



IN PARTNERSHIP WITH:
CNRS

**Institut polytechnique de
Grenoble**

Université de Grenoble Alpes

Activity Report 2018

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

1. Team, Visitors, External Collaborators	1
2. Overall Objectives	2
3. Research Program	3
3.1. Motivation	3
3.2. Strategy	3
3.3. Research Directions	4
4. Application Domains	5
4.1. Data Aware Batch Scheduling	5
4.1.1. Algorithms	5
4.1.2. Locality Aware Allocations	5
4.1.3. Data-Centric Processing	6
4.1.4. Learning	7
4.1.5. Multi-objective Optimization	7
4.2. Empirical Studies of Large Scale Platforms	8
4.2.1. Workload Traces with Resource Consumption	8
4.2.2. Simulation	9
4.2.3. Job and Platform Models	9
4.2.4. Emulation and Reproducibility	9
4.3. Integration of High Performance Computing and Data Analytics	10
4.3.1. Programming Model and Software Architecture	10
4.3.2. Resource Sharing	11
4.3.3. Co-Design with Data Scientists	12
5. New Software and Platforms	12
5.1. FlowVR	12
5.2. OAR	13
5.3. MELISSA	13
5.4. Platforms	14
6. New Results	14
6.1. Integration of High Performance Computing and Data Analytics	14
6.1.1. I/O Survey	14
6.1.2. Task Based In Situ Processing	14
6.1.3. Stream Processing	15
6.1.3.1. Packed Memory QuadTree.	15
6.1.3.2. Flink based in situ Processing.	15
6.2. Data Aware Batch Scheduling	16
6.2.1. Batch Scheduling for Energy	16
6.2.2. Learning Methods for Batch Scheduling	16
6.2.3. Reproducibility	16
6.2.4. Online Algorithms	17
7. Bilateral Contracts and Grants with Industry	17
8. Partnerships and Cooperations	17
8.1. National Initiatives	17
8.1.1. ANR	17
8.1.2. Competitvity Clusters	18
8.1.3. Inria	18
8.2. European Initiatives	18
8.3. International Initiatives	18
8.3.1. Inria International Labs	18
8.3.2. Participation in Other International Programs	19

8.4. International Research Visitors	19
9. Dissemination	19
9.1. Promoting Scientific Activities	19
9.1.1. Scientific Events Organisation	19
9.1.1.1. General Chair, Scientific Chair	19
9.1.1.2. Member of the Organizing Committees	20
9.1.2. Scientific Events Selection	20
9.1.3. Journal	20
9.1.4. Scientific Expertise	20
9.1.5. Research Administration	20
9.2. Teaching - Supervision - Juries	20
9.2.1. Teaching	20
9.2.2. Supervision	21
9.2.3. Juries	21
9.3. Popularization	22
10. Bibliography	22

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:

- B1.1.7. - Bioinformatics
- B3.3.2. - Water: sea & ocean, lake & river
- B5.5. - Materials

1. Team, Visitors, External Collaborators

Research Scientists

- Fanny Dufosse [Inria, Researcher]
- Bruno Raffin [Team leader, Inria, Senior Researcher, HDR]
- Francieli Zanon Boito [Inria, Researcher, from Nov 2018]

Faculty Members

- Yves Denneulin [Institut polytechnique de Grenoble, Professor, HDR]
- Pierre Francois 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]
- Giorgio Lucarelli [Institut polytechnique de Grenoble, Associate Professor, until Aug 2018]

PhD Students

- Danilo Carastan Dos Santos [Universidade Federal do ABC Brazil, from May 2017]
- Estelle Dirand [CEA, until Oct 2018]
- Adrien Faure [ATOS/BULL, from May 2017]
- Mohammed Khatiri [Faculte des sciences U.M.P, Morocco]
- Michael Mercier [ATOS/Bull - Inria, until Jan 2019]
- Clement Mommessin [Institut polytechnique de Grenoble]
- Valentin Reis [Institut polytechnique de Grenoble, until Aug 2018]
- Salah Zrigui [Univ Grenoble Alpes]

Technical staff

- Tristan Ezequel [Inria, until Aug 2018]
- Nicolas Michon [Inria, until Sep 2018]
- Clement Labastrou [Inria, from Mar 2018 until Aug 2018]
- Pierre Neyron [CNRS]

Theophile Terraz [Inria]

Interns

Zulhusni Abdul Rahman [Inria, from Feb 2018 until Jun 2018]

Sekina Belguendouz [Inria, from May 2018 until Jul 2018]

Henrique Colao Zanuz [Inria, from Feb 2018 until Jun 2018]

Konstantinos Dogeas [Institut polytechnique de Grenoble, from Feb 2018 until Aug 2018]

Antoine Duquennoy [Inria, until Jun 2018]

Pathikrit Ghosh [Inria, until Jul 2018]

Carlos Eduardo Gomez Hernandez [Inria, from Apr 2018 until Aug 2018]

Florian Groguelin [Inria, until Jun 2018]

Quentin Guilloteau [Inria, Jun 2018]

Ciceron Kapena Bantu Muetu [Institut polytechnique de Grenoble, until Jun 2018]

Tuan Hiep Pham [Inria, from Feb 2018 until Aug 2018]

Saurabh Manish Raje [Inria, from Sep 2018]

Ning Tang [Inria, until Jul 2018]

Dinh Khoa Tran [Inria, until Jun 2018]

Tuo Zhao [Inria, until Jun 2018]

Administrative Assistant

Annie Simon [Inria]

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-fed 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 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.

¹Top500 Ranking, <http://www.top500.org>

²10¹⁵ floating point operations per second

³SciDAC Review, 2010

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

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-fed 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 about jobs behavior on target machines to actually be able to make efficient allocation decisions. Asking users to characterize 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 themselves 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 staging 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 arbitrarily 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 SWF⁵ 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.

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

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 platforms 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. 5.4) 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 foundations 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 thinner 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 better 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. New Software and Platforms

5.1. 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, instantiate 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>

5.2. 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).

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

5.3. 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 throw 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>

5.4. Platforms

5.4.1. 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).

6. New Results

6.1. Integration of High Performance Computing and Data Analytics

6.1.1. I/O Survey

First contribution is a comprehensive survey on parallel I/O in the HPC context [14]. As the available processing power and amount of data increase, I/O remains a central issue for the scientific community. This survey focuses on a traditional I/O stack, with a POSIX parallel file system. Through the comprehensive study of publications from the most important conferences and journals in a five-year time window, we discuss the state of the art of I/O optimization approaches, access pattern extraction techniques, and performance modeling, in addition to general aspects of parallel I/O research. This survey enables us to identify the general characteristics of the field and the main current and future research topics.

6.1.2. Task Based In Situ Processing

One approach to bypass the I/O bottleneck is *in situ* processing, an important research topic at DataMove. The *in situ* paradigm proposes to reduce data movement and to analyze data while still resident in the memory of the compute node by co-locating simulation and analytics on the same compute node. The simplest approach consists in modifying the simulation timeloop to directly call analytics routines. However, several works have shown that an *asynchronous* approach where analytics and simulation run concurrently can lead to a significantly better performance. Today, the most efficient approach consists in running the analytics processes on a set of dedicated cores, called helper cores, to isolate them from the simulation processes. Simulation and analytics thus run concurrently on different cores but this static isolation can lead to underused resources if the simulation or the analytics do not fully use all the assigned cores.

In this work performed in collaboration with CEA, we developed TINS, a task-based in situ framework that implements a novel *dynamic helper core* strategy. TINS relies on a work stealing scheduler and on task-based programming. Simulation and analytics tasks are created concurrently and scheduled on a set of worker threads created by a single instance of the work stealing scheduler. Helper cores are assigned dynamically: some worker threads are dedicated to analytics when analytics tasks are available while they join the other threads for processing simulation tasks otherwise, leading to a better resource usage. We leverage the good compositionality properties of task-based programming to seamlessly keep the analytics and simulation codes well separated and a plugin system enables to develop parallel analytics codes outside of the simulation code.

TINS is implemented with the Intel Threading Building Blocks (TBB) library that provides a task-based programming model and a work stealing scheduler. The experiments are conducted with the hybrid MPI+TBB ExaStamp molecular dynamics code that we associate with a set of analytics representative of computational physics algorithms. We show up to 40% performance improvement over various other approaches, including the standard helper core, on experiments on up to 14,336 Broadwell cores.

6.1.3. Stream Processing

Stream processing is the Big Data equivalent of in situ processing. It consists in analyzing on-line incoming streams of data, often produced from sensors or social networks like Twitter. We investigated the convergence between both paradigms through different directions: how the programming environment developed specifically for stream processing can be applied to the data produced by large parallel simulations [18]; Proposing a dynamics data structure to keep sorted data streams [12]; Evaluating the performance of the FlameMR framework on data produced from a parallel simulation [13]. We summarize here the 2 first contributions.

6.1.3.1. Packed Memory QuadTree.

Over the past years, several in-memory big-data management systems have appeared in academia and industry. In-memory databases systems avoid the overheads related to traditional I/O disk-based systems and have made possible to perform interactive data-analysis over large amounts of data. A vast literature of systems and research strategies deals with different aspects, such as the limited storage size and a multi-level memory-hierarchy of caches. Maintaining the right data layout that favors locality of accesses is a determinant factor for the performance of in-memory processing systems. Stream processing engines like Spark or Flink support the concept of *window*, which collects the latest events without a specific data organization. It is possible to trigger the analysis upon the occurrence of a given criterion (time, volume, specific event occurrence). After a window is updated, the system shifts the processing to the next batch of events. There is a need to go one step further to keep a live window continuously updated while having a fine grain data replacement policy to control the memory footprint. The challenge is the design of dynamic data structures to absorb high rate data streams, stash away the oldest data to stay in the allowed memory budget while enabling fast queries executions to update visual representations. A possible solution is the extension of database structures like R-trees used in Spatialite or PostGis, or to develop dedicated frameworks like Kit based on a pyramid structure.

We developed a novel self-organized cache-oblivious data structure, called PMQ, for in-memory storage and indexing of fixed length records tagged with a spatiotemporal index. We store the data in an array with a controlled density of gaps (*i.e.*, empty slots) that benefits from the properties of the *Packed Memory Arrays*. The empty slots guarantee that insertions can be performed with a low amortized number of data movements ($O(\log^2(N))$) while enabling efficient spatiotemporal queries. During insertions, we rebalance parts of the array when required to respect density constraints, and the oldest data is stashed away when reaching the memory budget. To spatially subdivide the data, we sort the records according to their Morton index, thus ensuring spatial locality in the array while defining an implicit, recursive quadtree, which leads to efficient spatiotemporal queries. We validate PMQ for consuming a stream of tweets to answer visual and range queries. PMQ significantly outperforms the widely adopted spatial indexing data structure R-tree, typically used by relational databases, as well as the conjunction of Geohash and B^+ -tree, typically used by NoSQL databases.

6.1.3.2. Flink based in situ Processing.

We proposed to leverage Apache Flink, a scalable stream processing engine from the Big Data domain, in this HPC context. Flink enables to program analyses within a simple window based map/reduce model, while

the runtime takes care of the deployment, load balancing and fault tolerance. We build a complete in transit analytics workflow, connecting an MD simulation to Apache Flink and to a distributed database, Apache HBase, to persist all the desired data. To demonstrate the expressivity of this programming model and its suitability for HPC scientific environments, two common analytics in the Molecular Dynamics field have been implemented. We assessed the performance of this framework, concluding that it can handle simulations of sizes used in the literature while providing an effective and versatile tool for scientists to easily incorporate on-line parallel analytics in their current workflows.

6.2. Data Aware Batch Scheduling

6.2.1. Batch Scheduling for Energy

The project COSMIC [24], [22], [16], [17], in collaboration with Myriads team in Inria Rennes-Atlantique, targets the optimization of green energy usage in Clouds. The project considers a geographically distributed cloud, with each data center associated with a local photovoltaic (PV) farm. The objective is to maximize the photovoltaic energy by allocation the computing workload to the data centers according to its energy production. The production forecasting is modeled with a truncated normal law, permitting to consider the uncertainty of the forecast.

Chapter [24] considers a simple model with homogeneous Virtual Machines submitted at unpredictable rate. This study has resulted in a scheduling algorithm for task allocation. The chapter demonstrates the optimality of this algorithm at current time slot according to production forecast parameters.

Paper [22] extends these results to heterogeneous VM. Each VM is defined by its arrival date, its execution time, its memory requirement and its CPU usage. In this model, due to execution time durations, the possibility to migrate running VM was considered. An algorithm is detailed in the paper that is compared to standard algorithm through simulations.

A third study [16], [17] has carefully modeled the interactions between the Cloud and the energy supplier. Due to variability of PV production and workload submission, each data center will alternatively inject energy into the electricity grid or purchase energy. The energy model considers a virtual energy pool mitigating the surplus and deficit of the different data center, with reduced costs regarding the difference between electricity cost and electricity injection tariff. The algorithm detailed in this paper outperforms well-known round-robin approaches, as shown by simulations.

6.2.2. Learning Methods for Batch Scheduling

Most of Job Scheduling algorithms apply greedy tasks ordering, as First Come First Served (FCFS) or Shortest Processing time First (SPF). They give simple methods, highly practical with certain guarantees. They are however far from optimal. Mixed methods, combining many of this basic methods permit to improve their performance. DataMove has developed [27] a learning method permitting to adapt the Mixed method to benchmarks. An extensive experimental campaign has permitted to determine the possibilities of basic and mixed methods according to the benchmarks characteristics, enhancing the efficiency of mixed methods.

6.2.3. Reproducibility

Related to batch scheduling experimentation, DataMove has led investigations on reproducibility [23]. Existing approaches focus on repeatability, but this is only the first step to reproducibility: Continuing a scientific work from a previous experiment requires to be able to modify it. This ability is called reproducibility with Variation. We show that capturing the environment of execution is necessary but not sufficient ; we also need the environment of development. The variation also implies that those environments are subject to evolution, so the whole software development lifecycle needs to be considered. To take into account these evolutions, software environments need to be clearly defined, reconstructible with variation, and easy to share. In this context, we propose new way of seeing reproducibility through the scientific software development lifecycle. Each step in this lifecycle requires a software environment. We define a software environment by a set of applications and libraries, with all their dependencies, and their configurations, required to achieve a step in a scientific workflow.

6.2.4. Online Algorithms

Rob van Stee wrote a review of 2018 online algorithms including our recent contributions on resource augmentation ⁶ We quote him here:

Progress was also made on scheduling to minimize weighted flow time on unrelated machines. In ESA 2016, Giorgio Lucarelli et al. [1] had considered a version where the online algorithm can reject some $\varepsilon_r > 0$ fraction (by weight) of the jobs and have machines that are $1 + \varepsilon_s$ as fast as the offline machines, for some $\varepsilon_s > 0$. They showed that this is already enough to achieve a competitive ratio of $O(1/(\varepsilon_s \varepsilon_r))$.

In SPAA 2018, Giorgio Lucarelli et al.[20] (a superset of the previous authors) showed that it is in fact sufficient to reject a 2ε fraction of the total number of jobs to achieve a competitive ratio of $2(\frac{1+\varepsilon}{\varepsilon})$ for minimizing the total flow time. This algorithm sometimes rejects a job other than the one that has just arrived. The authors show that this is necessary, as otherwise there is a lower bound of $\Omega(\Delta)$ even on a single machine. Here Δ is the size ratio (the ratio of largest to smallest job size). (Obviously this lower bound also holds if you cannot reject jobs at all.)

They also consider the speed scaling model, in which machines can be sped up if additional energy is invested, and the goal is to minimize the total weighted flow time plus energy usage. If the power function of machine i is given by $P(s_i(t)) = s_i(t)^\alpha$, where $s_i(t)$ is the current speed of machine i , there is an algorithm which is $O((1 + 1/\varepsilon)^{\alpha/(\alpha-1)})$ -competitive that rejects jobs of total weight at most a fraction ε of the total weight of all the jobs. They also give a positive result for jobs with hard deadlines, where the goal is to minimize the total energy usage and no job may be rejected.

In ESA 2018, the same set of authors [11] improved/generalized these results by showing that rejection alone is sufficient for an algorithm to be competitive even for weighted flow time. They presented an $O(1/\varepsilon^3)$ -competitive algorithm that rejects at most $O(\varepsilon)$ of the total weight of the jobs. In this algorithm, jobs are assigned (approximately) greedily to machines, and each machine runs the jobs assigned to it using Highest Density First. A job may be rejected if it is running while much heavier jobs arrive or if it is in the queue while very many jobs arrive. The second rule simulates the resource augmentation on the speed.

7. Bilateral Contracts and Grants with Industry

7.1. 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.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

- **ANR grant GRECO (2017-2020)**. Resource manager for cloud of things. Coordinator: Quarnot Computing. Partners: Quarnot Computing, Grenoble-INP, Inria.
- **ANR grant Energumen (2018-2022)**. Resource management: malleable jobs for a better use of the resources along with energy optimization. Coordinator: Denis Trystram. Partners: Grenoble-INP, IRIT, Sorbonne Université.

⁶Rob van Stee. 2018. SIGACT News Online Algorithms Column 34: 2018 in review. SIGACT News 49, 4 (December 2018), 36-45.

8.1.2. Competitiveness Clusters

- **PIA Avido (2015-2018)**. In situ analysis and visualization for large scale numerical simulation. Coordinator: EDF SA. Partners: EDF R&D, Total SA, Kitware SAS , Université Pierre et Marie CURIE, Inria (DataMove).
- **FUI OverMind (2015-2018)**. Task planification and asset management for the cartoon productions. Coordinator: Teamto Studio. Partners: Teamto Studio, Folimage Studio, Ecole de Gobelins, Inria (DataMove).
- **FUI IDIOM (2018-2020)**. Monitoring and optimization of I/Os. Coordinator DDN Storage. Partners: DDN Storage, Criteo, Quarnot, QuasarDB, CEA, Université de Bretagne Occidentale, Telecom SudParis, Inria (DataMove).

8.1.3. Inria

- Inria PRE COSMIC (exploratory research project), 2017-2019. Photovoltaic Energy Management for Distributed Cloud Platforms. Myriads, DataMove.
- Inria IPL HPC-BigData, 2018-2021). Convergence between HPC, Big Data and AI. Coordinator: Bruno Raffin. Partners: the Inria teams Zenith, Kerdata, Datamove, Tadaam, SequeL, Parietal, Tau, and the external partners ATOS, ANL, IBPC, ESI-Group. See <https://project.inria.fr/hpcbigdata/>

8.2. European Initiatives

8.2.1. Collaborations in European Programs, Except FP7 & H2020

Program: SKŁODOWSKA-CURIE ACTIONS - Individual Fellowship

Project acronym: DAMA

Project title: Extreme-Scale Data Management

Duration: November 2018 - October 2020

Coordinator: Bruno Raffin

Fellowship Recipient: Francieli Zanon Boito.

Abstract: This project is concerned with the I/O challenges that arise from the convergence between these two different paradigms. It is clear data analytics tools cannot simply replace their typical storage solutions for the HPC I/O stack, centered on the abstraction of files and powered by a parallel file system, because their workload is not well suited for that and would observe poor performance. Moreover, the separated storage infrastructure breaks the data affinity idea in which they are built upon. Finally, even among traditional HPC applications there is a need to minimize data movement, as it imposes high latency and increases energy consumption.

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).

8.3.2. Participation in Other International Programs

8.3.2.1. 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 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.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Professor visit: Alba Cristina Magalhaes Alves De Melo, Professor at University of Brasilia, visited the Datamove for one month in 2018.

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

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Organisation

9.1.1.1. 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.

Co-chair of the First HPML Workshop collocated with SBAC-PAD Lyon , October 2018.

9.1.1.2. Member of the Organizing Committees

Euro-Par Advisory Board Member

9.1.2. Scientific Events Selection

9.1.2.1. Member of the Conference Program Committees

ISAV 2018 (Workshop on In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization), November, Dallas, USA

EGPGV 2018 (Eurographics Symposium on Parallel Graphics and Visualization), June, Brno, Czech Republic.

LADV 2018 (IEEE Symposium on Large Data Analysis and Visualization), October, Berlin, Germany.

IPDPS 2018 (International Parallel and Distributed Processing Symposium), May, Vancouver, Canada.

LATIN, April 2018, Buenos Aires, Argentina.

CCgrid 2018 (18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing), May, Washington DC.

SPAA (30th ACM Symposium on Parallelism in Algorithms and Architecture), July 2018, Vienna, Austria.

ISPDC, June 2018, Geneva, Switzerland.

CONPAS (Conference d'informatique en Parallelisme, Architecture et Systeme), July 2018, Toulouse, France.

HiPC 2018 (IEEE internat. Conf on High Performance Computing, Data and Analytics), December 2018, Bengaluru, India.

EURO 2018 (29th European Conference On Operational Research), July, Valencia.

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 Computational Methods in Science and Technology.

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

9.1.4. Scientific Expertise

ANR project evaluation expert

9.1.5. Research Administration

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

Head of the international Master program (MOSIG1) at Grenoble INP (ENSIMAG)

Steering committee of Grid'5000

Steering committee of GRICAD

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). 200 hours per year in average.

Master: Fanny Dufossé. 30 hours per year. Combinatorial scientific computing in Master at ENS Lyon and Algorithmic in Licence at Grenoble INP.

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 is responsible of the third year of the computer science department of Grenoble INP. 222 hours per year. Master at Engineering school Polytech-Grenoble, Univ. Grenoble-Alpes.

Master: Frédéric Wagner. 220 hours per year. Engineering school ENSIMAG, Grenoble-INP (M1/2nd year and M2/3rd year).

Master: Yves Denneulin. 192 hours per year. Engineering school ENSIMAG, Grenoble-INP (M1/2nd year and M2/3rd year).

9.2.2. Supervision

PhD: Estelle Dirand, Integration of High-Performance Data Analytics and IOs for Molecular Dynamics on Exascale Computer, Univ. Grenoble-Alpes. November 2018. Bruno Raffin and Laurent Colombet (CEA).

PhD: Valentin Reis, Learning to Control Large-Scale Parallel Platforms, Univ. Grenoble-Alpes. October 2018. Advisers: Denis Trystram and Eric Gaussier

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: 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, Scheduling with Resource Augmentation, Advisers: Denis Trystram

PhD in progress: Clément Mommessin, Scheduling on heterogeneous platforms, Advisers: Denis Trystram

PhD in progress: Loris Felardos, Deep Learning for the Analytics of Molecular Systems, Advisers: Bruno Raffin, Guillaume Charpiat (Inria team Tau), Jérôme Hénin (IBPC).

PhD in Progress: Salah Zrigui, Learning Scheduling Strategies, Advisers: Denis Trystram and Fanny Dufossé.

9.2.3. Juries

PhD Defense of Jonathan Sarton, Visualisations interactives haute-performance de données volumiques massives : une approche out-of-core multi-résolution basée GPU, 28 Novembre 2018, Université de Reims Champagne-Ardenne. Reviewer

PhD Defense of Adrian Perez Dieguez, Parallel Prefix Operations on Heterogeneous Platforms, December 2018, Universidad da Coruna. Reviewer.

PhD Defense of Khalil Labidi, Parallelisation of hybrid metaheuristics for COP solving, 20 septembre 2018, Université de Tunis et Université Paris-Dauphine. President

PhD Defense of Stéphane Durand, Contrôle distribué et théorie des jeux : application aux systèmes auto-optimisants, 13 décembre 2018, Univ Grenoble Alpes. President

HDR Defense of Samuel Thibault, 13 décembre 2018, University Bordeaux I. Reviewer

9.3. Popularization

9.3.1. Interventions

- Talk *Des besoins en calcul de plus en plus performant* for the conference cycle "Accompagnement de l'informatique au lycée en 2018".

10. Bibliography

Major publications by the team in recent years

- [1] 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>

Publications of the year

Doctoral Dissertations and Habilitation Theses

- [2] E. DIRAND. *Integration of High-Performance Task-Based In Situ for Molecular Dynamics on Exascale Computers*, Université Grenoble - Alpes, November 2018, <https://hal.archives-ouvertes.fr/tel-01949170>
- [3] V. REIS. *Learning to control large-scale parallel platforms*, Université Grenoble Alpes (France), September 2018, <https://hal.inria.fr/tel-01965150>

Articles in International Peer-Reviewed Journals

- [4] M. AMARIS, G. LUCARELLI, C. MOMMESSIN, D. TRYSTRAM. *Generic Algorithms for Scheduling Applications on Heterogeneous Platforms*, in "Concurrency and Computation: Practice and Experience", July 2018, pp. 1-29 [DOI : 10.1002/CPE.4647], <https://hal.inria.fr/hal-01896868>
- [5] E. BAMPIS, A. KONONOV, D. LETSIOS, G. LUCARELLI, M. SVIRIDENKO. *Energy Efficient Scheduling and Routing via Randomized Rounding*, in "Journal of Scheduling", February 2018, vol. 21, n^o 1, pp. 35-51 [DOI : 10.1007/s10951-016-0500-2], <https://hal.inria.fr/hal-01725140>
- [6] J. BLAZEWICZ, B. MOSELEY, E. PESCH, D. TRYSTRAM, G. ZHANG. *New challenges in scheduling theory*, in "Journal of Scheduling", December 2018, vol. 21, n^o 6, pp. 581-582 [DOI : 10.1007/s10951-018-0571-3], <https://hal.archives-ouvertes.fr/hal-01963221>
- [7] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Further notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices*, in "Linear Algebra and Applications", 2018, vol. 554, pp. 68-78 [DOI : 10.1016/J.LAA.2018.05.017], <https://hal.inria.fr/hal-01586245>
- [8] É. GAUSSIER, J. LELONG, V. REIS, D. TRYSTRAM. *Online Tuning of EASY-Backfilling using Queue Reordering Policies*, in "IEEE Transactions on Parallel and Distributed Systems", October 2018, vol. 29, n^o 10, pp. 2304-2316 [DOI : 10.1109/TPDS.2018.2820699], <https://hal.archives-ouvertes.fr/hal-01963216>

- [9] L. JOURDAN, S. JACQUIN, F. DUFOSSÉ. *An exact algorithm for the bi-objective timing problem*, in "Optimization Letters", 2018, vol. 12, n^o 4, pp. 903–914 [DOI : 10.1007/s11590-018-1237-Y], <https://hal.archives-ouvertes.fr/hal-01716581>
- [10] 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", 2018, vol. 29, n^o 1, pp. 63-90 [DOI : 10.1142/S012905411850003X], <https://hal.sorbonne-universite.fr/hal-01516700>
- [11] G. LUCARELLI, B. MOSELEY, N. K. THANG, A. SRIVASTAV, D. TRYSTRAM. *Online Non-Preemptive Scheduling to Minimize Weighted Flow-time on Unrelated Machines*, in "Leibniz International Proceedings in Informatics ", 2018, vol. 112 [DOI : 10.4230/LIPICS.ESA.2018.59], <https://hal.archives-ouvertes.fr/hal-01986352>
- [12] J. TOSS, C. A. L. PAHINS, B. RAFFIN, J. L. D. COMBA. *Packed-Memory Quadtree: a cache-oblivious data structure for visual exploration of streaming spatiotemporal big data*, in "Computers and Graphics", 2018, vol. 76, pp. 117-128 [DOI : 10.1016/J.CAG.2018.09.005], <https://hal.archives-ouvertes.fr/hal-01876579>
- [13] J. VEIGA, R. R. EXPÓSITO, B. RAFFIN, J. TOURINO. *Optimization of Real-World MapReduce Applications With Flame-MR: Practical Use Cases*, in "IEEE Access", 2018, vol. 6, pp. 69750-69762 [DOI : 10.1109/ACCESS.2018.2880842], <https://hal.inria.fr/hal-01955503>
- [14] F. ZANON BOITO, E. CAMILO INACIO, J. LUCA BEZ, P. O. A. NAVAU, M. A. R. DANTAS, Y. DENNEULIN. *A Checkpoint of Research on Parallel I/O for High Performance Computing*, in "ACM Computing Surveys", March 2018, vol. 51, n^o 2, pp. 23:1-23:35 [DOI : 10.1145/3152891], <http://hal.univ-grenoble-alpes.fr/hal-01591755>

International Conferences with Proceedings

- [15] R. BLEUSE, K. DOGEAS, G. LUCARELLI, G. MOUNIÉ, D. TRYSTRAM. *Interference-Aware Scheduling using Geometric Constraints*, in "Euro-Par 2018 - European Conference on Parallel Processing", Torino, Italy, Euro-Par 2018: Parallel Processing, Springer, August 2018, vol. LNCS 11014, pp. 205-217 [DOI : 10.1007/978-3-319-96983-1_15], <http://hal.univ-grenoble-alpes.fr/hal-01884542>
- [16] B. CAMUS, A. BLAVETTE, F. DUFOSSÉ, A.-C. ORGERIE. *Self-Consumption Optimization of Renewable Energy Production in Distributed Clouds*, in "Cluster 2018 - IEEE International Conference on Cluster Computing", Belfast, United Kingdom, IEEE, September 2018, pp. 1-11, <https://hal.archives-ouvertes.fr/hal-01856660>
- [17] B. CAMUS, F. DUFOSSÉ, A. BLAVETTE, M. QUINSON, A.-C. ORGERIE. *Network-aware energy-efficient virtual machine management in distributed Cloud infrastructures with on-site photovoltaic production*, in "SBAC-PAD 2018 - International Symposium on Computer Architecture and High Performance Computing", Lyon, France, September 2018, pp. 1-8, <https://hal.archives-ouvertes.fr/hal-01856657>
- [18] E. DIRAND, L. COLOMBET, B. RAFFIN. *TINS: A Task-Based Dynamic Helper Core Strategy for In Situ Analytics*, in "SCA18 - Supercomputing Frontiers Asia 2018", Singapore, Singapore, March 2018, pp. 159-178 [DOI : 10.1007/978-3-319-69953-0_10], <https://hal.inria.fr/hal-01730910>
- [19] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Approximation algorithms for maximum matchings in undirected graphs*, in "CSC 2018 - SIAM Workshop on Combinatorial Scientific Computing", Bergen,

Norway, Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing, SIAM, June 2018, pp. 56-65 [DOI : 10.1137/1.9781611975215.6], <https://hal.archives-ouvertes.fr/hal-01740403>

- [20] G. LUCARELLI, B. MOSELEY, N. K. THANG, A. SRIVASTAV, D. TRYSTRAM. *Online Non-preemptive Scheduling on Unrelated Machines with Rejections*, in "Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures", Vienna, Austria, ACM Press, July 2018, pp. 291-300 [DOI : 10.1145/3210377.3210402], <https://hal.archives-ouvertes.fr/hal-01986312>
- [21] H. C. ZANÚZ, B. RAFFIN, O. A. MURES, E. J. PADRÓN. *In-Transit Molecular Dynamics Analysis with Apache Flink*, in "ISAV 2018 - In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization", DALLAS, United States, November 2018, pp. 1-8, <https://hal.inria.fr/hal-01889939>

Conferences without Proceedings

- [22] B. CAMUS, A. BLAVETTE, F. DUFOSSÉ, A.-C. ORGERIE. *Harnessing the geographical flexibility of distributed computing clouds for cooperative self-consumption*, in "ISGT-Europe 2018 - 8th IEEE PES Innovative Smart Grid Technologies Conference Europe", Sarajevo, Bosnia and Herzegovina, October 2018, pp. 1-6, <https://hal.archives-ouvertes.fr/hal-01856665>
- [23] M. MERCIER, A. FAURE, O. RICHARD. *Considering the Development Workflow to Achieve Reproducibility with Variation*, in "SC 2018 - Workshop: ResCuE-HPC", Dallas, United States, November 2018, pp. 1-5, <https://hal.inria.fr/hal-01891084>

Scientific Books (or Scientific Book chapters)

- [24] B. CAMUS, F. DUFOSSÉ, A.-C. ORGERIE. *The SAGITTA approach for optimizing solar energy consumption in distributed clouds with stochastic modeling*, in "Smart Cities, Green Technologies, and Intelligent Transport Systems", December 2018, pp. 1-25, <https://hal.archives-ouvertes.fr/hal-01945821>

Research Reports

- [25] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Effective heuristics for matchings in hypergraphs*, Inria Grenoble Rhône-Alpes, November 2018, n^o RR-9224, pp. 1-18, <https://hal.archives-ouvertes.fr/hal-01924180>
- [26] F. DUFOSSÉ, K. KAYA, I. PANAGIOTAS, B. UÇAR. *Scaling matrices and counting the perfect matchings in graphs*, Inria Grenoble Rhône-Alpes, March 2018, n^o RR-9161, pp. 1-22, <https://hal.inria.fr/hal-01743802>
- [27] A. LEGRAND, D. TRYSTRAM, S. ZRIGUI. *Adapting Batch Scheduling to Workload Characteristics: What can we expect From Online Learning?*, Grenoble 1 UGA - Université Grenoble Alpe, October 2018, n^o RR-9212, pp. 1-23, <https://hal.inria.fr/hal-01896121>

Other Publications

- [28] A. FAURE, M. POQUET, O. RICHARD. *Évaluation d'algorithmes d'ordonnancement par simulation réaliste*, April 2018, working paper or preprint, <https://hal.inria.fr/hal-01779936>
- [29] P. NEYRON. *Grid'5000*, June 2018, working paper or preprint, <https://hal.inria.fr/hal-01805122>