IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université de Grenoble Alpes

Activity Report 2017

Project-Team DATAMOVE

Data Aware Large Scale Computing

IN COLLABORATION WITH: Laboratoire d'Informatique de Grenoble (LIG)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Distributed and High Performance Computing

Table of contents

Project-Team DATAMOVE

Creation of the Team: 2016 January 01, updated into Project-Team: 2017 November 01

The DataMove team is located in the IMAG building on the Campus of Univ. Grenoble Alpes.

Keywords:

Computer Science and Digital Science:

A1.1.4. - High performance computing
A1.1.5. - Exascale
A2.1.10. - Domain-specific languages
A2.6.2. - Middleware
A7.1.2. - Parallel algorithms
A8.2. - Optimization

Other Research Topics and Application Domains:

B1.1.9. - Bioinformatics
B3.3.2. - Water: sea & ocean, lake & river
B5.5. - Materials

1. Personnel

Research Scientists

Bruno Raffin [Team leader, Inria, Senior Researcher, HDR]
Fanny Dufosse [Inria, Researcher, from Nov 2017]
Giorgio Lucarelli [Inria, Starting Research Position]

Faculty Members

Yves Denneulin [Institut polytechnique de Grenoble, Professor]
Pierre François Dutot [Univ Grenoble Alpes, Associate Professor]
Gregory Mounie [Institut polytechnique de Grenoble, Associate Professor]
Olivier Richard [Univ Grenoble Alpes, Associate Professor]
Denis Trystram [Institut polytechnique de Grenoble, Professor, HDR]
Frederic Wagner [Institut polytechnique de Grenoble, Associate Professor]

PhD Students

Raphaël Bleuse [Univ Grenoble Alpes, until Oct 2017]
Danilo Carastan Dos Santos [Universidade Federal do ABC Brazil, from May 2017]
Estelle Dirand [CEA]
Adrien Faure [ATOS/BULL, from May 2017]
Mohammed Khatiri [Faculte des sciencesU.M.P, Morocco]
Alessandro Kraemer [Federal Technological University of Paraná, until Aug 2017]
Fernando Machado Mendonca [USP Brazil, until Apr 2017]
Michael Mercier [ATOS/Bull]
Clement Mommessin [Institut polytechnique de Grenoble, from Sep 2017]
Millian Poquet [Univ de Joseph Fourier]
Valentin Reis [Institut polytechnique de Grenoble]
Abhinav Srivastav [Univ Grenoble Alpes, until Feb 2017]
Julio Toss [UFRGS Brazil, until Aug 2017]
Salah Zrigui [Univ Grenoble Alpes,from Oct 2017]

Technical staff

Tristan Ezequel [Inria]
Nicolas Michon [Inria]
Pierre Neyron [CNRS]
Baptiste Pichot [Inria, until Sep 2017]
Theophile Terraz [Inria]

Interns

Lucas Barallon [Inria, from Feb 2017 until Jul 2017]
Youcef Djebour [Inria, from Feb 2017 until Jul 2017]
Konstantinos Dogeas [Institut polytechnique de Grenoble, from Feb 2017 until Jun 2017]
Vasilii Feofanov [Inria, until Jul 2017]
Thomas Lavocat [Inria, from Feb 2017 until Jul 2017]
Celia Manardo [Inria, from Mar 2017 until Jun 2017]
Sofiia Sandomirskaia [Inria, from Feb 2017 until Jun 2017]

Administrative Assistant

Annie Simon [Inria]

Visiting Scientist

Jorge Veiga Fachal [Universidade da Coruña Spain , from Apr 2017 until Jul 2017]

External Collaborators

Bruno Bzeznik [Univ de Joseph Fourier, until Sep 2017]
Christian Seguy [CNRS, until Oct 2017]

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 proposed DataMove team will work 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.

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 will address. We may also see in the coming

[1] Top500 Ranking, http://www.top500.org
[2] 10^{15} floating point operations per second
[3] SciDAC Review, 2010

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 will 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 will 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 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 (national supercomputers like Curie, or the BlueWaters machine accessible through our collaboration with Argonne National Lab). Conversely, a 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.

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 will make our new algorithms accessible through the team flagship software tools, **the OAR batch scheduler and the in-situ processing framework FlowVR**. We will maintain and enforce strong links with teams closely connected with large architecture design and operation (CEA DAM, BULL, Argonne National Lab), as well as scientists of other disciplines, in particular computational biologists, with whom we will elaborate and validate new usage scenarios (IBPC, CEA DAM, EDF).

3.3. Research Directions

DataMove research activity is organised around three directions. When a parallel job executes on a machine, it triggers data movements through the input data it needs to read, the results it produces (simulation results as well as traces) that need to be stored in the file system, as well as internal communications and temporary storage (for fault tolerance related data for instance). Modeling in details the simulation and the target machines to analyze scheduling policies is not feasible at large scales. We propose to investigate alternative approaches, including learning approaches, to capture and model the influence of data movements on the performance metrics of each job execution to develop **Data Aware Batch Scheduling** models and algorithms (Sec. 4.1). Experimenting new scheduling policies on real platforms at scale is unfeasible. Theoretical performance guarantees are not sufficient to ensure a new algorithm will actually perform as expected on a real platform. An intermediate evaluation level is required to probe novel scheduling policies. The second research axe focuses on the **Empirical Studies of Large Scale Platforms** (Sec. 4.2). The goal is to investigate how we could extract from actual computing centers traces information to replay the job allocations and executions on a simulated or emulated platform with new scheduling policies. Schedulers need information

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

about jobs behavior on target machines to actually be able to make efficient allocation decisions. Asking users to caracterize jobs often does not lead to reliable information. The third research direction **Integration of High Performance Computing and Data Analytics** (Sec. 4.3) addresses the data movement issue from a different perspective. New data analysis techniques on the HPC platform introduce new type of workloads, potentially more data than compute intensive, but could also enable to reduce data movements by directly enabling to pipe-line simulation execution with a live analysis of the produced results. Our goal is here to investigate how to program and schedule such analysis workflows in the HPC context.

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. 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 [5] 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.

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

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.

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. 6.4) 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 Sec. 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 (Sec. 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. Highlights of the Year

5.1. Highlights of the Year

5.1.1. Startup Company

Creation of the Ryax company [6] by two former PhD students, Yiannis Georgiou, David Glesser. Ryax Technologies builds software to enable the seamless execution of Big Data and IoT applications upon Hybrid computing infrastructures, distributed across Edge, Fog and Cloud environments. The core software named Ryax is a new generation resource manager.

5.1.2. Best Paper Nominee

Danilo Carastan-Santos, DataMove, Univ ABC, Brazil, was nominated for the Best Paper and Best Student Paper at Supercomputing 2017 for his paper *Obtaining Dynamic Scheduling Policies with Simulation and Machine Learning* [11].

6. New Software and Platforms

6.1. FlowVR

SCIENTIFIC 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.

The base entity, called a module or component, is an autonomous process, potentially multi-threaded with tools like OpenMP, TBB, or deferring computations to a GPU or Xeon Phi. This module processes data coming from input ports and write data on output ports. A module has no global insight on where the data comes from or goes to. The programming interface is designed to limit code refactoring, easing turning an existing code into a FlowVR component. The three main functions are:

wait(): Blocking function call that waits for the availability of new messages on input ports. get(): Retrieve a handle to access the message received at the previous wait() call on a given input port. put(): Notify FlowVR that a new message on a given output port is ready for dispatch. FlowVR manages data transfers. Intra-node communications between two components take place through a shared memory segment, avoiding copies. Once the sender has prepared the data in a shared memory segment, it simply handles a pointer to the destination that can directly access them. Inter-node communications extend this mechanism, FlowVR taking care of packing and transferring the data from the source shared memory segment to the destination shared memory segment.

Assembling components to build an application consists in writing a Python script, instanciate it according to the target machine. FlowVR will process it and prepare everything so that in one command line you can deploy and start your application.
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, Clément Ménier, Emmanuel Melin, Jean Denis Lesage, Jérémie Allard, Jérémy Jaussaud, Matthieu Dreher, Sébastien Limet, Sophie Robert and Valérie Gourantou
- Contact: Bruno Raffin
- URL: http://flowvr.sf.net

6.2. OAR

KEYWORDS: Resource manager - Clusters - Cloud - HPC - Light grid

[6] http://ryax-technologies.com/

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).

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

6.3. MELISSA

Modular External Library for In Situ Statistical Analysis
KEYWORD: Sensitivity Analysis
FUNCTIONAL DESCRIPTION: Melissa is an in situ solution for sensitivity analysis. It implements iterative algorithms to compute spatio-temporal statistic fields over results of large scale sensitivity studies. Melissa relies on a client/server architecture, composed of three main modules:

Melissa Server: an independent parallel executable. It receives data from the simulations, updates iterative statistics as soon as possible, then trow data away. Melissa API: a shared library to be linked within the simulation code. It mainly transmit simulation data to Melissa Server at each timestep. The simulations of the sensitivity analysis become the clients of Melissa Server. Melissa Launcher: A Python script in charge of generating and managing the whole global sensitivity analysis.

- Authors: Théophile Terraz, Bruno Raffin, Alejandro Ribes and Bertrand Iooss
- Partner: Edf
- Contact: Bruno Raffin
- Publications: In Situ Statistical Analysis for Parametric Studies - Melissa: Large Scale In Transit Sensitivity Analysis Avoiding Intermediate Files
- URL: https://melissa-sa.github.io

6.4. Platforms

6.4.1. 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).

7. New Results

7.1. Integration of High Performance Computing and Data Analytics

New results on the topic *Integration of High Performance Computing and Data Analytics* are related to compression [15], automatic data extraction for in situ processing [14], in transit sensitivity analysis [16] and management of heterogeneous HPC and BigData workloads [21]. We detail the two last here.

- **Large Scale In Transit Sensitivity Analysis Avoiding Intermediate Files [16].** Global sensitivity analysis is an important step for analyzing and validating numerical simulations. One classical approach consists in computing statistics on the outputs from well-chosen multiple simulation runs. Simulation results are stored to disk and statistics are computed postmortem. Even if supercomputers enable to run large studies, scientists are constrained to run low resolution simulations with a limited number of probes to keep the amount of intermediate storage manageable. In this paper we propose a file avoiding, adaptive, fault tolerant and elastic framework that enables high resolution global sensitivity analysis at large scale. Our approach combines iterative statistics and in transit processing to compute Sobol' indices without any intermediate storage. Statistics are updated on-the-fly as soon as the in transit parallel server receives results from one of the running simulations. For one experiment, we computed the Sobol' indices on 10M hexahedra and 100 timesteps, running 8000 parallel simulations executed in 1h27 on up to 28672 cores, avoiding 48TB of file storage. Based on this work we open sourced the associated framework called Melissa (https://melissa-sa.github.io).
- **Big Data and HPC collocation: Using HPC idle resources for Big Data Analytics [21]**. Executing Big Data workloads upon High Performance Computing (HPC) infrastractures has become an attractive way to improve their performances. However, the collocation of HPC and Big Data workloads is not an easy task, mainly because of their core concepts' differences. This paper focuses on the challenges related to the scheduling of both Big Data and HPC workloads on the same computing platform. In classic HPC workloads, the rigidity of jobs tends to create holes in the schedule: we can use those idle resources as a dynamic pool for Big Data workloads. We propose a new idea based on Resource and Job Management System's (RJMS) configuration, that makes HPC and Big Data systems to communicate through a simple prolog/epilog mechanism. It leverages the built-in resilience of Big Data frameworks, while minimizing the disturbance on HPC workloads. We present the first study of this approach, using the production RJMS middleware OAR and Hadoop YARN from the HPC and Big Data ecosystems respectively. Our new technique is evaluated with real experiments upon the Grid5000 platform. Our experiments validate our assumptions and show promising results. The system is capable of running an HPC workload with 70% cluster utilization, with a Big Data workload that fills the schedule holes to reach a full 100% utilization. We observe a penalty on the mean waiting time for HPC jobs of less than 17% and a Big Data effectiveness of more than 68% in average.

7.2. Data Aware Batch Scheduling

New results on the topic *Data Aware Batch Scheduling* are related to graph algorithm for dense k-subset detection [8], scheduling heuristic for multi-CPU multi-GPU computing platform with performance guarantee [9], machine learning for designing scheduling policies [11] and multi-objective scheduling heuristic [13]. We detail the two last here.

- **Obtaining Dynamic Scheduling Policies with Simulation and Machine Learning [11].** Dynamic scheduling of tasks in large-scale HPC platforms is normally accomplished using ad-hoc heuristics, based on task characteristics, combined with some backfilling strategy. Defining heuristics that work efficiently in different scenarios is a difficult task, specially when considering the large variety of task types and platform architectures. In this work, we present a methodology based on simulation and machine learning to obtain dynamic scheduling policies. Using simulations and a workload generation model, we can determine the characteristics of tasks that lead to a reduction in the mean

slowdown of tasks in an execution queue. Modeling these characteristics using a nonlinear function and applying this function to select the next task to execute in a queue dramatically improved the mean task slowdown in synthetic workloads. When applied to real workload traces from highly different machines, these functions still resulted in important performance improvements, attesting the generalization capability of the obtained heuristics.

- **A new on-line method for scheduling independent tasks [13].** We present a new method for scheduling independent tasks on a parallel machine composed of identical processors. This problem has been studied extensively for a long time with many variants. We are interested here in designing a generic algorithm in the on-line non-preemptive setting whose performance is good for various objectives. The basic idea of this algorithm is to detect some problematic tasks that are responsible for the delay of other shorter tasks. Then the former tasks are redirected to be executed in a dedicated part of the machine. We show through an extensive experimental campaign that this method is effective and in most cases is closer to some standard lower bounds than the base-line method for the problem.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

- **BULL-ATOS SE (2016-2019)**. Two PhD grants (Michael Mercier and Adrien Faure). Job and resource management algorithms.
- **CEA DAM (2016-2018)**. PhD grant support contract (PhD of Estelle Dirand, funded by CEA). In situ analysis for Molecular Simulations.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. ANR

- **ANR grant MOEBIUS (2013-2017).** Multi-objective scheduling for large computing platforms. Coordinator: Grenoble-INP (DataMove). Partners: Grenoble-INP, Inria, BULL-ATOS .
- **ANR grant GRECO (2017-2020).** Resource manager for cloud of things. Coordinator: Quarnot Computing. Partners: Grenoble-INP, Inria,

9.1.2. 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).

9.1.3. Inria

- Inria PRE COSMIC (exploratory research project), 2017-2019. Photovoltaic Energy Management for Distributed Cloud Platforms. Myriads, DataMove.

9.2. International Initiatives

9.2.1. Inria International Labs

9.2.1.1. 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).

9.2.2. Participation in Other International Programs

9.2.2.1. 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 parternship 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.

9.3. International Research Visitors

9.3.1. Visits of International Scientists

PhD in progress: Danilo Carastan Dos Santos, Dynamic Scheduling of Tasks in High Performance Platforms with Machine Learning (Sao Paulo, Brasil). 1 year "sandwich" visit. Local adviser: Denis Trystram

PhD in progress: Jorge Veiga Fachal, High Performance Map-Reduce, Universidade da Coruña, Spain. 3 month stay. Local adviser: Bruno Raffin.

9.3.2. Visits to International Teams

Yes Denneulin spent 3 months at University of Los Andes, Bogota, Columbia.

PhD in progress: Clement Mommessin spent 6 months at ANL, Argonne, USA. Adviser: Tom Perterka.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. General Chair, Scientific Chair

President of the steering committee of Edu-Europar.

President of the steering committee of EGPGV (Eurographics Symposium on Parallel Graphics and Visualization).

Member of the steering committee of Europar.

Member of the steering committee of *Journée de visualisation scientifique*.

Member of the steering committee of HeteroPar.

10.1.1.2. Member of the Organizing Committees

Euro-Par, ,Santiago de Compostela, Spain, August 2017. Topic chairs.

ICPP (46th Internat. Conference on Parallel Processing), August 14-17, Bristol, UK. Track chair.

10.1.2. Scientific Events Selection

10.1.2.1. Member of the Conference Program Committees

ISAV 2017 (Workshop on In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization) , November 2017, Denver, USA

EGPGV 2017 (Eurographics Symposium on Parallel Graphics and Visualization), June 2017, Barcelona, Spain.

LADV 2017 (IEEE Symposium on Large Data Analysis and Visualization), October 2017, Phoenix, USA.

ROADEF, Feb. 22-24, Metz, France

CCgrid (17th IEEE/ACM Internat. Symposium on Cluster, Cloud and Grid Computing) May 14-17, Madrid, Spain

COMPAS, June 27-30, Sophia-Antipolis, France

ISPDC (16th Internat Symposium on Parallel and Distributed Computing) July 3-6, Innsbruck, Austria

WAOA (15th workshop on Approximation and Oline Algorithms) Sept. 7-8, Vienna, Austria

PPAM (12th internat. conf. on Parallel Processing and Applied Maths) Sept. 10-13, Lublin, Poland

ParCo, Sept. 12-15, Bologna, Italy

CloudTech (3rd International Conference on Cloud Computing Technologies and Applications) Oct. 24-26, Rabat, Morocco

MISTA (8th edition) Dec. 5-8, Kuala Lumpur, Malaysia

HiPC (23rd IEEE internat. Conf on High Performance Computing, Data and Analytics) Dec. 18-21, Jaipur, India

10.1.3. Journal

10.1.3.1. 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).

10.1.4. Scientific Expertise

ANR project evaluation expert

Nederlands e-science center expert

10.1.5. Research Administration

Director of Pôle MSTIC of COMUE Univ. Grenoble-Alpes.

Steering committee of Grid'5000

Steering committee of GRICAD

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

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

Master: 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).

Master: Yves Denneulin. 70 hours per year. Engineering school ENSIMAG, Grenoble-INP (M1/2nd year and M2/3rd year).

10.2.2. Supervision

PhD: Raphaël Bleuse, Affinity Scheduling, Defended November 2017, Univ. Grenoble-Alpes. Adviser: Denis Trystram and Gregory Mounié.

PhD: Millian Poquet, Energy consumption optimization for high performance computing, Defended December 2017, Univ. Grenoble-Alpes. Advisers: Denis Trystram and Pierre-François Dutot

PhD: Abhinav Srivastav, Multi-objective Scheduling, Defended May 2017, Univ. Grenoble-Alpes. Advisers: Denis Trystram and Oded Maler

PhD: Fernando Machado Mendonca, Locality Aware Scheduling, Univ. Grenoble-Alpes, Defended April 2017, Advisers: Frederic Wagner and Denis Trystram.

PhD: Julio Toss, Parallel Algorithms and Data Structures for Physically Based Simulation of Deformable Objects, Univ. Grenoble-Alpes and UFRGS (co-tutelle). Defended October 2015. 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: Valentin Reis, Machine Learning for resource management, Univ. Grenoble-Alpes. Started October 2015. Advisers: Denis Trystram and Eric Gaussier

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: 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, Morocco)

PhD in progress: Adrien Faure, Advisers: Denis Trystram

PhD in progress: Clément Mommessin, Advisers: Denis Trystram

10.2.3. Juries

PhD Defense of Jorge F. Fabeiro, Tools for Improving Performance Portability in Heterogeneous Environments, 13th of July 2017. Univesidade da Coruña, Spain. Jury President.

PhD Defense Xavier Martinez, Tracking sans marqueur de modèles physiques modulaires et articulés: vers une interface tangible pour la manipulation de simulations moléculaires, 10th of October 2017. Universite´ Paris-Sud, Paris Saclay. Jury Member.

PhD defense Jérôme Richard, Conception d'un mode`le de composants logiciels avec ordonnancement de taˆches pour les architectures paralle`les multi-cœurs, application au code Gysela, 6th of December 2017. Université de Lyon. Jury President

PhD defense Sébastien Doutreligne, Interactive Molecular Dynamics Software Development: applications to biomolecule folding, 27th of October 2017. UPMC. Reviewer.

PhD defense Suraj Kumar, Scheduling of Dense Algebra Kernels on Heterogeneous Resources , Avril 2017. Univ of Bordeaux . Jury President.

10.3. Popularization

Responsible of the workshop *Proof without Words* and participation to the *CS-Unplugged* workshop, Inria, Fête de la Science 2017.

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] R. BLEUSE. *Affinity Scheduling*, Univ. Grenoble-Alpes, 2017

[2] F. M. MENDONCA. *Locality Aware Scheduling*, Univ. Grenoble-Alpes, 2017

[3] M. POQUET. *Energy consumption optimization for high performance computing*, Univ. Grenoble-Alpes, 2017

[4] A. SRIVASTAV. *Multi-objective Scheduling*, Univ. Grenoble-Alpes, 2017

[5] J. TOSS. *Parallel Algorithms and Data Structures for Physically Based Simulation of Deformable Objects*, Univ. Grenoble-Alpes and UFRGS, 2017

Articles in International Peer-Reviewed Journals

[6] S. ALBERS, E. BAMPIS, D. LETSIOS, G. LUCARELLI, R. STOTZ. *Scheduling on power-heterogeneous processors*, in "Information and Computation", December 2017, vol. 257, pp. 22 - 33 [*DOI :* 10.1016/J.IC.2017.09.013], https://hal.inria.fr/hal-01668736

[7] R. BLEUSE, S. HUNOLD, S. KEDAD-SIDHOUM, F. MONNA, G. MOUNIÉ, D. TRYSTRAM. *Scheduling Independent Moldable Tasks on Multi-Cores with GPUs*, in "IEEE Transactions on Parallel and Distributed Systems", 2017, 14 p. [*DOI :* 10.1109/TPDS.2017.2675891], https://hal.inria.fr/hal-01516752

[8] N. BOURGEOIS, A. GIANNAKOS, G. LUCARELLI, I. MILIS, V. T. PASCHOS. *Exact and superpolynomial approximation algorithms for the densest k-subgraph problem*, in "European Journal of Operational Research", 2017, vol. 262, pp. 894 - 903 [*DOI :* 10.1016/J.EJOR.2017.04.034], https://hal.inria.fr/hal-01539561

[9] S. KEDAD-SIDHOUM, F. MONNA, G. MOUNIÉ, D. TRYSTRAM. *A Family of Scheduling Algorithms for Hybrid Parallel Platforms*, in "International Journal of Foundations of Computer Science", 2017, forthcoming, http://hal.upmc.fr/hal-01516700

International Conferences with Proceedings

[10] M. AMARIS, G. LUCARELLI, C. MOMMESSIN, D. TRYSTRAM. *Generic algorithms for scheduling applications on hybrid multi-core machines*, in "23rd International European Conference on Parallel and Distributed Computing (EuroPar 2017)", Santiago de Compostela, Spain, August 2017, https://hal.inria.fr/hal-01420798

[11] D. CARASTAN-SANTOS, R. Y. DE CAMARGO. *Obtaining Dynamic Scheduling Policies with Simulation and Machine Learning*, in "SC'17 -2 International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing)", Denver, United States, November 2017, https://hal.inria.fr/hal-01618940

[12] P.-F. DUTOT, Y. GEORGIOU, D. GLESSER, L. LEFÈVRE, M. POQUET, I. RAÏS. *Towards Energy Budget Control in HPC*, in "17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing", Madrid, Spain, Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017, Madrid, Spain, May 14-17, 2017, May 2017, n° 17, pp. 381-390, https://hal.archives-ouvertes.fr/hal-01533417

[13] G. LUCARELLI, F. MACHADO MENDONCA, D. TRYSTRAM. *A new on-line method for scheduling independent tasks*, in "17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2017)", Madrid, Spain, May 2017, https://hal.inria.fr/hal-01527746

[14] C. MOMMESSIN, M. DREHER, T. PETERKA, B. RAFFIN. *Automatic Data Filtering for In Situ Workflows*, in "IEEE International Conference on Cluster Computing", Hawai, United States, September 2017, https://hal.inria.fr/hal-01581032

[15] K. SCHARNOWSKI, S. FREY, B. RAFFIN, T. ERTL. *Spline-based Decomposition of Streamed Particle Trajectories for Efficient Transfer and Analysis*, in "EuroVis'17 - Proceedings of the 19th EG/VGTC Conference on Visualization", Barcelone, Spain, June 2017, 4 p. [*DOI :* 10.2312/EGSH.20171010], https://hal.inria.fr/hal-01529371

[16] T. TERRAZ, A. RIBES, Y. FOURNIER, B. IOOSS, B. RAFFIN. *Melissa: Large Scale In Transit Sensitivity Analysis Avoiding Intermediate Files*, in "The International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing)", Denver, United States, November 2017, pp. 1 - 14, https://hal.inria.fr/hal-01607479

[17] L. YALA, P. FRANGOUDIS, G. LUCARELLI, A. KSENTINI. *Balancing between cost and availability for CDNaaS resource placement*, in "IEEE Global Communications Conference (GLOBECOM 2017)", Singapore, Singapore, December 2017, https://hal.inria.fr/hal-01590885

Conferences without Proceedings

[18] R. BLEUSE, G. LUCARELLI, G. MOUNIÉ, D. TRYSTRAM. *Interference-Aware Scheduling with 2D-Torus as a Case Study*, in "Joint EURO/ORSC/ECCO Conference 2017 on Combinatorial Optimization, ECCO2017", Koper, Slovenia, May 2017, http://hal.univ-grenoble-alpes.fr/hal-01669062

[19] J. LELONG, V. REIS, D. TRYSTRAM. *Tuning EASY-Backfilling Queues*, in "21st Workshop on Job Scheduling Strategies for Parallel Processing", Orlando, United States, 31st IEEE International Parallel & Distributed Processing Symposium, May 2017, https://hal.archives-ouvertes.fr/hal-01522459

[20] G. LUCARELLI, K. NGUYEN, A. SRIVASTAV, D. TRYSTRAM. *Online Min-Sum Flow Scheduling with Rejections*, in "The 13th Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP 2017)", Seeon Abbey, Germany, July 2017, https://hal.archives-ouvertes.fr/hal-01672351

[21] M. MERCIER, D. GLESSER, Y. GEORGIOU, O. RICHARD. *Big Data and HPC collocation: Using HPC idle resources for Big Data Analytics*, in "IEEE BigData 2017", Boston, United States, December 2017, https://hal.archives-ouvertes.fr/hal-01633507

[22] P. NEYRON, L. NUSSBAUM. *Resources management on the Grid'5000 testbed*, in "GEFI 17 meeting - Global Experimentation for Future Internet", Rio de Janeiro, Brazil, October 2017, https://hal.inria.fr/hal-01626320

Other Publications

[23] P. NEYRON, B. BZEZNIK, L. NUSSBAUM. *Propositions pour l'architecture pour un cluster mutualisé entre CIMENT et Grid'5000*, January 2017, working paper or preprint, https://hal.inria.fr/hal-01511285

[24] P. NEYRON. *Le projet HPCDA@UGA*, October 2017, pp. 1-12, Journées SUCCES 2017 - Rencontre Scientifiques des Utilisateurs de Calcul intensif, de Cloud Et de Stockage, https://hal.inria.fr/hal-01618946