RESEARCH CENTRE

Grenoble - Rhône-Alpes

IN PARTNERSHIP WITH:

CNRS, Institut polytechnique de Grenoble, Université de Grenoble Alpes

2020 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

Project-Team DATAMOVE

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

Keywords

Computer sciences and digital sciences

A1.1.4. – High performance computing

A1.1.5. – Exascale

A2.6.2. – Middleware

A7.1.2. – Parallel algorithms

A8.2.1. – Operations research

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]
- Gwendal Jouan [Inria, Starting Research Position, from Nov 2020]

Faculty Members

- 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

- Essam Morsi [Inria, until Sep 2020]
- Millian Poquet [Inria, until Jan 2020]
- Malin Rau [Institut polytechnique de Grenoble]

PhD Students

- Anderson Andrei Da Silva [Ryax, CIFRE]
- Vincent Fagnon [Univ Grenoble Alpes]
- Adrien Faure [Atos, CIFRE]
- Ernest Foussard [Univ Grenoble Alpes, from Oct 2020]
- Sebastian Friedemann [Inria]
- Amal Gueroudji [CEA, CIFRE, from Apr 2020]
- Mohammed Khatiri [Grenoble INP, ATER]
- Lucas Meyer [EDF, CIFRE, from Nov 2020]
- Angan Mitra [Qarnot Computing, CIFRE]
- Clement Mommessin [Institut polytechnique de Grenoble]
- Ioannis Panagiotas [Inria, until Sep 2020]
- Miguel Silva Vasconcelos [Univ Grenoble Alpes, from Dec 2020]
- Paul Youssef [Univ Grenoble Alpes]
- Salah Zrigui [Univ Grenoble Alpes]

Technical Staff

- Christoph Conrads [Inria, Engineer, from Aug 2020]
- Baptiste Jonglez [Inria, Engineer, from Sep 2020]
- Samir Noir [Inria, Engineer]
- Albin Petit [Inria, Engineer, until Oct 2020]
- Millian Poquet [Inria, Engineer, from Feb 2020]
- Theophile Terraz [Floralis, Engineer, until May 2020]

Interns and Apprentices

- Achal Agarwal [Inria, until Jan 2020]
- Mohammed Almarakby [Univ Grenoble Alpes, from Feb 2020 until Jun 2020]
- Juan Baldonado [INPG Entreprise SA, from Jun 2020 until Jul 2020]
- Juan Baldonado [Inria, until Jun 2020]
- Louis Boulanger [Inria, from Feb 2020 until Aug 2020]
- Clement Domps [Univ Grenoble Alpes, from Feb 2020 until Jun 2020]
- David Emukpere [Univ Grenoble Alpes, from Feb 2020 until Jun 2020]
- Priya Mishra [Inria, from May 2020 until Aug 2020]
- Maiko Muller [Inria, from Feb 2020 until Aug 2020]
- Ian Thomas [Institut polytechnique de Grenoble, from Feb 2020 until Jun 2020]

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-feeded into adequate theoretical models.

3 Research program

3.1 Motivation

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

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

- Resource allocation
- Integration of numerical simulation and data analysis

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

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

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

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

3.2 Strategy

DataMove targets HPC (High Performance Computing) at Exascale. But such machines and the associated applications are expected to be available only in 5 to 10 years. Meanwhile, we expect to see a growing number of petaflop machines to answer the needs for advanced numerical simulations. A sustainable exploitation of these petaflop machines is a real and hard challenge that we 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.

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

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

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

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

3.3 Research Directions

DataMove research activity is organised around three directions. When a parallel job executes on a machine, it triggers data movements through the input data it needs to read, the results it produces (simulation results as well as traces) that need to be stored in the file system, as well as internal communications and temporary storage (for fault tolerance related data for instance). Modeling in details the simulation and the target machines to analyze scheduling policies is not feasible at large scales. We propose to investigate alternative approaches, including learning approaches, to capture and model the influence of data movements on the performance metrics of each job execution to develop **Data Aware Batch Scheduling** models and algorithms (Sec. 4.1). Experimenting new scheduling policies on real platforms at scale is unfeasible. Theoretical performance guarantees are not sufficient to ensure a new algorithm

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

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. 4.7). 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 caracterize jobs often does not lead to reliable information.

The third research direction **Integration of High Performance Computing and Data Analytics** (Sec. 4.12) 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.

4.2 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**.

4.3 Locality Aware Allocations

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

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

While current results are encouraging, they are however limited in performance by the low amount of information available to the scheduler. We look forward to extend ongoing work by progressively increasing application and network knowledge (by technical mechanisms like profiling or monitoring or by more sophisticated methods like learning). It is also important to anticipate on application resource usage in terms of compute units, memory as well as network and I/Os to efficiently schedule a mix of applications with different profiles. For instance, a simple solution is to partition the jobs as "communication intensive" or "low communications". Such a tag could be achieved by the users them selves or obtained by learning techniques. We could then schedule low communications jobs using leftover spaces while taking care of high communication jobs. More sophisticated options are possible, for instance those that use more detailed communication patterns and networking models. Such options would leverage the work proposed in Section 4.7 for gathering application traces.

4.4 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 4.12). By reducing data movements and scheduling the extra processing on resources not fully exploited yet, in-situ processing is expected to have also a significant positive energetic impact. Analytics codes can be executed in the same nodes than the application, often on dedicated cores commonly called helper cores, or on dedicated nodes called stagging nodes. The results are either forwarded to the users for visualization or saved to disk through I/O nodes. In-situ analytics can also take benefit of node local disks or burst buffers to reduce data movements. Future job scheduling strategies should take into account in-situ processes in addition to the job allocation to optimize both energy consumption and execution time. On the one hand, this problem can be reduced to an allocation problem of extra asynchronous tasks to idle computing units. But on the other hand, embedding analytics in applications brings extra difficulties by making the application more heterogeneous and imposing more constraints (data affinity) on the required resources. Thus, the main point here is to develop efficient algorithms for dealing with heterogeneity without increasing the global computational cost.

4.5 Learning

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

4.6 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.

4.7 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.

4.8 Workload Traces with Resource Consumption

Workload traces are the base element to capture the behavior of complete systems composed of submitted jobs, running applications, and operating tools. These traces must be obtained on production platforms to provide relevant and representative data. To get a better understanding of the use of such systems, we need to look at both, how the jobs interact with the job management system, and how they use the allocated resources. We propose a general workload trace format that adds jobs resource consumption to the commonly used SWF [5] workload trace format. This requires to instrument the platforms, in particular

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

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.

4.9 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.

4.10 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.

4.11 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.

4.12 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.

4.13 Programming Model and Software Architecture

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

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

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

4.14 Resource Sharing

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

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

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

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

But today the choice of the resource allocation strategy is mostly ad-hoc and defined by the programmer. We will investigate solutions that enable a cooperative use of the resource between the analytics and the simulation with minimal hints from the programmer. In situ processings inherit from the parallelization scale and data distribution adopted by the simulation, and must execute with minimal perturbations on the simulation execution (whose actual resource usage is difficult to know a priori). We need to develop adapted scheduling strategies that operate at compile and run time. Because analysis are often data intensive, such solutions must take into consideration data movements, a point that classical scheduling strategies designed first for compute intensive applications often overlook. We expect to develop new scheduling strategies relying on the methodologies developed in Sec. 4.6. 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).

4.15 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 enrergy 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.
- 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
- Bicycle is Datamove favorite transportation mode for commuting

6 Highlights of the year

s, Understand Computing by Arnold Rosenberg and Denis Trystram [14]. In this book the authors aim to endow the reader with an operational, conceptual, and methodological understanding of the discrete mathematics that can be used to study, understand, and perform computing.

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, instanciate it according to the target machine. FlowVR will process it and prepare everything so that in one command line you can deploy and start your application.

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

URL: `http://flowvr.sf.net`

Authors: Jérémie Allard, Clément Ménier, Bruno Raffin, Jean Denis Lesage, Emmanuel Melin, Sophie Robert, Sébastien Limet, Valérie Gourantou

Contact: Bruno Raffin

Participants: Bruno Raffin, Clément Ménier, 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

Contacts: Olivier Richard, Pierre Neyron

Participants: Bruno Bzeznik, Olivier Richard, Pierre Neyron

Partners: LIG, CNRS, Grid'5000, CIMENT

7.1.3 MELISSA

Name: Modular External Library for In Situ Statistical Analysis

Keyword: Sensitivity Analysis

Functional Description: Melissa is an in situ solution for sensitivity analysis. It implements iterative algorithms to compute spatio-temporal statistic fields over results of large scale sensitivity studies. Melissa relies on a client/server architecture, composed of three main modules:

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

URL: https://melissa-sa.github.io

Publications: hal-01383860, hal-01607479

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

Contacts: Bruno Raffin, Theophile Terraz

Partner: Edf

7.2 New platforms

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 ingineers dedicated to Grid'5000 support.

8 New results

8.1 Integration of High Performance Computing and Data Analytics

8.2 Ensemble Runs

Datamove is investigating on-line data processing for ensemble runs. Work first focused on sensibility analysis and lead to the development of the Melissa framework, and then extended to data assimilation and surrogate model training. A broad perspective on on-line data processing for ensemble runs has been published in [11] with lesson learned from the Melissa experience. In parallel we pursued a work to enable very large scale ensemble-based data assimilation [22]. Prediction of chaotic systems relies on a floating fusion of sensor data (observations) with a numerical model to decide on a good system trajectory and to compensate nonlinear feedback effects. Ensemble-based data assimilation (DA) is a major method for this concern depending on propagating an ensemble of perturbed model realizations. We developed an elastic, online, fault-tolerant and modular framework called Melissa-DA for large-scale ensemble-based DA. Melissa-DA allows elastic addition or removal of compute resources for state propagation at runtime. Dynamic load balancing based on list scheduling ensures efficient execution. Online processing of the data produced by ensemble members enables to avoid the I/O bottleneck of file-based approaches. Our implementation embeds the PDAF parallel DA engine, enabling the use of various DA methods. Melissa-DA can support extra ensemble-based DA methods by implementing the transformation of member background states into analysis states. Experiments confirm the excellent scalability of Melissa-DA, running on up to 16,240 cores, to propagate 16,384 members for a regional hydrological critical zone assimilation relying on the ParFlow model on a domain with about 4M grid cells.

8.3 Data Aware Batch Scheduling

8.3.1 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 been established to deal with the challenge of computing in edge infrastructures [10]. This paper describes 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.

8.3.2 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 in [24] 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.

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

- **EDF R&D (2020)**: Study of physics informed neural networks for training computational fluids dynamics metamodels.

9.2 Bilateral grants with industry

- **EDF R&D (2020-2023)**. PhD Grant (Lucas Meyer)
- **Qarnot Computing (2019-2022)**. PhD grant (Angan Mitra).
- **Ryax (2020-2023)**. PhD Grant (Anderson Andrei Da Silva)

10 Partnerships and cooperations

10.1 European initiatives

10.1.1 H2020 Projects

PRACE-6IP

Title: PRACE 6th Implementation Phase Project

Duration: 2019 - 2021

Coordinator: FORSCHUNGSZENTRUM JULICH GMBH

Partners:

- AKADEMIA GORNICZO-HUTNICZA IM. STANISLAWA STASZICA W KRAKOWIE (Poland)
- ASSOCIACAO DO INSTITUTO SUPERIOR TECNICO PARA A INVESTIGACAO E DESENVOLVIMENTO (Portugal)
- ASSOCIATION NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS (Bulgaria)
- BARCELONA SUPERCOMPUTING CENTER - CENTRO NACIONAL DE SUPERCOMPUTACION (Spain)
- BAYERISCHE AKADEMIE DER WISSENSCHAFTEN (Germany)
- BILKENT UNIVERSITESI VAKIF (Turkey)
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (France)
- CENTRUM SPOLOCNYCH CINNOSTI SLOVENSKEJ AKADEMIE VIED (Slovakia)
- CINECA CONSORZIO INTERUNIVERSITARIO (Italy)
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (France)
- DANMARKS TEKNISKE UNIVERSITET (Denmark)
- EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH (Switzerland)
- FORSCHUNGSZENTRUM JULICH GMBH (Germany)
- FUNDACION PUBLICA GALLEGA CENTRO TECNOLOGICO DE SUPERCOMPUTACION DE GALICIA (Spain)
- GEANT VERENIGING (Netherlands)
- GRAND EQUIPEMENT NATIONAL DE CALCUL INTENSIF (France)
- Gauss Centre for Supercomputing (GCS) e.V. (Germany)
- ISTANBUL TEKNIK UNIVERSITESI (Turkey)
- KOBENHAVNS UNIVERSITET (Denmark)
- KORMANYZATI INFORMATIKAI FEJLESZTESI UGYNOKSEG (Hungary)
- KUNGLIGA TEKNISKA HOEGSKOLAN (Sweden)
- LINKOPINGS UNIVERSITET (Sweden)
- MACHBA - INTERUNIVERSITY COMPUTATION CENTER (Israel)

- MAX-PLANCK-GESELLSCHAFT ZUR FORDERUNG DER WISSENSCHAFTEN EV (Germany)
- NATIONAL INFRASTRUCTURES FOR RESEARCH AND TECHNOLOGY (Greece)
- NATIONAL UNIVERSITY OF IRELAND GALWAY (Ireland)
- NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET NTNU (Norway)
- PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE AISBL (Belgium)
- POLITECHNIKA GDANSKA (Poland)
- POLITECHNIKA WROCLAWSKA (Poland)
- SYDDANSK UNIVERSITET (Denmark)
- TECHNISCHE UNIVERSITAET WIEN (Austria)
- THE CYPRUS INSTITUTE (Cyprus)
- THE UNIVERSITY OF EDINBURGH
- UNINETT SIGMA2 AS (Norway)
- UNITED KINGDOM RESEARCH AND INNOVATION
- UNIVERSIDADE DE COIMBRA (Portugal)
- UNIVERSIDADE DE EVORA (Portugal)
- UNIVERSIDADE DO MINHO (Portugal)
- UNIVERSIDADE DO PORTO (Portugal)
- UNIVERSITAET INNSBRUCK (Austria)
- UNIVERSITAET STUTTGART (Germany)
- UNIVERSITE DU LUXEMBOURG (Luxembourg)
- UNIVERSITEIT ANTWERPEN (Belgium)
- UNIVERSITETET I OSLO (Norway)
- UNIVERZA V LJUBLJANI (Slovenia)
- UPPSALA UNIVERSITET (Sweden)
- VSB - Technical University of Ostrava (Czech Republic)

Inria/DataMove contact: Bruno Raffin

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

EoCoE-II

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

Duration: 2019 - 2021

Coordinator: CEA

Partners:

- AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE (Italy)
- BARCELONA SUPERCOMPUTING CENTER - CENTRO NACIONAL DE SUPERCOMPUTACION (Spain)
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE CNRS (France)
- CENTRO DE INVESTIGACIONES ENERGETICAS, MEDIOAMBIENTALES Y TECNOLOGICAS-CIEMAT (Spain)

- CERFACS CENTRE EUROPEEN DE RECHERCHE ET DE FORMATION AVANCEE EN CALCUL SCIENTIFIQUE SOCIETE CIVILE (France)
- COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (France)
- CONSIGLIO NAZIONALE DELLE RICERCHE (Italy)
- ECOLE NORMALE SUPERIEURE DE LYON (France)
- FORSCHUNGSZENTRUM JULICH GMBH (Germany)
- FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. (Germany)
- FRIEDRICH-ALEXANDER-UNIVERSITAET ERLANGEN-NUERNBERG (Germany)
- IFP Energies nouvelles (France)
- INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE (France)
- MAX-PLANCK-GESELLSCHAFT ZUR FORDERUNG DER WISSENSCHAFTEN EV (Germany)
- RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN (Germany)
- UNIVERSITA DEGLI STUDI DI ROMA TOR VERGATA (Italy)
- UNIVERSITA DEGLI STUDI DI TRENTO (Italy)
- UNIVERSITE DE STRASBOURG (France)
- UNIVERSITE GRENOBLE ALPES (France)
- UNIVERSITE LIBRE DE BRUXELLES (Belgium)
- UNIVERSITY OF BATH

Inria contact: Bruno Raffin

Summary: DataMove is in charge of the workpackage on Ensemble Runs with the goal of developping inovative 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.

10.2 National initiatives

10.2.1 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.

10.2.2 Competitivity Clusters

- **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).

10.2.3 INRIA

- 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/

10.2.4 Univ Grenoble Alpes

- Denis Trystram leading the Edge Intelligence chair of the new Institute of Artificial Intelligence of Univ. Grenoble Alpes (MIA@Grenoble-Alpes).

11 Dissemination

11.1 Promoting scientific activities

Event Organization

- EGPGV (Eurographics Symposium on Parallel Rendering and Visualization). Chair of the Steering Comittee
- HeteroPar 2020, Steering committee.
- ISAV (Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, Supercomputing Workshop). Program chair (2020), (https://vis.lbl.gov/events/ISAV2020/).
- HPML (High Performance Machine Learning Workshop), workshop of IEEE/ACM CCGRID 2020. Co-organizer. https://hpml2020.github.io/

Member of the editorial boards

- Co-editor special issue Journal of Parallel and Distributed Computing, Advances on High Performance Computing for Artificial Intelligence Applications 2020.

Member of conference program committees

- SuperComputing
- IEEE Cluster
- AlgoCloud
- Europar
- HiPC
- CCGrid
- ISPDC
- ISRL
- ISAV
- HPC ASIA
- LDAV

Reviewer DataMove members regularly participate to the reviewing of publication for national and international venues and journals.

11.1.1 Invited talks

- Deep learning and high performance computing synergies. Journée Informatique de la Région Centre Val de Loire, https://jirc2020.sciencesconf.org/, 2020. Invited Speaker.

11.1.2 Scientific expertise

- ERC proposal reviewer
- DOE proposal reviewer call 2020: "DATA, ARTIFICIAL INTELLIGENCE, AND MACHINE LEARNING AT DOE SCIENTIFIC USER FACILITIES"
- Member of the Phidias Scientific and User Committee (SUC). https://www.phidias-hpc.eu/

11.2 Teaching

- Master: Denis Trystram. 200 hours per year in average.
- Licence: Fanny Dufossé. 17 hours per year. Algorithmic at Univ. Grenoble-Alpes.
- Master: Pierre-François Dutot. 226 hours per year. Licence (first and second year) at IUT2/UPMF (Institut Universitaire Technologique de Univ. Grenoble-Alpes) and 9 hours Master M2R-ISC Informatique-Systèmes-Communication at Univ. Grenoble-Alpes.
- Master: Grégory Mounié is responsible of the first year (M1) of the international Master of Science in Informatics at Grenoble (MOSIG-M1). 237 hours per year. Master (M1/2nd year and M2/3rd year) at Engineering school ENSIMAG, Grenoble-INP, Univ Grenoble Alpes.
- 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).

11.3 PhD Advising

- PhD: Clément Mommessin, Scheduling on heterogeneous platforms. Defended 11-12-2020. Advisers: Denis Trystram.
- PhD: Mohammed Khatiri, Tasks scheduling on heterogeneous Multicore. Defended 26-09-2020. Advisers: Denis Trystram, El Mostafa Daoudi (University Mohammed First, Oujda, Morocco).
- PhD: Adrien Faure, Advanced Resource Management for Supercomputers. Defended 02-12-2020. Advisers: Denis Trystram, Olivier Richard.
- PhD: Ioannis Panagiotas,On matchings and related problems in graphs, hypergraphs, and doubly stochastic matrices. Defended 09-10-2020. Advisers: Bora Uçar (LIP) and Fanny Dufossé.
- PhD in progress: Loris Felardos, Deep Learning for the Analytics of Molecular Systems, Advisers: Bruno Raffin, Guillaume Charpiat (INRIA team Tau), Jérome Hénin (IBPC).
- PhD in progress: Salah Zrigui, Learning Scheduling Strategies, Advisers: Denis Trystram, Arnaud Legrand.
- PhD in progress: Sebastian Friedemann, Large Scale Data Assimilation, Adviser: Bruno Raffin.
- PhD in progress: Vincent Fagnon, Analyse de politique d'ordonnancement dynamique pour objets mobiles , Adviser: Denis Trystram.

- PhD in progress: Angan Mitra, Theoretical and implementation challenges in Lifelong Learning and Edge Computing, Adviser: Denis Trystram.
- PhD in progress: Paul Youssef, Coresets Analysis , Adviser: Denis Trystram.
- PhD in progress: Lucas Meyer, Deep surrogate training. Advisers: Bruno Raffin and Alejanddro Ribes (EDF).
- PhD in progress: Amal Gueroudji, In Situ Data Processing with Distributed Task Programming Environments. Advisers: Bruno Raffin and Julien Bigot (CEA).
- PhD in progress: Miguel Silva Vasconcelos. Solar Scheduling. Advisers: Fanny Dufossé and Daniel de Angelis Cordeiro (USP Brazil).
- PhD in progress: Quentin Guilloteau. Autonomic computing for resource management in HPC clsuters. Advisers: Eric Rutten (Ctrl-A) and Olivier Richard.
- PhD in progress: Anderson Andrei Da Silva. Efficient middleware for data analysis in the Edge/Cloud. Advisers: Denis Trystram, Grégory Mounié and Yiannis Georgiou (Ryax).

11.4 PhD and HDR Juries

- HDR Defense Nicolas Gast, *Refinements of Mean Field Approximatio.* UGA, January 2020. President.
- HDR Defense Thierry Gautier, *Contribution à la définition des supports logiciels pour une Programmation et une Exécution Portable et Efficace en HPC – genèse et conception de Kaapi et X-Kaapi.* UGA, February 2020. UGA, February 2020. President.
- HDR Defense Stéphane Desvimes, *Versatility and Efficiency in Self-Stabilizing Distributed Systems.* UGA, 17 décembre 2020. Président.
- PhD Defense of Changjiang Gou, *Task Mapping and Load-balancing for Performance, Memory, Reliability and Energy.* ENS Lyon, September 2020. Jury.
- PhD Defense Li Han, *Algorithms for detecting and correcting silent and non-functional errors in scientific workflows.* ENS Lyon, May 2020. Jury.
- PhD Defense Massinissa Ait Aba (4 juin 2020) - Examinateur
- PhD Defense Alexandre Teiller (1 décembre 2020) - Rapporteur

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. 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] D. Carastan-Santos, R. Y. De Camargo, D. Trystram and S. Zrigui. ‘One can only gain by replacing EASY Backfilling: A simple scheduling policies case study’. In: *CCGrid 2019 - International Symposium in Cluster, Cloud, and Grid Computing*. Larnaca, Cyprus: IEEE, May 2019, pp. 1–10. DOI: 10.1109/CCGRID.2019.00010. URL: https://hal.archives-ouvertes.fr/hal-02237895.

[4] G. Lucarelli, N. Kim Thang, A. Srivastav and D. Trystram. 'Online Non-preemptive Scheduling in a Resource Augmentation Model based on Duality'. In: *European Symposium on Algorithms (ESA 2016)*. Vol. 57. 63. Aarhus, Denmark, Aug. 2016, pp. 1–17. DOI: 10.4230/LIPIcs.ESA.2016.63. URL: http://hal.univ-grenoble-alpes.fr/hal-01334219.

[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. URL: https://hal.univ-grenoble-alpes.fr/hal-01591755.

12.2 Publications of the year

International journals

[7] O. Beaumont, L.-C. Canon, L. Eyraud-Dubois, G. Lucarelli, L. Marchal, C. Mommessin, B. Simon and D. Trystram. 'Scheduling on Two Types of Resources: a Survey'. In: *ACM Computing Surveys* 53.3 (1st May 2020). DOI: 10.1145/3387110. URL: https://hal.inria.fr/hal-02432381.

[8] J. Luca Bez, F. Zanon Boito, R. Nou, A. Miranda, T. Cortes and P. O. Navaux. 'Adaptive Request Scheduling for the I/O Forwarding Layer using Reinforcement Learning'. In: *Future Generation Computer Systems* 112 (2020), pp. 1156–1169. DOI: 10.1016/j.future.2020.05.005. URL: https://hal.inria.fr/hal-01994677.

[9] J. Morrissey, P. Totoo, K. Hanley, S.-A. Papanicolopulos, J. Ooi, I. C. Gonzalez, B. Raffin, S. Mostajabodaveh and T. Gierlinger. 'Post-processing and visualization of large-scale DEM simulation data with the open-source VELaSSCo platform'. In: *SIMULATION* 96.7 (July 2020), pp. 567–581. DOI: 10.1177/0037549720906465. URL: https://hal.inria.fr/hal-02966840.

International peer-reviewed conferences

[10] A. A. Da Silva, C. Mommessin, P. Neyron, D. Trystram, A. Bauskar, A. Lebre, A. V. Kempen, Y. Ngoko and Y. Ricordel. 'Evaluating Computation and Data Placements in Edge Infrastructures through a Common Simulator'. In: SBAC-PAD 2020 - IEEE 32nd International Symposium on Computer Architecture and High Performance Computing. Porto, Portugal, 8th Sept. 2020. URL: https://hal.inria.fr/hal-02915346.

[11] A. Ribés and B. Raffin. 'The Challenges of In Situ Analysis for Multiple Simulations'. In: ISAV 2020 - In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization. Atlanta, United States, 12th Nov. 2020, pp. 1–6. URL: https://hal.inria.fr/hal-02968789.

National peer-reviewed Conferences

[12] V. J. Vendramini, A. Goldman and G. Mounié. 'Improving mobile app development using transpilers with maintainable outputs'. In: SBES 2020 - 34th Brazilian Symposium on Software Engineering. Natal, Brazil, 2020, pp. 1–10. DOI: 10.1145/3422392.3422426. URL: https://hal.archives-ouvertes.fr/hal-03000163.

Conferences without proceedings

[13] A. Faure, G. Lucarelli, O. Richard and D. Trystram. 'Online Scheduling with Redirection for Parallel Jobs'. In: 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). New Orleans, France, 18th May 2020, pp. 1–4. DOI: 10.1109/IPDPSW50202.2020.00066. URL: https://hal.archives-ouvertes.fr/hal-02944032.

Scientific books

[14] A. Rosenberg and D. Trystram. *Understand Mathematics, Understand Computing*. 2020. DOI: 10.1007/978-3-030-58376-7. URL: https://hal.archives-ouvertes.fr/hal-03153797.

Doctoral dissertations and habilitation theses

[15] A. Faure. 'Advanced Simulation for Resource Management'. Université Grenoble Alpes [2020-....], 2nd Dec. 2020. URL: https://tel.archives-ouvertes.fr/tel-03155702.

[16] M. Khatiri. 'Task scheduling on heterogeneous multi-core'. Université Grenoble Alpes [2020-....]; Université Mohammed Premier Oujda (Maroc), 26th Sept. 2020. URL: https://tel.archives-ouvertes.fr/tel-03026378.

[17] C. Mommessin. 'Efficient Management of Resources in Heterogeneous Platforms'. Université Grenoble Alpes [2020-....], 11th Dec. 2020. URL: https://tel.archives-ouvertes.fr/tel-03179102.

[18] I. Panagiotas. 'On matchings and related problems in graphs, hypergraphs, and doubly stochastic matrices'. Université de Lyon, 9th Oct. 2020. URL: https://tel.archives-ouvertes.fr/tel-03011794.

Reports & preprints

[19] A. Bauskar, A. da Silva, A. Lebre, C. Mommessin, P. Neyron, Y. Ngoko, Y. Ricordel, D. Trystram and A. Van Kempen. *Investigating Placement Challenges in Edge Infrastructures through a Common Simulator (extended version)*. INRIA, 25th May 2020, p. 21. URL: https://hal.inria.fr/hal-02153203.

[20] J. Bertrand, F. Dufossé and B. Uçar. *Algorithms and data structures for hyperedge queries*. Inria Grenoble Rhône-Alpes, 1st Feb. 2021, p. 21. URL: https://hal.inria.fr/hal-03127673.

[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.

[22] S. Friedemann and B. Raffin. *An elastic framework for ensemble-based large-scale data assimilation*. Inria Grenoble Rhône-Alpes, Université de Grenoble, 20th Nov. 2020. URL: https://hal.inria.fr/hal-03017033.

[23] M. Khatiri, D. Trystram and F. Wagner. *Work Stealing Simulator*. 17th Jan. 2020. URL: https://hal.archives-ouvertes.fr/hal-02444049.

[24] S. Zrigui, R. Y. De Camargo, A. Legrand and D. Trystram. *Improving the Performance of Batch Schedulers Using Online Job Runtime Classification*. 2020. URL: https://hal.archives-ouvertes.fr/hal-03023222.

Other scientific publications

[25] Q. Guilloteau. 'Minimizing Cluster Under-use using a Control-Based Approach'. Grenoble INP Ensimag; Université Grenoble Alpes, 25th June 2020. URL: https://hal.archives-ouvertes.fr/hal-03167242.