

RESEARCH CENTRE

**Grenoble - Rhône-Alpes**

IN PARTNERSHIP WITH:

Université de Grenoble Alpes, Institut  
polytechnique de Grenoble, CNRS

2021

ACTIVITY REPORT

Project-Team

DATAMOVE

## **Data Aware Large Scale Computing**

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

### **DOMAIN**

**Networks, Systems and Services,  
Distributed Computing**

### **THEME**

**Distributed and High Performance  
Computing**

# Contents

<b>Project-Team DATAMOVE</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>3</b>
<b>3 Research program</b>	<b>4</b>
3.1 Motivation . . . . .	4
3.2 Strategy . . . . .	4
3.3 Research Directions . . . . .	5
<b>4 Application domains</b>	<b>6</b>
4.1 Data Aware Batch Scheduling . . . . .	6
4.1.1 Algorithms . . . . .	6
4.1.2 Locality Aware Allocations . . . . .	7
4.1.3 Data-Centric Processing . . . . .	7
4.1.4 Learning . . . . .	8
4.1.5 Multi-objective Optimization . . . . .	8
4.2 Empirical Studies of Large Scale Platforms . . . . .	9
4.2.1 Workload Traces with Resource Consumption . . . . .	9
4.2.2 Simulation . . . . .	10
4.2.3 Job and Platform Models . . . . .	10
4.2.4 Emulation and Reproducibility . . . . .	11
4.3 Integration of High Performance Computing and Data Analytics . . . . .	11
4.3.1 Programming Model and Software Architecture . . . . .	11
4.3.2 Resource Sharing . . . . .	12
4.3.3 Co-Design with Data Scientists . . . . .	13
<b>5 Social and environmental responsibility</b>	<b>13</b>
<b>6 Highlights of the year</b>	<b>13</b>
<b>7 New software and platforms</b>	<b>14</b>
7.1 New software . . . . .	14
7.1.1 FlowVR . . . . .	14
7.1.2 OAR . . . . .	14
7.1.3 MELISSA . . . . .	15
7.2 New platforms . . . . .	15
7.2.1 SILECS/Grid'5000 and Meso Center Ciment . . . . .	15
<b>8 New results</b>	<b>15</b>
8.1 Data Aware Batch Scheduling . . . . .	15
8.1.1 Scheduling and Machine Learning. . . . .	16
8.1.2 On-line Scheduling Using Resource Augmentation . . . . .	16
8.1.3 Energy Saving through Smart Scheduling . . . . .	17
8.1.4 Scheduling for Edge Infrastructures . . . . .	17
8.1.5 Job Runtime Classification . . . . .	17
8.2 Empirical Studies of Large Scale Platforms . . . . .	17
8.2.1 An infrastructure simulation framework . . . . .	17
8.2.2 I/O aware scheduling . . . . .	17
8.2.3 Energy profiling and classification . . . . .	18
8.2.4 Reproducibility . . . . .	18
8.3 Integration of High Performance Computing and Data Analytics . . . . .	18
8.3.1 Data analysis for a single simulation run. . . . .	18
8.3.2 Data analysis for ensemble simulation runs. . . . .	18

<b>9</b>	<b>Bilateral contracts and grants with industry</b>	<b>19</b>
9.1	Bilateral grant with industry . . . . .	19
<b>10</b>	<b>Partnerships and cooperations</b>	<b>19</b>
10.1	STIC/MATH/CLIMAT AmSud project . . . . .	19
10.2	European initiatives . . . . .	19
10.3	National initiatives . . . . .	20
10.3.1	ANR . . . . .	20
10.3.2	INRIA . . . . .	20
10.3.3	Univ. Grenoble Alpes . . . . .	21
<b>11</b>	<b>Dissemination</b>	<b>21</b>
11.1	Promoting scientific activities . . . . .	21
11.1.1	Scientific events: organisation . . . . .	21
11.1.2	Scientific events: selection . . . . .	21
11.1.3	Journal . . . . .	21
11.1.4	Invited talks . . . . .	22
11.1.5	Scientific expertise . . . . .	22
11.2	Teaching - Supervision - Juries . . . . .	22
11.2.1	Teaching . . . . .	22
11.2.2	Juries . . . . .	22
<b>12</b>	<b>Scientific production</b>	<b>23</b>
12.1	Major publications . . . . .	23
12.2	Publications of the year . . . . .	23

## **Project-Team DATAMOVE**

*Creation of the Project-Team: 2017 November 01*

### **Keywords**

#### **Computer sciences and digital sciences**

- A1.1.4. – High performance computing
- A1.1.5. – Exascale
- A1.3.6. – Fog, Edge
- A1.6. – Green Computing
- A2.6.2. – Middleware
- A7.1.1. – Distributed algorithms
- A7.1.2. – Parallel algorithms
- A8.2.1. – Operations research
- A9.9. – Distributed AI, Multi-agent

#### **Other research topics and application domains**

- B3.3.2. – Water: sea & ocean, lake & river
- B6.4. – Internet of things

## 1 Team members, visitors, external collaborators

### Research Scientists

- Bruno Raffin [Team leader, Inria, Senior Researcher, HDR]
- Fanny Dufosse [Inria, Researcher]

### Faculty Members

- Christophe Cerin [Université de Paris, Professor, from Sep 2021, Délégation]
- Yves Denneulin [Institut polytechnique de Grenoble, Professor, HDR]
- Pierre-François Dutot [Univ Grenoble Alpes, Associate Professor]
- Gregory Mounie [Institut polytechnique de Grenoble, Associate Professor]
- Olivier Richard [Univ Grenoble Alpes, Associate Professor]
- Denis Trystram [Institut polytechnique de Grenoble, Professor, HDR]
- Frederic Wagner [Institut polytechnique de Grenoble, Associate Professor]

### Post-Doctoral Fellows

- Danilo Carastan Dos Santos [Univ Grenoble Alpes, from Oct 2021]
- Gwendal Jouan [Inria, until Feb 2021]

### PhD Students

- Luc Angelelli [Université Grenoble Alpes, from Oct 2022 until Sept 2024]
- Louis Boulanger [Univ Grenoble Alpes, from Oct 2021]
- Louis Boulanger [Institut polytechnique de Grenoble, from Oct 2021 until Sept 2024]
- Anderson Andrei Da Silva [Ryax technologies, CIFRE]
- Vincent Fagnon [Univ Grenoble Alpes]
- Ernest Foussard [Univ Grenoble Alpes]
- Sebastian Friedemann [Inria]
- Amal Gueroudji [CEA, CIFRE]
- Mathilde Jay [Univ Grenoble Alpes, from Oct 2021]
- Lucas Meyer [EDF, CIFRE]
- Angan Mitra [Qarnot Computing, CIFRE]
- Tuan Anh Nguyen [Univ Grenoble Alpes, from May 2021]
- Miguel Silva Vasconcelos [Univ Grenoble Alpes]
- Paul Youssef [Univ Grenoble Alpes]
- Salah Zrigui [Institut polytechnique de Grenoble, until Jun 2021]

## Technical Staff

- Jonathan Bleuzen [Univ Grenoble Alpes, Engineer, from Oct 2021]
- Christoph Conrads [Inria, Engineer]
- Adrien Faure [Institut polytechnique de Grenoble, Engineer, from Apr 2021]
- Baptiste Jonglez [Inria, Engineer]
- Tuan Anh Nguyen [INPG Entreprise SA, Engineer, until Apr 2021]
- Samir Noir [Inria, Engineer]
- Yacine Ould Rouis [CNRS, Engineer]
- Millian Poquet [Inria, Engineer]

## Interns and Apprentices

- Julien Buisine [Inria, from May 2021 until Jul 2021]
- Antoine Gitton [Institut polytechnique de Grenoble, from Apr 2021 until Jul 2021]
- Tirth Jain [Inria, from Jun 2021 until Aug 2021]

## Administrative Assistant

- Annie Simon [Inria]

## 2 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 are composed of few millions of cores, with performances almost reaching 100 PetaFlops<sup>1</sup> 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<sup>2</sup>. 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**: *Increasing*

<sup>1</sup>10<sup>15</sup> floating point operations per second

<sup>2</sup>SciDAC Review 2010

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

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

To tackle these issues, we propose to intertwine theoretical research and practical developments in an agile mode. Algorithms with performance guarantees will be designed and experimented on large scale platforms with realistic usage scenarios developed with partner scientists or based on logs of the biggest available computing platforms (national supercomputers like Curie, or the BlueWaters machine accessible through our collaboration with Argonne National Lab). Conversely, a strong experimental expertise will enable to feed theoretical models with sound hypotheses, to twist proven algorithms with practical heuristics that could be further retro-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 open 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  $L_\infty$ -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 **Standard Workload Format** workload trace format. This requires to instrument the platforms, in particular to trace resource consumptions like CPU, data movements at memory, network and I/O levels, with an acceptable performance impact. In a previous work we studied and proposed a dedicated job monitoring tool whose impact on the system has been measured as lightweight (0.35% speed-down) with a 1 minute sampling rate. Other tools also explore job monitoring, like TACC Stats. A unique feature from our tool is its ability to monitor distinctly jobs sharing common nodes.

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

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

#### 4.2.2 Simulation

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

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

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

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

#### 4.2.3 Job and Platform Models

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

enhanced or reworked to reflect the HPC platform evolutions. These models are then generally calibrated by running suitable benchmarks.

#### 4.2.4 Emulation and Reproducibility

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

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

In this context, the Grid'5000 (see Sec. 7.2) 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 beter performance than in-simulation analytics.

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

FlowVR enables to mix these different resources allocation strategies for the different stages of an

analytics pipeline. 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.

## 5 Social and environmental responsibility

DataMove is environmentally involved at different levels:

- Pursuing research on energy optimization of large scale distributed compute infrastructures
- Intend to include in publications the total amount of compute hours required for running all associated experiments, especially when using supercomputers, to, in a first step get a measure of the impact of our experimentation activity.
- Lead and participate to different local LIG and INRIA groups in charge of evaluating, proposing and implementing solutions to limit our environmental impact in the lab
- Take actions for lowering our carbon impact (extend laptop, smart phones, servers life to 5-8 years, favor fixing equipment rather than replacing them, put priority on train rather than plane)
- Bicycle is just our favorite, very low carbon, way for commuting.

## 6 Highlights of the year

This was a Covid year, very perturbed....



## 7 New software and platforms

### 7.1 New software

#### 7.1.1 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.

**URL:** <http://flowvr.sf.net>

**Contact:** Bruno Raffin

**Participants:** Bruno Raffin, Clément Ménéier, Emmanuel Melin, Jean Denis Lesage, Jérémie Allard, Jérémy Jaussaud, Matthieu Dreher, Sébastien Limet, Sophie Robert, Valérie Gourantou

#### 7.1.2 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).

**URL:** <http://oar.imag.fr>

**Contact:** Olivier Richard

**Participants:** Bruno Bzeznik, Olivier Richard, Pierre Neyron

**Partners:** LIG, CNRS, Grid'5000, CIMENT, UAR GRICAD

### 7.1.3 MELISSA

**Keywords:** Sensitivity Analysis, HPC, Data assimilation, Exascale

**Functional Description:** Melissa is a middleware framework for on-line processing of data produced from large scale ensemble runs (parameter sweep data analysis). Initial developments focused on sensibility analysis, Melissa relying on iterative statistics to provide a file avoiding, fault tolerant and elastic framework. Largest runs so far involved up to 30k core, executed 80 000 parallel simulations, and generated 288 TB of intermediate data that did not need to be stored on the file system. Melissa was next extended to large scale data assimilation, with the integration of a simulation (or member) virtualization mechanism that enables to abstract the number of members from the actual resource allocations, further improving execution efficiency and elasticity.

**News of the Year:** Melissa has been extended for supporting large scale data assimilation. See <https://hal.archives-ouvertes.fr/LIG/hal-03017033v2>

**URL:** <https://gitlab.inria.fr/melissa>

**Publications:** [hal-01383860](#), [hal-01607479](#), [hal-03017033](#)

**Authors:** Theophile Terraz, Bruno Raffin, Alejandro Ribes, Bertrand Iooss

**Contact:** Bruno Raffin

**Partner:** Edf

## 7.2 New platforms

**Participants:** Olivier Richard.

### 7.2.1 SILECS/Grid'5000 and Meso Center Ciment

We are 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](#), and nationally by contributing to the [SILECS/Grid'5000 platform](#), our local resources being included in this platform. Olivier Richard is member of SILECS/Grid'5000 scientific committee. The OAR scheduler in particular is deployed on both infrastructures. DataMove is hosting several engineers dedicated to Grid'5000 support.

## 8 New results

### 8.1 Data Aware Batch Scheduling

**Participants:** Denis Trystram, Fanny Dufossé, Gregory Mounié, Pierre-François Dutoit.

### 8.1.1 Scheduling and Machine Learning.

Machine Learning is a hot topic that received recently a great attention for dealing with huge amount of data and uncertainties. Members of DataMove promoted a methodology based on simulation and machine learning to obtain efficient dynamic scheduling policies. The main idea is to focus on the learning scheme targeting the policies themselves, and not the specific parameters of the problem. With regards to the other axes of DataMove, we are also interested in using algorithms for analyzing the huge amount of produced data and to develop the links between Machine Learning methods, distributed computing and (local) resource management. This has been done in a series of studies involving three consecutive PhDs (Valentin Reis, Danilo Carastan Santos and Salah Zrigui). We started by classical investigations to figure out if putting some effort for predicting the parameters of the jobs is fruitful or not to improve the scheduling decisions. Thus, we identified that it was better to put emphasis on learning the scheduling policies directly. The last piece of this puzzle was in showing that a superficial knowledge of some well-chosen characteristics are enough for improving the results.

Using intensive simulations and powerful synthetic workload generation model, it is possible to determine the characteristics of jobs that lead to a reduction in the mean slowdown in an execution queue. Modeling these characteristics using a non-linear function and applying this function to select the next job to execute in a queue dramatically improves the mean slowdown in synthetic workloads. When applied to actual workload traces from highly different platforms, these functions lead to important performance improvements, attesting the generalization property of the obtained heuristics.

A closely related problem is to design new policies. Most job scheduling algorithms apply greedy job priority ordering, such as First Come First Served (FCFS), Smallest Area First (SAF) or Shortest Processing time First (SPF). These methods are simple, highly practical but achieve only limited performance guarantees. We proposed to consider a new type of scheduling by combining policies. DataMove has developed a learning method permitting to adapt the mixed method to benchmarks.

### 8.1.2 On-line Scheduling Using Resource Augmentation

Resource augmentation is a well-established (and powerful) model for analyzing algorithms, particularly in online setting. It has been successfully used for providing theoretical evidence for several heuristics in scheduling with good performance in practice. According to this model, the algorithm is applied to a more powerful environment than that of the adversary. Several types of resource augmentation for scheduling problems have been proposed up to now, including speed augmentation, machine augmentation and more recently rejection. We presented a framework that unifies the various types of resource augmentation. It allows to generalize the notion of resource augmentation for other types of resources. Our framework is based on Dual Fitting. It consists of extending the domain of feasible solutions for the algorithm with respect to the domain of an adversary. This, in turn, allows the natural concept of duality for mathematical programming to be used as a tool for the analysis of the algorithm performance. As an illustration of the above ideas, we apply this framework, and we propose a primal-dual algorithm for the online scheduling problem of minimizing the total weighted flow time of jobs on unrelated machines when the preemption of jobs is not allowed.

We derived a collection of results on on-line non-preemptive scheduling algorithms: Targeting the minimization of the weighted flow time on unrelated machines, we 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  $\varepsilon_s > 0$ . We proved that this is already enough to achieve a competitive ratio of  $O(1/(\varepsilon_s \varepsilon_r))$ . More recently, we continued the study showing that it is sufficient to reject a  $2\varepsilon$  fraction of the total number of jobs to achieve a competitive ratio of  $2(\frac{1+\varepsilon}{\varepsilon})$ . We also considered 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  $\varepsilon$  of the total weight of all the jobs. We also derived a positive result for jobs with hard deadlines, where the objective is to minimize the total energy usage and no job may be rejected. Finally, we closed the story in generalizing 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.

### 8.1.3 Energy Saving through Smart Scheduling

DataMove is the principal investigator of an ANR project **Energumen** that studies the way of reducing the energy consumption of HPC cluster by the re-distribution of resources during the execution of jobs. The leverage mechanisms are speed-scaling and switch-off leading to malleability and elasticity.

### 8.1.4 Scheduling for Edge Infrastructures

Edge computing is a new paradigm that promotes to compute as close as possible to the place where the data are produced. Then, the heavy communications to the large data centers in the cloud are avoided. This induces a more complex task orchestration and also, to revisit the computations by low-cost methods, well-suited to more simpler computing devices like smart phone and small local data-centers. A first result has focused on several extensions implemented on Batsim/SimGrid toolkit. These extensions permit to develop and compare scheduling policies and data placement strategies. The efficiency of these extension is validated on the use-case of Qarnot Computing platform by comparing several scheduling strategies.

### 8.1.5 Job Runtime Classification

Job scheduling in high-performance computing platforms is a hard problem that involves uncertainties on both the job arrival process and their execution times. Most of the classical scheduling heuristics considered Jobs arrival time and duration as parameters of the scheduling problem. We proposed a way to classify the jobs depending of their runtime estimates. The classifier is based on classical low-cost machine learning methods. The idea is to use the data of the past weeks as learning dataset, to classify the jobs of the current week. The jobs are then classified between small and large. An evaluation over several benchmarks with different classical scheduling methods shows results as efficient as clairvoyant schedules.

## 8.2 Empirical Studies of Large Scale Platforms

**Participants:** Olivier Richard, PierreFrançois Dutot, Yves Denneulin.

### 8.2.1 An infrastructure simulation framework

The main scientific achievement is the study and development of the simulation framework Batsim. As large scale computation systems are growing to exascale, RJMSs need to evolve to manage this scale modification. Batsim is an extendable, language-independent and scalable RJMS simulator. It allows researchers and engineers to test and compare any scheduling algorithm, using a simple event-based communication interface, which allows different levels of realism. We have demonstrated that Batsim's behaviour matches the one of the real RJMS **OAR**.

### 8.2.2 I/O aware scheduling

Scheduling large jobs with respect to their respective I/O needs to maximize throughput is a complex problem. It requires to build a job model that includes the I/O aspects and also a precise representation of the I/O capabilities of the platforms. The latter part is especially difficult because it involves various parameters both hardware (disk throughput, network connectivity,...) and software (I/O scheduling policy, device drivers capabilities,...). To complicate the matter, the software often evolve thus rendering the previous representation obsolete. In the FUI IDIOM project, we helped build a set of tools to automatically characterize a platform I/O behavior. We also defined a job scheduling strategy to optimize throughput based on this representation and the jobs I/O needs. In an another way, we are exploring a

control-based approach (feedback loop) to prevent I/O congestion on shared file system in HPC platform due to bag-of-task workloads [13]. This work is done in collaboration with the Ctrl-A INRIA team.

### 8.2.3 Energy profiling and classification

were performed on traces collected on **Gricad** HPC center of Grenoble Alps university [18]. The results have been exploited to propose scheduling policies based on machine learning.

### 8.2.4 Reproducibility

The ability to reproduce an experiment is fundamental in this research axis. 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. We called this ability reproducibility with variation. We have shown 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. We proposed to leverage functional package managers to achieve this goal.

## 8.3 Integration of High Performance Computing and Data Analytics

**Participants:** Bruno Raffin, Frederic Wagner, Yves Denneulin.

### 8.3.1 Data analysis for a single simulation run.

We are working with CEA (PhD of Amal Gueroudji) to enable in situ processing for the Dask distributed task programming environment. Initial results are very positive, DASK presenting a good trade-off between the performance of pure HPC solutions and the parallel programming ease of Big Data ones.

### 8.3.2 Data analysis for ensemble simulation runs.

We put significant efforts in investigating in situ processing beyond a single large-scale simulation, considering use cases where the analysis needs to combine data from multiple simulation runs (also commonly called ensemble run). Such use-cases are becoming more common with the need to sample the simulation behavior within some parameter ranges for extracting knowledge using statistical or machine learning based methods, combined with the availability of large supercomputers capable today of running thousands of large simulation instances. Each simulation being potentially large, the amount of data generated by multiple runs is huge, leading to a pressing need for frugal I/O solutions. Initial work focused on sensibility analysis, where the data produced by the simulation are aggregated to compute statistics. We developed Melissa, a file avoiding, fault tolerant and elastic framework. Melissa is built around two key concepts: iterative (sometimes also called incremental) statistics algorithms and asynchronous client/server model for data transfer. Simulation outputs are never stored on disc. They are sent by the simulations to a parallel server, which aggregates them to the statistic fields in an iterative fashion, and then throw them away. This allows to compute oblivious statistics maps on every mesh element for every time step on a full scale study. Largest runs so far involved up to 30k core, executed 80 000 parallel simulations, and generated 288 TB of intermediate data that did not need to be stored on the file system. Then we extended this work in two directions. Instead of computing statistics we are investigating how to train on-line surrogate models using deep learning. This work is still on-going, with a PhD student (Lucas Meyer) funded by EDF [17]. The other direction focused on extending Melissa for ensemble based Data Assimilation (DA) (Sebastian Friedemann PhD). Data assimilation is more demanding than open-loop data processing, as the analysis results are used to steer the simulations progresses. The general approach consists in periodically correcting the simulations trajectories by minimizing the global error obtained from the combination of observation data, typically acquired by

on-the-field sensors, and simulation data. Data assimilation is particularly used in domains like weather forecast and climate simulation where numerical models are highly sensible to parameter values. Our solution extends Melissa with simulation virtualization and dynamics load balancing, further improving the elasticity and efficiency of executions. Experiments run on up to 16240 cores, to propagate 16384 members using the ParFlow hydrology simulation code.

## 9 Bilateral contracts and grants with industry

**Participants:** Bruno Raffin, Denis Trystram, Olivier Richard.

### 9.1 Bilateral grant with industry

- **EDF R&D (2020-2023)**. PhD grant (Lucas Meyer). 160K euros.
- **Qarnot Computing (2019-2022)**. PhD grant (Angan Mitra). 175K euros
- **Ryax Technologies (2020-2023)**. PhD grant (Anderson Andrei Da Silva). 170K euros.

## 10 Partnerships and cooperations

### 10.1 STIC/MATH/CLIMAT AmSud project

STIC AmSud SAQED

**Participants:** Bruno Raffin.

- Title: Scalable Approximate Query Evaluation on Document Inverted Files for GPU based Big-Data Applications
- International Partner:
  - Universidad Nacional de San Luis - UNSL, Argentina
  - Universidad de Santiago de Chile - USACH, Chile
  - Universidade Federal de São Carlos - UFSCAR, Brazil
- Duration: 2019-2021
- Project providing travel money, mainly untouched due to Covid.

### 10.2 European initiatives

**Participants:** Bruno Raffin, Olivier Richard, Denis Trystram, Fanny Dufossé.

- **REGALE**

**Title:** An open architecture to equip next generation HPC applications with exascale capabilities

**Budget:** 7.5 M euros (Datamove: 680K euros)

**Duration:** 2021 - 2023

**Web Site:** [REGALE](#)

**Coordinator:** Institute of Communication and Computer Systems, Greece

**Partners:** ICCS, Andritz, Atos, BSC, Cineca, E4 Computer Engineering, EDF, LRZ, NTUA, Ryax Tech., Scio, TUM, TWT, Ubitech, University of Bologna, UGA.

**Summary:** DataMove is in charge of the workpackage *Sophisticated Resources Allocation and Management* with the goal of developing innovative resource allocation algorithms with a strong focus on energy savings. DataMove also brings The OAR batch scheduler that will be extended with novel resource allocation policies and a use-case based on large scale sensibility analysis performed with Melissa, that will be used to evaluate new resource allocation policies and monitoring tools.

- **EoCoE-II**

**Title:** Energy Oriented Center of Excellence : toward exascale for energy

**Duration:** 2019 - 2021

**Budget:** 8.6M euros (DataMove: 248K euros)

**Web Site:** [EoCoE-II](#)

**Coordinator:** CEA, France

**Partners:** CEA, FGZ, ENEA, BSC, CNRS, INRIA, CERFACS, MPG, Fraunhofer Institute, FAU, CNR, UNITN, PSNC, ULB, UBAH, CIEMAT, IFPEN, DDN, UNITOV.

**Summary:** DataMove is in charge of the workpackage on Ensemble Runs with the goal of developing innovative solutions for large scale data assimilation based on the Melissa architecture. Target applications include 2 of the 4 EoCoE-II challenges: hydrology and weather forecast.

- **PRACE-6IP**

**Title:** PRACE 6th Implementation Phase Project

**Duration:** 2019 - 2021

**Coordinator:** FORSCHUNGSZENTRUM JULICH GMBH

**Budget:** 25M euros (DataMove: 31K euros)

**Partners:** 49 partners

**Summary:** DataMove is working on providing a service for large scale sensitivity analysis based on the Melissa architecture.

## 10.3 National initiatives

**Participants:** Bruno Raffin, Olivier Richard, Denis Trystram, Fanny Dufossé, Gregory Mounié, Pierre-François Dutot.

### 10.3.1 ANR

- **ANR grant GRECO (2017-2021).** 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.

### 10.3.2 INRIA

- **INRIA Challenge 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 [Web Site](#)

### 10.3.3 Univ. Grenoble Alpes

- **Edge Intelligence chair of the Institute of Artificial Intelligence of Univ. Grenoble Alpes (MIA@Grenoble-Alpes)** (2019-2023). PI: Denis Trystram. The challenges are to design new machine learning methods that fully exploit the distributed character of the edge and to develop algorithms and subsequent pieces of software that will allow the deployment of the edge/fog hybrid infrastructures. The research agenda is two-fold. In the first hand, we study new methods for distributed machine learning and data analytic. In the second hand, we develop the models and mechanisms for the orchestration of efficient local resource management. Budget: 335K euro
- **IRS SoSCloud.** Dimensioning of green energy in Clouds, 2020-2023. UGA Grant. PhD funding. Co-advised by D. Cordeiro, USP, Brasil. Budget : 120 Keuros

## 11 Dissemination

**Participants:** Bruno Raffin, Olivier Richard, Denis Trystram, Fanny Dufossé, Olivier Richard, Yves Dennueulin, Frederic Wagner, Gregory Mounié.

### 11.1 Promoting scientific activities

#### 11.1.1 Scientific events: organisation

- Co-organizer of the Workshop on the **In Situ Co-Execution of High-Performance Computing and Data Analysis of ISC - July 2, 2021**

#### 11.1.2 Scientific events: selection

##### Member of the conference program committees

- EGPGV 2021.
- NPC 2021.
- CCgrid (IEEE/ACM International Symposium on Cluster, Cloud and Grid), 2018-2021.
- ISAV (Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, Supercomputing WOrkshop), 2016-2021.
- Lдав (IEEE Symposium on Large Data Analysis and Visualization), 2016-2021.
- ICPP (Internat. Conference on Parallel Processing) 2021.
- IEEE Cluster 2018, 2020-2021.
- Euro-par 2019-2021.
- IPDPS 2016-2021.

#### 11.1.3 Journal

##### Member of the editorial boards

- Member of the Editorial Board of Algorithms
- Member of the Editorial Board of Theory of Computing System TOCS
- 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).



#### 11.1.4 Invited talks

- JLESC Panel AI in HPC 15/12/2021
  - Moderator: Franck Cappello, ANL
  - Panelists:
    - \* Prasanna, Balaprakash, ANL
    - \* Piotr Luszczek, UTK
    - \* Takahide Matsuoka, Riken
    - \* Dario Garcia-Gasulla, BSC
    - \* Martin Schultz, JSC
    - \* Volodymyr Kindratenko,
    - \* Bruno Raffin, INRIA
- HPC3 (HPC CoE Council), Co-design workshop. 12 march 2021. Round-table Co-Design for New Usages
  - Panelists : Peter Coveney, Marta Garcia, Berk Hess, Giovanni Pizzi, Bruno Raffin

#### 11.1.5 Scientific expertise

- DOE proposal reviewer
- Reviewer for FWF Erwin Schrodinger Fellowship, Austria

### 11.2 Teaching - Supervision - Juries

#### 11.2.1 Teaching

- Denis Trystram. 200 hours per year, ENSIMAG, Grenoble-INP, Master
- Fanny Dufossé. 17 to 90 hours per year, Algorithmic, Licence. Univ. Grenoble-Alpes and Licence Ensimag, Combinatorial scientific computing, Master, ENS Lyon.
- 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.
- Grégory Mounié is responsible for the first year (M1) of the international Master of Science in Informatics at Grenoble (MOSIG-M1). 317 hours per year. Master (M1/2nd year and M2/3rd year) at Engineering school ENSIMAG, Grenoble-INP, Univ Grenoble Alpes.
- Bruno Raffin. 28 hours per year. Parallel System. International Master of Science in Informatics at Grenoble (MOSIG-M2).
- Olivier Richard is responsible for the third year of the computer science department of Grenoble INP. 222 hours per year. Master at Engineering school Polytech-Grenoble, Univ. Grenoble-Alpes.
- Frédéric Wagner. 220 hours per year. Engineering school ENSIMAG, Grenoble-INP, Master (M1/2nd year and M2/3rd year).
- Yves Denneulin. 192 hours per year. Engineering school ENSIMAG, Grenoble-INP, Master (M1/2nd year and M2/3rd year).

#### 11.2.2 Juries

- President of PhD jury of Alena Shilova, Strategies Saving Memory During Learning Deep Neural Networks, 7/12/2021, Bordeaux.

## 12 Scientific production

### 12.1 Major publications

- [1] R. Bleuse, S. Kedad-Sidhoum, F. Monna, G. Mounié and D. Trystram. ‘Scheduling independent tasks on multi-cores with GPU accelerators’. In: *Concurrency and Computation: Practice and Experience* 27.6 (Apr. 2015), pp. 1625–1638. DOI: [10.1002/cpe.3359](https://doi.org/10.1002/cpe.3359). URL: <https://hal.archives-ouvertes.fr/hal-01081625>.
- [2] B. Camus, A. Blavette, F. Dufossé and 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, Sept. 2018, pp. 1–11. URL: <https://hal.archives-ouvertes.fr/hal-01856660>.
- [3] P.-F. Dutot, M. Mercier, M. Poquet and O. Richard. ‘Batsim: a Realistic Language-Independent Resources and Jobs Management Systems Simulator’. In: *20th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*. 20th Workshop on Job Scheduling Strategies for Parallel Processing. Chicago, United States, 27th May 2016. URL: <https://hal.archives-ouvertes.fr/hal-01333471>.
- [4] G. Lucarelli, B. Moseley, N. K. Thang, A. Srivastav and D. Trystram. ‘Online Non-preemptive Scheduling on Unrelated Machines with Rejections’. In: *SPAA 2018 - 30th ACM Symposium on Parallelism in Algorithms and Architectures*. Vienna, Austria: ACM Press, 2018, pp. 291–300. DOI: [10.1145/3210377.3210402](https://doi.org/10.1145/3210377.3210402). URL: <https://hal.archives-ouvertes.fr/hal-01986312>.
- [5] T. Terraz, A. Ribes, Y. Fournier, B. Iooss and B. Raffin. ‘Melissa: Large Scale In Transit Sensitivity Analysis Avoiding Intermediate Files’. In: *The International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing)*. Denver, United States, Nov. 2017, pp. 1–14. URL: <https://hal.inria.fr/hal-01607479>.
- [6] F. Zanon Boito, E. Camilo Inacio, J. Luca Bez, P. O. A. Navaux, M. A. R. Dantas and Y. Denneulin. ‘A Checkpoint of Research on Parallel I/O for High Performance Computing’. In: *ACM Computing Surveys* 51.2 (Mar. 2018), 23:1–23:35. DOI: [10.1145/3152891](https://doi.org/10.1145/3152891). URL: <https://hal.univ-grenoble-alpes.fr/hal-01591755>.

### 12.2 Publications of the year

#### International journals

- [7] J. Blazewicz, D. Trystram, G. Zhang, E. Pesch and B. Moseley. ‘New Perspectives in Scheduling Theory’. In: *Journal of Scheduling* (24th Aug. 2021). DOI: [10.1007/s10951-021-00693-8](https://doi.org/10.1007/s10951-021-00693-8). URL: <https://hal.archives-ouvertes.fr/hal-03356485>.
- [8] F. Dufossé, C. Dürr, N. Nadal, D. Trystram and Ó. Vásquez. ‘Scheduling with a processing time oracle’. In: *Applied Mathematical Modelling* 104 (Apr. 2022), pp. 701–720. DOI: [10.1016/j.apm.2021.12.020](https://doi.org/10.1016/j.apm.2021.12.020). URL: <https://hal.archives-ouvertes.fr/hal-03523262>.
- [9] F. Dufossé, K. Kaya, I. Panagiotas and B. Uçar. ‘Scaling matrices and counting the perfect matchings in graphs’. In: *Discrete Applied Mathematics* 308 (Feb. 2022), pp. 130–146. DOI: [10.1016/j.dam.2020.07.016](https://doi.org/10.1016/j.dam.2020.07.016). URL: <https://hal.inria.fr/hal-01743802>.
- [10] N. Gast, M. Khatiri, D. Trystram and F. Wagner. ‘Analysis of Work Stealing with latency’. In: *Journal of Parallel and Distributed Computing* 153 (July 2021), pp. 119–129. DOI: [10.1016/j.jpdc.2021.03.010](https://doi.org/10.1016/j.jpdc.2021.03.010). URL: <https://hal.inria.fr/hal-03356234>.
- [11] S. Zrigui, R. Y. De Camargo, A. Legrand and D. Trystram. ‘Improving the Performance of Batch Schedulers Using Online Job Runtime Classification’. In: *Journal of Parallel and Distributed Computing* 164 (2nd Feb. 2022), pp. 83–95. DOI: [10.1016/j.jpdc.2022.01.003](https://doi.org/10.1016/j.jpdc.2022.01.003). URL: <https://hal.archives-ouvertes.fr/hal-03023222>.

### International peer-reviewed conferences

- [12] E. Foussard, M.-L. Espinouse, G. Mounié and M. Nattaf. ‘A Lot-Sizing Model for Maintenance Planning in a Circular Economy Context’. In: APMS 2021 - Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems. Vol. 631. IFIP Advances in Information and Communication Technology. Nantes, France: Springer, 2021, pp. 673–682. DOI: [10.1007/978-3-030-85902-2\\_72](https://doi.org/10.1007/978-3-030-85902-2_72). URL: <https://hal.archives-ouvertes.fr/hal-03352725>.
- [13] Q. Guilloteau, O. Richard, B. Robu and E. Rutten. ‘Controlling the Injection of Best-Effort Tasks to Harvest Idle Computing Grid Resources’. In: ICSTCC 2021 - 25th International Conference on System Theory, Control and Computing. Iași, Romania, 20th Oct. 2021, pp. 1–6. DOI: [10.1109/ICSTCC52150.2021.9607292](https://doi.org/10.1109/ICSTCC52150.2021.9607292). URL: <https://hal.inria.fr/hal-03363709>.

### Conferences without proceedings

- [14] E. Foussard, M.-L. Espinouse, G. Mounié and M. Nattaf. ‘Économie circulaire et planification de maintenances pour optimiser la durée de vie d’un produit’. In: ROADEF 2021 - 22ème congrès annuel de la Société française de recherche opérationnelle et d’aide à la décision. Mulhouse (en ligne), France, 26th Apr. 2021, pp. 1–2. URL: <https://hal.archives-ouvertes.fr/hal-03217743>.
- [15] A. Gueroudji, J. Bigot and B. Raffin. ‘DEISA: dask-enabled in situ analytics’. In: HiPC 2021 - 28th International Conference on High Performance Computing, Data, and Analytics. virtual, India: IEEE, 17th Dec. 2021, pp. 1–10. URL: <https://hal.archives-ouvertes.fr/hal-03509198>.
- [16] Q. Guilloteau, O. Richard, E. Rutten and B. Robu. ‘Collecte de ressources libres dans une grille en préservant le système de fichiers : une approche autonome’. In: COMPAS 2021 - Conférence d’informatique en Parallélisme, Architecture et Système. Lyon, France, 6th July 2021, pp. 1–11. URL: <https://hal.inria.fr/hal-03282727>.
- [17] L. Meyer, L. Pottier, A. Ribes and B. Raffin. ‘Deep Surrogate for Direct Time Fluid Dynamics’. In: NeurIPS 2021 - Thirty-fifth Workshop on Machine Learning and the Physical Sciences. Vancouver, Canada, 13th Dec. 2021, pp. 1–7. URL: <https://hal.archives-ouvertes.fr/hal-03451432>.

### Doctoral dissertations and habilitation theses

- [18] S. Zrigui. ‘Understanding and improving HPC performance using Machine Learning and Statistical analysis’. Université Grenoble Alpes [2020-....], 25th Mar. 2021. URL: <https://tel.archives-ouvertes.fr/tel-03327540>.

### Reports & preprints

- [19] J. Bertrand, F. Dufossé and B. Uçar. *Algorithms and data structures for hyperedge queries*. RR-9390. Inria Grenoble Rhône-Alpes, 1st Feb. 2021, p. 25. URL: <https://hal.inria.fr/hal-03127673>.
- [20] D. Carastan-Santos, A. A. Da Silva, D. Trystram, A. Goldman, A. Mitra, Y. Ngoko and C. Mommessin. *Short-Term Ambient Temperature Forecasting for Smart Heaters*. 4th Oct. 2021. URL: <https://hal.inria.fr/hal-03364728>.
- [21] E. Foussard. *Maintenance Planning for Circular Economy: Laundromat Washing Machines Case*. G-SCOP - Laboratoire des sciences pour la conception, l’optimisation et la production, 24th Feb. 2021. URL: <https://hal.archives-ouvertes.fr/hal-03151214>.