extra constraints can arise for applications requiring accelerators that are gathered on the nodes at the edge of the network topology.

While current results are encouraging, they are however limited in performance by the low amount of information available to the scheduler. We look forward to extend ongoing work by progressively increasing application and network knowledge (by technical mechanisms like profiling or monitoring or by more sophisticated methods like learning). It is also important to anticipate on application resource usage in terms of compute units, memory as well as network and I/Os to efficiently schedule a mix of applications with different profiles. For instance, a simple solution is to partition the jobs as "communication intensive" or "low communications". Such a tag could be achieved by the users them selves or obtained by learning techniques. We could then schedule low communications jobs using leftover spaces while taking care of high communication jobs. More sophisticated options are possible, for instance those that use more detailed communication patterns and networking models. Such options would leverage the work proposed in Section 4.2 for gathering application traces.

4.1.3. Data-Centric Processing

Exascale computing is shifting away from the traditional compute-centric models to a more data-centric one. This is driven by the evolving nature of large scale distributed computing, no longer dominated by pure computations but also by the need to handle and analyze large volumes of data. These data can be large databases of results, data streamed from a running application or another scientific instrument (collider for instance). These new workloads call for specific resource allocation strategies.

Data movements and storage are expected to be a major energy and performance bottleneck on next generation platforms. Storage architectures are also evolving, the standard centralized parallel file system being complemented with local persistent storage (Burst Buffers, NVRAM). Thus, one data producer can stage data on some nodes' local storage, requiring to schedule close by the associated analytics tasks to limit data movements. This kind of configuration, often referred as *in situ analytics*, is expected to become common as it enables to switch from the traditional I/O intensive workflow (batch-processing followed by *post mortem* analysis and visualization) to a more storage conscious approach where data are processed as closely as possible to where and when they are produced (in situ processing is addressed in details in section 4.3). By reducing data movements and scheduling the extra processing on resources not fully exploited yet, in situ processing is expected to have also a significant positive energetic impact. Analytics codes can be executed in the same nodes than the application, often on dedicated cores commonly called helper cores, or on dedicated nodes called stagging nodes. The results are either forwarded to the users for visualization or saved to disk through I/O nodes. In situ analytics can also take benefit of node local disks or burst buffers to reduce data movements. Future job scheduling strategies should take into account in situ processes in addition to the job allocation to optimize both energy consumption and execution time. On the one hand, this problem can be reduced to an allocation problem of extra asynchronous tasks to idle computing units. But on the other hand, embedding analytics in applications brings extra difficulties by making the application more heterogeneous and imposing more constraints (data affinity) on the required resources. Thus, the main point here is to develop efficient algorithms for dealing with heterogeneity without increasing the global computational cost.

4.1.4. Learning

Another important issue is to adapt the job management system to deal with the bad effects of uncertainties, which may be catastrophic in large scale heterogeneous HPC platforms (jobs delayed arbitrarly far or jobs killed). A natural question is then: *is it possible to have a good estimation of the job and platform parameters in order to be able to obtain a better scheduling ?* Many important parameters (like the number or type of required resources or the estimated running time of the jobs) are asked to the users when they submit their jobs. However, some of these values are not accurate and in many cases, they are not even provided by the end-users. In DataMove, we propose to study new methods for a better prediction of the characteristics of the jobs and their execution in order to improve the optimization process. In particular, the methods well-studied in the field of big data (in supervised Machine Learning, like classical regression methods, Support Vector Methods, random forests, learning to rank techniques or deep learning) could and must be used to improve job scheduling in large scale HPC platforms. This topic received a great attention recently in the field of parallel and distributed processing. A preliminary study has been done recently by our team with the target of predicting the job running times (called wall times). We succeeded to improve significantly in average the reference EASY Back Filling algorithm by estimating the wall time of the jobs, however, this method leads to big delay for the stretch of few jobs. Even if we succeed in determining more precisely hidden parameters, like the wall time of the jobs, this is not enough to determine an optimized solution. The shift is not only to learn on dedicated parameters but also on the scheduling policy. The data collected from the accounting and profiling of jobs can be used to better understand the needs of the jobs and through learning to propose adaptations for future submissions. The goal is to propose extensions to further improve the job scheduling and improve the performance and energy efficiency of the application. For instance preference learning may enable to compute on-line new priorities to back-fill the ready jobs.

4.1.5. Multi-objective Optimization

Several optimization questions that arise in allocation and scheduling problems lead to the study of several objectives at the same time. The goal is then not a single optimal solution, but a more complicated mathematical object that captures the notion of trade-off. In broader terms, the goal of multi-objective optimization is not to externally arbitrate on disputes between entities with different goals, but rather to explore the possible solutions to highlight the whole range of interesting compromises. A classical tool for studying such multi-objective optimization problems is to use *Pareto curves*. However, the full description of the Pareto curve can be very hard because of both the number of solutions and the hardness of computing each point. Addressing this problem will opens new methodologies for the analysis of algorithms.

To further illustrate this point here are three possible case studies with emphasis on conflicting interests measured with different objectives. While these cases are good representatives of our HPC context, there are other pertinent trade-offs we may investigate depending on the technology evolution in the coming years. This enumeration is certainly not limitative.

Energy versus Performance. The classical scheduling algorithms designed for the purpose of performance can no longer be used because performance and energy are contradictory objectives to some extent. The scheduling problem with energy becomes a multi-objective problem in nature since the energy consumption should be considered as equally important as performance at exascale. A global constraint on energy could be a first idea for determining trade-offs but the knowledge of the Pareto set (or an approximation of it) is also very useful.

Administrators versus application developers. Both are naturally interested in different objectives: In current algorithms, the performance is mainly computed from the point of view of administrators, but the users should be in the loop since they can give useful information and help to the construction of better schedules. Hence, we face again a multi-objective problem where, as in the above case, the approximation of the Pareto set provides the trade-off between the administrator view and user demands. Moreover, the objectives are usually of the same nature. For example, *max stretch* and *average stretch* are two objectives based on the slowdown factor that can interest administrators and users, respectively. In this case the study of the norm of stretch can be also used to describe the trade-off (recall that the L_1-norm corresponds to the average objective while the L_∞-norm to the max objective). Ideally, we would like to design an algorithm that gives good approximate solutions at the same time for all norms. The L_2 or L_3-norm are useful since they describe the performance of the whole schedule from the administrator point of view as well as they provide a fairness indication to the users. The hard point here is to derive theoretical analysis for such complicated tools.

Resource Augmentation. The classical resource augmentation models, i.e. speed and machine augmentation, are not sufficient to get good results when the execution of jobs cannot be frequently interrupted. However, based on a resource augmentation model recently introduced, where the algorithm may reject a small number of jobs, some members of our team have given the first interesting results in the non-preemptive direction. In general, resource augmentation can explain the intuitive good behavior of some greedy algorithms while, more interestingly, it can give ideas for new algorithms. For example, in the rejection context we could dedicate a small number of nodes for the usually problematic rejected jobs. Some initial experiments show that this can lead to a schedule for the remaining jobs that is very close to the optimal one.

4.2. Empirical Studies of Large Scale Platforms

Experiments or realistic simulations are required to take into account the impact of allocations and assess the real behavior of scheduling algorithms. While theoretical models still have their interest to lay the groundwork for algorithmic designs, the models are necessarily reflecting a purified view of the reality. As transferring our algorithm in a more practical setting is an important part of our creed, we need to ensure that the theoretical results found using simplified models can really be transposed to real situations. On the way to exascale computing, large scale systems become harder to study, to develop or to calibrate because of the costs in both time and energy of such processes. It is often impossible to convince managers to use a production cluster for several hours simply to test modifications in the RJMS. Moreover, as the existing RJMS production systems need to be highly reliable, each evolution requires several real scale test iterations. The consequence is that scheduling algorithms used in production systems are mostly outdated and not customized correctly. To circumvent this pitfall, we need to develop tools and methodologies for alternative empirical studies, from analysis of workload traces, to job models, simulation and emulation with reproducibility concerns.

4.2.1. Workload Traces with Resource Consumption

Workload traces are the base element to capture the behavior of complete systems composed of submitted jobs, running applications, and operating tools. These traces must be obtained on production platforms to provide relevant and representative data. To get a better understanding of the use of such systems, we need to look at both, how the jobs interact with the job management system, and how they use the allocated resources. We propose a general workload trace format that adds jobs resource consumption to the commonly used SWF [6] workload trace format. This requires to instrument the platforms, in particular to trace resource consumptions like CPU, data movements at memory, network and I/O levels, with an acceptable performance impact. In a previous work we studied and proposed a dedicated job monitoring tool whose impact on the system has been measured as lightweight (0.35% speed-down) with a 1 minute sampling rate. Other tools also explore job monitoring, like TACC Stats. A unique feature from our tool is its ability to monitor distinctly jobs sharing common nodes.

Collected workload traces with jobs resource consumption will be publicly released and serve to provide data for works presented in Section 4.1. The trace analysis is expected to give valuable insights to define models encompassing complex behaviours like network topology sensitivity, network congestion and resource interferences.

We expect to join efforts with partners for collecting quality traces (ATOS/Bull, Ciment meso center, Joint Laboratory on Extreme Scale Computing) and will collaborate with the Inria team POLARIS for their analysis.

4.2.2. Simulation

Simulations of large scale systems are faster by multiple orders of magnitude than real experiments. Unfortunately, replacing experiments with simulations is not as easy as it may sound, as it brings a host of new problems to address in order to ensure that the simulations are closely approximating the execution of typical workloads on real production clusters. Most of these problems are actually not directly related to scheduling algorithms assessment, in the sense that the workload and platform models should be defined independently from the algorithm evaluations, in order to ensure a fair assessment of the algorithms' strengths and weaknesses. These research topics (namely platform modeling, job models and simulator calibration) are addressed in the other subsections.

We developed an open source platform simulator within DataMove (in conjunction with the OAR development team) to provide a widely distributable test bed for reproducible scheduling algorithm evaluation. Our simulator, named Batsim, allows to simulate the behavior of a computational platform executing a workload scheduled by any given scheduling algorithm. To obtain sound simulation results and to broaden the scope of the experiments that can be done thanks to Batsim, we did not chose to create a (necessarily limited) simulator from scratch, but instead to build on top of the SimGrid simulation framework.

[6] Standard Workload Format: http://www.cs.huji.ac.il/labs/parallel/workload/swf.html

To be open to as many batch schedulers as possible, Batsim decouples the platform simulation and the scheduling decisions in two clearly-separated software components communicating through a complete and documented protocol. The Batsim component is in charge of simulating the computational resources behaviour whereas the scheduler component is in charge of taking scheduling decisions. The scheduler component may be both a resource and a job management system. For jobs, scheduling decisions can be to execute a job, to delay its execution or simply to reject it. For resources, other decisions can be taken, for example to change the power state of a machine i.e. to change its speed (in order to lower its energy consumption) or to switch it on or off. This separation of concerns also enables interfacing with potentially any commercial RJMS, as long as the communication protocol with Batsim is implemented. A proof of concept is already available with the OAR RJMS.

Using this test bed opens new research perspectives. It allows to test a large range of platforms and workloads to better understand the real behavior of our algorithms in a production setting. In turn, this opens the possibility to tailor algorithms for a particular platform or application, and to precisely identify the possible shortcomings of the theoretical models used.

4.2.3. Job and Platform Models

The central purpose of the Batsim simulator is to simulate job behaviors on a given target platform under a given resource allocation policy. Depending on the workload, a significant number of jobs are parallel applications with communications and file system accesses. It is not conceivable to simulate individually all these operations for each job on large plaforms with their associated workload due to implied simulation complexity. The challenge is to define a coarse grain job model accurate enough to reproduce parallel application behavior according to the target platform characteristics. We will explore models similar to the BSP (Bulk Synchronous Program) approach that decomposes an application in local computation supersteps ended by global communications and a global synchronization. The model parameters will be established by means of trace analysis as discussed previously, but also by instrumenting some parallel applications to capture communication patterns. This instrumentation will have a significant impact on the concerned application performance, restricting its use to a few applications only. There are a lot of recurrent applications executed on HPC platform, this fact will help to reduce the required number of instrumentations and captures. To assign each job a model, we are considering to adapt the concept of application signatures as proposed in. Platform models and their calibration are also required. Large parts of these models, like those related to network, are provided by Simgrid. Other parts as the filesystem and energy models are comparatively recent and will need to be enhanced or reworked to reflect the HPC platform evolutions. These models are then generally calibrated by running suitable benchmarks.

4.2.4. Emulation and Reproducibility

The use of coarse models in simulation implies to set aside some details. This simplification may hide system behaviors that could impact significantly and negatively the metrics we try to enhance. This issue is particularly relevant when large scale platforms are considered due to the impossibility to run tests at nominal scale on these real platforms. A common approach to circumvent this issue is the use of emulation techniques to reproduce, under certain conditions, the behavior of large platforms on smaller ones. Emulation represents a natural complement to simulation by allowing to execute directly large parts of the actual evaluated software and system, but at the price of larger compute times and a need for more resources. The emulation approach was chosen in to compare two job management systems from workload traces of the CURIE supercomputer (80000 cores). The challenge is to design methods and tools to emulate with sufficient accuracy the platform and the workload (data movement, I/O transfers, communication, applications interference). We will also intend to leverage emulation tools like Distem from the MADYNES team. It is also important to note that the Batsim simulator also uses emulation techniques to support the core scheduling module from actual RJMS. But the integration level is not the same when considering emulation for larger parts of the system (RJMS, compute node, network and filesystem).

Replaying traces implies to prepare and manage complex software stacks including the OS, the resource management system, the distributed filesystem and the applications as well as the tools required to conduct experiments. Preparing these stacks generate specific issues, one of the major one being the support for reproducibility. We propose to further develop the concept of reconstructability to improve experiment reproducibility by capturing the build process of the complete software stack. This approach ensures reproducibility over time better than other ways by keeping all data (original packages, build recipe and Kameleon engine) needed to build the software stack.

In this context, the Grid'5000 (see Sec. 5.3) experimentation infrastructure that gives users the control on the complete software stack is a crucial tool for our research goals. We will pursue our strong implication in this infrastructure.

4.3. Integration of High Performance Computing and Data Analytics

Data produced by large simulations are traditionally handled by an I/O layer that moves them from the compute cores to the file system. Analysis of these data are performed after reading them back from files, using some domain specific codes or some scientific visualisation libraries like VTK. But writing and then reading back these data generates a lot of data movements and puts under pressure the file system. To reduce these data movements, **the in situ analytics paradigm proposes to process the data as closely as possible to where and when the data are produced**. Some early solutions emerged either as extensions of visualisation tools or of I/O libraries like ADIOS. But significant progresses are still required to provide efficient and flexible high performance scientific data analysis tools. Integrating data analytics in the HPC context will have an impact on resource allocation strategies, analysis algorithms, data storage and access, as well as computer architectures and software infrastructures. But this paradigm shift imposed by the machine performance also sets the basis for a deep change on the way users work with numerical simulations. The traditional workflow needs to be reinvented to make HPC more user-centric, more interactive and turn HPC into a commodity tool for scientific discovery and engineering developments. In this context DataMove aims at investigating programming environments for in situ analytics with a specific focus on task scheduling in particular, to ensure an efficient sharing of resources with the simulation.

4.3.1. Programming Model and Software Architecture

In situ creates a tighter loop between the scientist and her/his simulation. As such, an in situ framework needs to be flexible to let the user define and deploy its own set of analysis. A manageable flexibility requires to favor simplicity and understandability, while still enabling an efficient use of parallel resources. Visualization libraries like VTK or Visit, as well as domain specific environments like VMD have initially been developed for traditional post-mortem data analysis. They have been extended to support in situ processing with some simple resource allocation strategies but the level of performance, flexibility and ease of use that is expected requires to rethink new environments. There is a need to develop a middleware and programming environment taking into account in its fundations this specific context of high performance scientific analytics.

Similar needs for new data processing architectures occurred for the emerging area of Big Data Analytics, mainly targeted to web data on cloud-based infrastructures. Google Map/Reduce and its successors like Spark or Stratosphere/Flink have been designed to match the specific context of efficient analytics for large volumes of data produced on the web, on social networks, or generated by business applications. These systems have mainly been developed for cloud infrastructures based on commodity architectures. They do not leverage the specifics of HPC infrastructures. Some preliminary adaptations have been proposed for handling scientific data in a HPC context. However, these approaches do not support in situ processing.

Following the initial development of FlowVR, our middleware for in situ processing, we will pursue our effort to develop a programming environment and software architecture for high performance scientific data analytics. Like FlowVR, the map/reduce tools, as well as the machine learning frameworks like TensorFlow, adopted a dataflow graph for expressing analytics pipe-lines. We are convinced that this dataflow approach is both easy to understand and yet expresses enough concurrency to enable efficient executions. The graph description can be compiled towards lower level representations, a mechanism that is intensively used by Stratosphere/Flink for instance. Existing in situ frameworks, including FlowVR, inherit from the HPC way of programming with a thiner software stack and a programming model close to the machine. Though this approach enables to program high performance applications, this is usually too low level to enable the scientist to write its analysis pipe-line in a short amount of time. The data model, i.e. the data semantics level accessible at the framework level for error check and optimizations, is also a fundamental aspect of such environments. The key/value store has been adopted by all map/reduce tools. Except in some situations, it cannot be adopted as such for scientific data. Results from numerical simulations are often more structured than web data, associated with acceleration data structures to be processed efficiently. We will investigate data models for scientific data building on existing approaches like Adios or DataSpaces.

4.3.2. Resource Sharing

To alleviate the I/O bottleneck, the in situ paradigm proposes to start processing data as soon as made available by the simulation, while still residing in the memory of the compute node. In situ processings include data compression, indexing, computation of various types of descriptors (1D, 2D, images, etc.). Per se, reducing data output to limit I/O related performance drops or keep the output data size manageable is not new. Scientists have relied on solutions as simple as decreasing the frequency of result savings. In situ processing proposes to move one step further, by providing a full fledged processing framework enabling scientists to more easily and thoroughly manage the available I/O budget.

The most direct way to perform in situ analytics is to inline computations directly in the simulation code. In this case, in situ processing is executed in sequence with the simulation that is suspended meanwhile. Though this approach is direct to implement and does not require complex framework environments, it does not enable to overlap analytics related computations and data movements with the simulation execution, preventing to efficiently use the available resources. Instead of relying on this simple time sharing approach, several works propose to rely on space sharing where one or several cores per node, called *helper cores*, are dedicated to analytics. The simulation responsibility is simply to handle a copy of the relevant data to the node-local in situ processes, both codes being executed concurrently. This approach often lead to significantly beter performance than in-simulation analytics.

For a better isolation of the simulation and in situ processes, one solution consists in offloading in situ tasks from the simulation nodes towards extra dedicated nodes, usually called *staging nodes*. These computations are said to be performed *in-transit*. But this approach may not always be beneficial compared to processing on simulation nodes due to the costs of moving the data from the simulation nodes to the staging nodes.

FlowVR enables to mix these different resources allocation strategies for the different stages of an analytics pile-line. Based on a component model, the scientist designs analytics workflows by first developing processing components that are next assembled in a dataflow graph through a Python script. At runtime the graph is instantiated according to the execution context, FlowVR taking care of deploying the application on the target architecture, and of coordinating the analytics workflows with the simulation execution.

But today the choice of the resource allocation strategy is mostly ad-hoc and defined by the programmer. We will investigate solutions that enable a cooperative use of the resource between the analytics and the simulation with minimal hints from the programmer. In situ processings inherit from the parallelization scale and data distribution adopted by the simulation, and must execute with minimal perturbations on the simulation execution (whose actual resource usage is difficult to know a priori). We need to develop adapted scheduling strategies that operate at compile and run time. Because analysis are often data intensive, such solutions must take into consideration data movements, a point that classical scheduling strategies designed first for compute intensive applications often overlook. We expect to develop new scheduling strategies relying on the methodologies developed in Section 4.1.5. Simulations as well as analysis are iterative processes exposing a strong spatial and temporal coherency that we can take benefit of to anticipate their behavior and then take more relevant resources allocation strategies, possibly based on advanced learning algorithms or as developed in Section 4.1.

In situ analytics represent a specific workload that needs to be scheduled very closely to the simulation, but not necessarily active during the full extent of the simulation execution and that may also require to access data from previous runs (stored in the file system or on specific burst-buffers). Several users may also need to run concurrent analytics pipe-lines on shared data. This departs significantly from the traditional batch scheduling model, motivating the need for a more elastic approach to resource provisioning. These issues will be conjointly addressed with research on batch scheduling policies (Section 4.1).

4.3.3. Co-Design with Data Scientists

Given the importance of users in this context, it is of primary importance that in situ tools be co-designed with advanced users, even if such multidisciplinary collaborations are challenging and require constant long term investments to learn and understand the specific practices and expectations of the other domain.

We will tightly collaborate with scientists of some application domains, like molecular dynamics or fluid simulation, to design, develop, deploy and assess in situ analytics scenarios, as already done with Marc Baaden, a computational biologist from LBT.

We recently extended our collaboration network. We started in 2015 a PhD co-advised with CEA DAM to investigate in situ analytics scenarios in the context of atomistic material simulations. CEA DAM is a French energy lab hosting one of the largest european supercomputer. They gather physicists, numerical scientists as well as high performance computer engineers, making it a very interesting partner for developing new scientific data analysis solutions. We also got a national grant (2015-2018) to compute in situ statistics for multi-parametric parallel studies with the research department of French power company EDF. In this context we collaborate with statisticians and fluid simulation experts to define in situ scenarios, revisit the statistic operators to be amenable to in situ processing, and define an adapted in situ framework.

5. New Software and Platforms

5.1. OAR

KEYWORDS: HPC - Cloud - Clusters - Resource manager - Light grid
SCIENTIFIC DESCRIPTION This batch system is based on a database (PostgreSQL (preferred) or MySQL), a script language (Perl) and an optional scalable administrative tool (e.g. Taktuk). It is composed of modules which interact mainly via the database and are executed as independent programs. Therefore, formally, there is no API, the system interaction is completely defined by the database schema. This approach eases the development of specific modules. Indeed, each module (such as schedulers) may be developed in any language having a database access library.
FUNCTIONAL DESCRIPTION OAR is a versatile resource and task manager (also called a batch scheduler) for HPC clusters, and other computing infrastructures (like distributed computing experimental testbeds where versatility is a key).
The OAR ecosystem also include several associated software tools that proved to be useful independently from OAR. Among theses, two softwares play a major role in the support our research studies. The first one is Kameleon (http://kameleon.imag.fr), a tool to help enhancing reproducibility of experiments by guarantee the ability to reproduce the complete used software stacks. The second one is Batsim (https://gforge.inria.fr/projects/batsim) a RJMS simulator based on SimGrid. Batsim simulates job execution taking into account the target platform hardware capabilities through SimGrid, while scheduling is performed by an actual job management system. A comprehensive API enables to easily plug into BatSim various job management systems like OAR.

- Participants: Olivier Richard, Pierre Neyron, Salem Harrache and Bruno Bzeznik
- Partners: CIMENT - CNRS - Grid'5000 - LIG
- Contact: Olivier Richard
- URL: http://oar.imag.fr

5.2. FlowVR

KEYWORDS: HPC - In Situ Processing - Computational Steering

SCIENTIFIC DESCRIPTION FlowVR is an open source middelware to augment parallel simulations running on thousands of cores with in situ processing capabilities and live steering. FlowVR offers a very flexible environment while enabling high performance asynchronous in situ and in transit processing.

FUNCTIONAL DESCRIPTION FlowVR adopts the "data-flow" paradigm, where your application is divided as a set of components exchanging messages (think of it as a directed graph). FlowVR enables to encapsulate existing codes in components, interconnect them through data channels, and deploy them on distributed computing resources. FlowVR takes care of all the heavy lifting such as application deployment and message exchange.

- Participants: Bruno Raffin, Matthieu Dreher, Jérémy Jaussaud
- Contact: Bruno Raffin
- URL: http://flowvr.sf.net

5.3. Platforms

5.3.1. *Grid'5000 (https://www.grid5000.fr/) and meso center Ciment (https://ciment.ujf-grenoble.fr)*

We have been very active in promoting the factorization of compute resources at a regional and national level. We have a three level implication, locally to maintain a pool of very flexible experimental machines (hundreds of cores), regionally through the CIMENT meso center (Equipex Grant), and nationally by contributing to the Grid'5000 platform, our local resources being included in this platform. Olivier Richard is member of Grid'5000 scientific committee and Pierre Neyron is member of the technical committee. The OAR scheduler in particular is deployed on both infrastructures. We are currently preparing proposals for the next generation machines within the context of the new university association (Univ. Grenoble-Alpes).

6. New Results

6.1. In Situ Statistical Analysis for Parametric Studies

In situ processing proposes to reduce storage needs and I/O traffic by processing results of parallel simulations as soon as they are available in the memory of the compute processes. We focus in this paper [11] on computing in situ statistics on the results of N simulations from a parametric study. The classical approach consists in running various instances of the same simulation with different values of input parameters. Results are then saved to disks and statistics are computed post mortem, leading to very I/O intensive applications. Our solution is to develop Melissa, an in situ library running on staging nodes as a parallel server. When starting, simulations connect to Melissa and send the results of each time step to Melissa as soon as they are available. Melissa implements iterative versions of classical statistical operations, enabling to update results as soon as a new time step from a simulation is available. Once all statistics ar updated, the time step can be discarded. We also discuss two different approaches for scheduling simulation runs: the jobs-in-job and the multi-jobs approaches. Experiments run instances of the Computational Fluid Dynamics Open Source solver Code_Saturne. They confirm that our approach enables one to avoid storing simulation results to disk or in memory.

6.2. Online Non-preemptive Scheduling in a Resource Augmentation Model based on Duality

Resource augmentation is a well-established model for analyzing algorithms, particularly in the online setting. It has been successfully used for providing theoretical evidence for several heuristics in scheduling with good performance in practice. According to this model, the algorithm is applied to a more powerful environment than that of the adversary. Several types of resource augmentation for scheduling problems have been proposed up to now, including speed augmentation, machine augmentation and more recently rejection. In this paper [7], we present a framework that unifies the various types of resource augmentation. Moreover, it allows generalize the notion of resource augmentation for other types of resources. Our framework is based on mathematical programming and it consists of extending the domain of feasible solutions for the algorithm with respect to the domain of the adversary. This, in turn allows the natural concept of duality for mathematical programming to be used as a tool for the analysis of the algorithm's performance. As an illustration of the above ideas, we apply this framework and we propose a primal-dual algorithm for the online scheduling problem of minimizing the total weighted flow time of jobs on unrelated machines when the preemption of jobs is not allowed. This is a well representative problem for which no online algorithm with performance guarantee is known. Specifically, a strong lower bound of $\Omega(\sqrt{n})$ exists even for the offline unweighted version of the problem on a single machine. In this paper, we first show a strong negative result even when speed augmentation is used in the online setting. Then, using the generalized framework for resource augmentation and by combining speed augmentation and rejection, we present an $(1+\epsilon_s)$-speed $O(\frac{1}{\epsilon_s \epsilon_r})$-competitive algorithm if we are allowed to reject jobs whose total weight is an ϵ_r-fraction of the weights of all jobs, for any $\epsilon_s > 0$ and $\epsilon_r \in (0,1)$. Furthermore, we extend the idea for analysis of the above problem and we propose an $(1+\epsilon_s)$-speed ϵ_r-rejection $O(\frac{k^{\frac{(k+3}{k}}}{\epsilon_r^{1/k} \epsilon_s^{\frac{(k+2)}{k}}})$-competitive algorithm for the more general objective of minimizing the weighted l_k-norm of the flow times of jobs.

6.3. Batsim: a Realistic Language-Independent Resources and Jobs Management Systems Simulator

As large scale computation systems are growing to exascale, Resources and Jobs Management Systems (RJMS) need to evolve to manage this scale modification. However, their study is problematic since they are critical production systems, where experimenting is extremely costly due to downtime and energy costs. Meanwhile, many scheduling algorithms emerging from theoretical studies have not been transferred to production tools for lack of realistic experimental validation. To tackle these problems we propose Batsim [6], an extendable, language-independent and scalable RJMS simulator. It allows researchers and engineers to test and compare any scheduling algorithm, using a simple event-based communication interface, which allows different levels of realism. In this paper we show that Batsim's behavior matches the one of the real RJMS OAR. Our evaluation process was made with reproducibility in mind and all the experiment material is freely available.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

BULL-ATOS SE (2015-2018). Two PhD grants (David Glesser and Michael Mercier). Job and resource management algorithms.

7.2. Bilateral Grants with Industry

CEA DAM (2016-2018). PhD grant support contract (PhD of Estelle Dirand, funded by CEA). In situ analysis for Molecular Simulations.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

- **ANR grant MOEBIUS (2013-2017).** Multi-objective scheduling for large computing platforms. Coordinator: Grenoble-INP (DataMove). Partners: Grenoble-INP, Inria, BULL-ATOS .

8.1.2. Competitivity Clusters

- **PIA Avido (2015-2018)**. In situ analysis and visualization for large scale numerical simulation. Coordinator: EDF SA. Partners: EDF SA, Total SA, Kitware SAS , Université Pierre et Marie CURIE, Inria (DataMove).
- **FUI OverMind (2015-2017)**. Task planification and asset management for the cartoon productions. Coordinator: Teamto Studio. Partners: Teamto Studio, Folimage Studio, Ecole de Gobelins, Inria (DataMove).

8.2. European Initiatives

8.2.1. FP7 & H2020 Projects

8.2.1.1. VELaSSCo

Title: Visualization For Extremely Large-Scale Scientific Computing

Program: STREP (Specific Targeted Research Project)

Duration: January 2014 - December 2016

Coordinator: Centre Internacional de Metodes Numerics en Enginyeria (Spain)

Partners: JOTNE (No.), SINTEF (No.), Fraunhofer IGD (D), ATOS (SP), Univ. Edinburgh (UK)

Inria contact: Toan Nguyen, Bruno Raffin

Abstract: VELaSSCo aims at developing a new concept of integrated end-user visual analysis methods with advanced management and post-processing algorithms for engineering modelling applications, scalable for real-time petabyte level simulations [59]. The interface will enable real-time interrogation of simulation data, generating key information for analysis. Main concerns have to do with handling of large amounts of data of a very specific kind intrinsically linked to geometrical properties; how to store, access, simplify and manipulate billion of records to extract the relevant information; how to represent information in a feasible and flexible way; and how to visualise and interactively inspect the huge quantity of information they produce taking into account end-user's needs. VELaSSCo achieves this by putting together experts with relevant background in Big Data handling, advanced visualisation, engineering simulations, and a User Panel including research centres, SMEs and companies form key European industrial sectors such as aerospace, household products, chemical, pharmaceutical and civil engineering.

8.3. International Initiatives

8.3.1. Inria International Labs

8.3.1.1. JLESC

Title: Joint Laboratory for Extreme-Scale-Computing.

International Partners:

University of Illinois at Urbana Champaign (USA)

Argonne National Laboratory (USA),

Barcelona Supercomputing Center (Spain),

Jülich Supercomputing Centre (Germany)

Riken Advanced Institute for Computational Science (Japan)

Start year: 2009

See also: https://jlesc.github.io/

The purpose of the Joint Laboratory for Extreme Scale Computing is to be an international, virtual organization whose goal is to enhance the ability of member organizations and investigators to make the bridge between Petascale and Extreme computing. The JLESC organizes a workshop every 6 months DataMove participates to. DataMove developed several collaborations related to in situ processing with Tom Peterka group (ANL) , the Argo exascale operating system with Swann Perarnau (ANL).

8.3.1.2. ANOMALIES@EXASCALE

Title: Anomalies Detection and Handling towards Exascale Platforms

International Partner:

University of Chicago (United States) - Argonne National Laboratory (ANL)

Start year: 2014. End year: 2016.

See also: http://anomalies.imag.fr

The Anomalies@exascale project intends to prospect new scheduling solutions for very large parallel computing platforms. In particular, we consider the new problems related to fault tolerance raising with the developments of exascale platforms. We expect to define new ways to detect both execution failures and more transient performance anomalies. Information gathered from the detectors will then be taken into account by schedulers to implement corrective measures. PI: Frederic Wagner

8.3.2. Inria Associate Teams Not Involved in an Inria International Labs

8.3.2.1. ExaSE

Title: Exascale Computing Scheduling and Energy

International Partners:

UFRGS, PUC Minas and UPS (Brazil)

Duration: 2014 - 2016

See also: https://team.inria.fr/exase/

The main scientific context of this project is high performance computing on Exascale systems: large-scale machines with billions of processing cores and complex hierarchical structures. This project intends to explore the relationship between scheduling algorithms and techniques and the energy constraints present on such exascale systems. PI: Jean-Marc Vincent (Polaris)

8.3.3. Participation in Other International Programs

8.3.3.1. LICIA

Title: International Laboratory in High Performance and Ubiquitous Computing

International Partner (Institution - Laboratory - Researcher):

UFRGS (Brazil)

Duration: 2011 - 2018

See also: http://licia-lab.org/

The LICIA is an Internacional Laboratory and High Performance and Ubiquitous Computing born in 2011 from the common desire of members of Informatics Institute of the Federal University of Rio Grande do Sul and of Laboratoire d'Informatique de Grenoble to enhance and develop their scientific partnership that started by the end of the 1970. LICIA is an Internacional Associated Lab of the CNRS, a public french research institution. It has support from several brazilian and french research funding agencies, such as CNRS, Inria, ANR, European Union (from the french side) and CAPES, CNPq, FAPERGS (from the Brazilian side). DataMove is deeply involved in the animation of LICIA. Bruno Raffin is LICIA associate director.

8.3.3.2. CAPES/COFECUB StarShip

Title: Scalable Tools and Algorithms para Resilient, Scalable, Hybrid Interactive Processing

International Partner (Institution - Laboratory - Researcher):

UFRGS (Brazil)

Duration: 2013 - 2016

PI: Bruno Raffin (DataMove) and Alexandre Carissimi (UFRGS)

8.4. International Research Visitors

8.4.1. Internships

PhD in progress: Marcos Amaris Gonzalez, Performance Evaluation for GPU, USP (Sao Paulo, Brasil). 1 year "sandwich" visit. Local adviser: Denis Trystram

8.4.2. Visits to International Teams

- Pierre François Dutot. Six month stay at University of Hawaii at Manoa (Sept. 2016 - Jan. 2017)

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Organisation

9.1.1.1. General Chair, Scientific Chair

- Euro-Par, Grenoble, August 2016: General chair and local organization.
- HCW'2016 (25th IEEE Heterogeneous Computing Workshop), Hyderabad, May 2016a: General Chair

9.1.1.2. Member of the Organizing Committees

- EGPGV (Eurographics Symposium on Parallel Rendering and Visualization): President of the steering committee.

9.1.2. Scientific Events Selection

9.1.2.1. Chair of Conference Program Committees

Euro-Par, Grenoble, August 2016: Topic chair.

9.1.2.2. Member of the Conference Program Committees

ISAV 2016, November 2016, Salt Lake City, USA

2nd IEEE BidDataSecurity, April 8-10 2016, New York, USA

IPDPS 2016 (27th IEEE International Parallel & Distributed Processing Symposium), May 23-27 2016, Chicago, USA

CloudTech, My 24-26 2016, Marrakech, Marocco

COMPAS, July 5-8 2016, Lorient, France

PMAA'16 (10th internat. workshop on Parallel Matrix Algorithms and Applications), July 6-8 2016, Bordeaux, France

ISPDC (15th Internat Symposium on Parallel and Distributed Computing), July 8-10 2016, Fuzhou, China

EuroMPI, September 25-28 2016, Edinburgh, Scotland, UK

28th SBAC-PAD, October 26-28 2016, Los Angeles, USA,

Edu-HPC (Workshop on Education for High-Performance Computing), November 2016, Salt Lake city, USA

CloudCom, December 12-15 2016, Luxemburg

9.1.3. Journal

9.1.3.1. *Member of the Editorial Boards*

Associate Editor of the Parallel Computing journal PARCO.

Member of the Editorial Board of JPDC.

Member of the Editorial Board of Computational Methods in Science and Technology.

Member of the Editorial Board of ARIMA (revue africaine de recherche en informatique et maths appliquées).

Member of the Editorial Board of IEEE Trans. Parallel and Distributed Systems TPDS.

9.1.4. Scientific Expertise

ANR project evaluation expert

Nederlands e-science center expert

9.1.5. Research Administration

Executive committee member of Mathematics and Computer Science Council of Univ. Grenoble-Alpes (Membre du directoire du Conseil du Pôle MSTIC de l'UGA)

Mathematics and Computer Science Council of Univ. Grenoble-Alpes Members (Membre du Conseil du Pôle MSTIC de l'UGA)

Steering commitee of Grid'5000

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master: Denis Trystram is responsible of the first year (M1) of the international Master of Science in Informatics at Grenoble (MOSIG-M1).

Master: D. Trystram, P.-F. Dutot, "Complexity, approximation theory and randomization" master course (M2) at Univ. Grenoble-Alpes

Master: Pierre-François Dutot. 226 hours per year. Licence (first and second year) at IUT2/UPMF (Institut Universitaire Technologique de Univ. Grenoble-Alpes) and 9 hours Master M2R-ISC Informatique-Systèmes-Communication at Univ. Grenoble-Alpes.

Master: Grégory Mounié. 242 hours per year. Master (M1/2nd year and M2/3rd year) at Engineering school ENSIMAG, Grenoble-INP.

Master: Bruno Raffin. 28 hours per year. Parallel System. International Master of Science in Informatics at Grenoble (MOSIG-M2).

Master: Olivier Richard. 222 hours per year. Master at Engineering school Polytech-Grenoble, Univ. Grenoble-Alpes.

Master: Denis Trystram. 200 hours per year in average, mainly at first level of Engineering School ENSIMAG, Grenoble-INP.

Master: Frédéric Wagner. 220 hours per year. Engineering school ENSIMAG, Grenoble-INP (M1/2nd year and M2/3rd year) (190h), Master DESS/M2-P SCCI Security (30h).

9.2.2. Supervision

PhD: David Glesser, Energy Aware Resource Management for HPC, Univ. Grenoble-Alpes. Defended November 2016. Advisers: Denis Trystram and Yianis Georgiou (ATOS/BULL)

PhD : Marwa Sridi, Un modèle de structure de données Cache-aware pour un parallélisme et un l'équilibrage dynamique de la charge, Univ Grenoble-Alpes. Defended April 2016. Advisers: Bruno Raffin, Vincent Faucher (CEA) and Thierry Gautier.

PhD in progress : Julio Toss, Parallel Algorithms and Data Structures for Physically Based Simulation of Deformable Objects, Univ. Grenoble-Alpes and UFRGS (co-tutelle). Started October 2013. Advisers: Bruno Raffin and Joao Comba (UFRGS).

PhD in progress : Estelle Dirand, Integration of High-Performance Data Analytics and IOs for Molecular Dynamics on Exascale Computer, Univ. Grenoble-Alpes. Started January 2016. Advisers: Bruno Raffin and Laurent Colombet (CEA).

PhD in progress: Michael Mercier, Resource Management and Job Scheduling in HPC–Cloud environments towards the Big Data era, Univ. Grenoble Alpes. Started October 2016. Advisers: Olivier Richard and Bruno Raffin.

PhD in progress: Raphaël Bleuse, Affinity Scheduling, Univ. Grenoble-Alpes. Started October 2013. Adviser: Denis Trystram and Gregory Mounié.

PhD in progress: Millian Poquet, Energy consumption optimization for high performance computing, Univ. Grenoble-Alpes. Started October 2014. Advisers: Denis Trystram and Pierre-François Dutot

PhD in progress: Valentin Reis, Machine Learning for resource management, Univ. Grenoble-Alpes. Started October 2015. Advisers: Denis Trystram and Eric Gaussier

PhD in progress: Abhinav Srivastav, Multi-objective Scheduling, Univ. Grenoble-Alpes. Started October 2015. Advisers: Denis Trystram and Oded Maler

PhD in progress: Alessandro Kraemer, Scheduling in the Cloud, Univ Grenoble-Alpes and UFPR (co-tutelle). Started October 2014. Advisers: Olivier Richard and Denis Trystram.

PhD in progress: Fernando Machado Mendonca, Locality Aware Scheduling, Univ. Grenoble-Alpes, Advisers: Frederic Wagner and Denis Trystram.

PhD in progress: Mohammed Khatiri, Tasks scheduling on heterogeneous Multicore, Univ. Grenoble-Alpes and University Mohammed First (co-tutelle), Advisers: Denis Trystram, El Mostafa DAOUDI (University Mohammed First, Oujda, Maroc)

9.2.3. Juries

PhD Defense of François Lehericey, 20th of September 2016. Jury Member. Ray Tracing Based Collision Detection: A quest for Performance. Université Bretagne Loire.

PhD Defense of Rémy Dautriche, 20th of October 2016. Jury President. Multi-scale Interaction Techniques for the Interactive Visualization of Execution Traces. Univ Grenoble-Alpes.

PhD Defense of Xiaohu Wu, 16th of February 2016. Reviewer. University of Nice.

PhD Defense of Ziad Sultan, 17th of June 2016. Jury President. Parallel Generic and Adaptive Exacte Linear Algebra. Univ. Grenoble-Alpes

9.3. Popularization

In conjonction with the Polaris team, the team (Seniors and PhDs) participates to several "computer science with hands" events, notably the "Fête de la science", every year, but also some visits of classes from high school of the area along the year at Inria Rhône-Alpes.

Talk at the ISN conference organized by Inria, dedicated to present the computer science to teachers of High School

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] D. GLESSER. *Road to exascale: Improving scheduling performances and reducing energy consumption with the help of end-users*, Univ. Grenoble Alpes, October 2016, https://hal.inria.fr/tel-01425620

[2] M. SRIDI. *Cache Aware Dynamics Data Layout for Efficient Shared Memory Parallelisation*, Université de Grenoble Alpes, April 2016, https://hal.archives-ouvertes.fr/tel-01430501

Articles in International Peer-Reviewed Journals

[3] K. JANSEN, D. TRYSTRAM. *Scheduling parallel jobs on heterogeneous platforms*, in "Electronic Notes in Discrete Mathematics", 2016, vol. 55, pp. 9–12 [*DOI :* 10.1016/J.ENDM.2016.10.003], https://hal.archives-ouvertes.fr/hal-01427256

Articles in Non Peer-Reviewed Journals

[4] J. TOSS, J. COMBA, B. RAFFIN. *Parallel Voronoi Computation for Physics-Based Simulations*, in "Computing in Science and Engineering", May 2016, vol. 18, n^{o} 3, 88 p. [*DOI :* 10.1109/MCSE.2016.52], https://hal.inria.fr/hal-01317549

International Conferences with Proceedings

[5] K. ALESSANDRO, C. MAZIERO, O. RICHARD. *Reducing the Number of Response Time SLO Violations by a Cloud-HPC Convergence Scheduler*, in "2nd International Conference on Cloud Computing Technologies and Applications (CloudTech'16)", Marrakech, Morocco, May 2016, https://hal.inria.fr/hal-01432583

[6] P.-F. DUTOT, M. MERCIER, M. POQUET, O. RICHARD. *Batsim: a Realistic Language-Independent Resources and Jobs Management Systems Simulator*, in "20th Workshop on Job Scheduling Strategies for Parallel Processing", Chicago, United States, May 2016, https://hal.archives-ouvertes.fr/hal-01333471

[7] G. LUCARELLI, N. KIM THANG, A. SRIVASTAV, D. TRYSTRAM. *Online Non-preemptive Scheduling in a Resource Augmentation Model based on Duality*, in "European Symposium on Algorithms (ESA 2016)", Aarhus, Denmark, August 2016, vol. 57, n^{o} 63, pp. 1-17 [*DOI :* 10.4230/LIPICS.ESA.2016.63], http://hal.univ-grenoble-alpes.fr/hal-01334219

[8] G. LUCARELLI, A. SRIVASTAV, D. TRYSTRAM. *From Preemptive to Non-preemptive Scheduling Using Rejections*, in "22nd International Computing and Combinatorics Conference (COCOON 2016)", Ho Chi Minh Ville, Vietnam, August 2016, vol. 9797, pp. 510-519 [*DOI :* 10.1007/978-3-319-42634-1_41], http://hal.univ-grenoble-alpes.fr/hal-01371023

[9] Y. NGOKO, D. TRYSTRAM, V. REIS, C. CÉRIN. *An Automatic Tuning System for Solving NP-Hard Problems in Clouds*, in "IPDPSW 2016 - IEEE International Parallel and Distributed Processing Symposium Workshops", Chicago, United States, May 2016, pp. 1443–1452 [*DOI :* 10.1109/IPDPSW.2016.68], https://hal.archives-ouvertes.fr/hal-01427255

[10] M. SRIDI, B. RAFFIN, V. FAUCHER. *Cache Aware Dynamics Data Layout for Efficient Shared Memory Parallelisation of EUROPLEXUS*, in "International Conference on Computational Science (ICCS)", San Diego, United States, Procedia Computer Science, June 2016, vol. 80, pp. 1083 - 1092 [*DOI :* 10.1016/J.PROCS.2016.05.413], https://hal.archives-ouvertes.fr/hal-01420005

[11] T. TERRAZ, B. RAFFIN, A. RIBES, Y. FOURNIER. *In Situ Statistical Analysis for Parametric Studies*, in "In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV2016)", Salt Lake City, United States, November 2016, https://hal.archives-ouvertes.fr/hal-01383860

Conferences without Proceedings

[12] P.-F. Dutot, E. Saule, A. Srivastav, D. Trystram. *Online Non-Preemptive Scheduling to Optimize Max Stretch on a Single Machine*, in "22nd International Computing and Combinatorics Conference (COCOON 2016)", Ho-Chi-Minh-Ville, Vietnam, August 2016, http://hal.univ-grenoble-alpes.fr/hal-01309052

Research Reports

[13] R. Bleuse, S. Hunold, S. Kedad-Sidhoum, F. Monna, G. Mounié, D. Trystram. *Scheduling Independent Moldable Tasks on Multi-Cores with GPUs*, Inria Grenoble Rhône-Alpes, Université de Grenoble, January 2016, n^o RR-8850, https://hal.archives-ouvertes.fr/hal-01263100

[14] B. Omidvar-Tehrani, S. Amer-Yahia, P.-F. Dutot, D. Trystram. *Multi-Objective Group Discovery on the Social Web (Technical Report)*, LIG, April 2016, n^o RR-LIG-052, Les rapports de recherche du LIG - ISSN: 2105-0422, https://hal.archives-ouvertes.fr/hal-01297763

Other Publications

[15] M. Amaris, G. Lucarelli, C. Mommessin, D. Trystram. *Generic algorithms for scheduling applications on hybrid multi-core machines*, December 2016, working paper or preprint, https://hal.inria.fr/hal-01420798