Inria Project Lab (IPL): Long Version

HPC-BigData
High Performance Computing and Big Data

- **Project lead:**
  Bruno Raffin

  Project lead's email:
  bruno.raffin@inria.fr

  Project lead's team (EP):
  DataMove

  Project lead's research center (CRI):
  Grenoble Rhône-Alpes

- **Internal partners (people involved per EP) and external partners**
  Inria partners:

  - DataMove, Grenoble: Bruno Raffin, Olivier Richard
  - KerData, Rennes: Gabriel Antoniu, Alexandru Costan
  - SequeL, Lille: Philippe Preux
  - Zenith, Montpellier: Patrick Valduriez, Alexis Joly
  - Parietal, Saclay: Gael Varoquaux
  - TADaaM, Bordeaux: Emmanuel Jeannot, Guillaume Aupy
  - HiePACS, Bordeaux: Olivier Coulaud
  - Storm, Bordeaux: Samuel Thibault
  - RealOpt, Bordeaux: Olivier Beaumont
  - Sierra, Paris: Francis Bach
  - Tau, Saclay: Guillaume Charpiat

  External partners:

  - ATOS/Bull, Grenoble: Stéphane Pralet
  - Argonne National Lab (ANL), USA: Rob Ross
  - Laboratoire de Biochimie Théorique (LBT), CNRS, Paris: Marc Baaden, Jérôme Hénin henin@ibpc.fr
  - ESI-Group, Paris: Fouad El-Khaldi.
  - Grid'5000: Pierre Neyron

Résumé

HPC and Big Data evolved with their own infrastructures (supercomputers versus clouds), applications (scientific simulations versus data analytics) and software tools (MPI and OpenMP versus Map/Reduce or Deep Learning frameworks). But Big Data analytics is becoming more compute-intensive (thanks to deep learning), while data handling is becoming a major concern for scientific computing. The goal of this IPL is to gather teams from HPC, Big Data and Machine Learning (ML) areas to work at the intersection between these domains. Research is organized along three main axes: high performance analytics for scientific computing applications, high performance analytics for big data applications, infrastructure and resource management.

1 Motivation

This IPL proposal is a follow-up of the work on the scientific challenge *When extreme-scale computing meets data-intensive science* for the next Inria strategic roadmap (see Iris group[1]). The goal is to gather Inria teams from the HPC, Big Data processing and Machine Learning (ML) areas and relevant external collaborators (ATOS/Bull, Argonne National Lab, ESI-group, Laboratoire de Biochimie Théorique, Grid'5000) to address some specific aspects of this challenge. Many EP are involved, but actually only a few people per EP will effectively be active in this IPL. The title page of this IPL lists all the persons involved.

High Performance Computing (HPC) aims to run scientific simulations on supercomputers to address major scientific and engineering challenges. Data-intensive processing (Big Data) aims to analyze, extract knowledge and make decisions based on huge data corpora. Both fields evolved separately. HPC supercomputers are standalone machines with millions of computing cores aggregated through a dedicated network. They run highly-optimized parallel simulations, and performance is the major concern. In contrast, Big Data analytics (data aggregation and mining) runs in the cloud, an elastic environment built on cost-effective commodity components. Data analytics relies on generic paradigms (map/reduce, neural networks) to process large static or streamed datasets, while trading performance for productivity and programming simplicity.

Today, Big Data applications are expected to move towards more compute-intensive algorithms for descriptive (data aggregation and mining) and predictive (statistics and forecasting) analysis, while increasingly leveraging machine-learning approaches. HPC performance-focused approaches could contribute to push Big Data solutions to higher performance levels. Symmetrically, analytics is expected to become a fully-fledged software component of the HPC ecosystem to process the massive results of large scientific simulations or to feed numerical models with complex data produced by other scientific tools. This analytics phase of HPC simulations should benefit from the advances in Big Data processing. More generally, *increasing coherence between the technology base used for modeling and simulation and that used for data analytic computing* is becoming a critical concern, as stated by the second top objective of the US Strategic Computing Initiative[2].

This convergence between Big Data and HPC has recently started to receive substantial international attention (BDEC working group [3] on the convergence of HPC and Big Data, joint actions between ETP4HPC[4] and BDVA [5]). The challenge is to jointly leverage the know-how and existing solutions from both domains to enable further progress in advanced computational science.

[1] https://reseau-iris.inria.fr/pages/view/81915/redaction-defis-scientifique

[2] https://obamawhitehouse.archives.gov/the-press-office/2015/07/29/executive-order-creating-national-strategic-computing-initiative

[3] http://www.exascale.org/bdec/

[4] http://www.etp4hpc.eu/

[5] http://www.bdva.eu

2 Challenges

Scientific challenges are organized around three main axis detailed bellow.

2.1 High Performance Analytics for Scientific Computing Applications

Scientists are spending today a substantial amount of their time to extract relevant information from the huge amount of data produced by scientific simulations. Big Data and ML have developed innovative solutions for applications related to web and business data. The goal is here to investigate how these solutions could be adapted for data produced by scientific simulations, in particular in an in situ processing model. As preliminary studies showed that existing Big Data software stacks (Hadoop, Flink, Spark, etc.) do not naturally perform well on supercomputers, we will explore how to revisit them for scientific-data analytics on the considered converged architecture. ML and DL are also offering novels ways to extract hidden knowledge, to perform dimension reduction or to detect events of interest. Scientific computing can leverage such capabilities for result analysis, setting the hyper parameters of simulations (potentially interactively while the simulation is running), produce surrogate models capturing the essence of the simulation but at a significantly lower compute cost to be used for sensitivity analysis or real-time decision making. The potential of ML approaches in the domain of scientific computing still need to be explored. Challenges regard several levels: the algorithmic level (leverage new ML-based algorithms to extract knowledge from simulation data); the software infrastructure level (design converged frameworks where Big Data analytics and ML tools can be used for in situ/in transit processing); the resource management level (explore how to manage resources for a converged architecture).

2.2 High-Performance Analytics for Big Data Applications

Machine learning algorithms need high computational power to deal with increasingly larger datasets. Our goal is to investigate ways to give them access to such a computational power on HPC machines at an almost null cost in terms of technical difficulty, and programming. This agenda faces algorithmic and engineering roadblocks. From the algorithmic standpoint, the challenges are to develop parallel algorithms that minimize data transfer across the nodes as well as synchronization barriers. From the engineering standpoint, their is dire need for an interface to express as naturally on HPC as on single workstations these algorithms to substantially improve the support for parallel tools like TensorFlow (Google) or scikit-learn (Parietal).

There are a variety of different data-processing scenario that lead to different challenges in distributed computing. Big data analytics perform database-like operations on data structures spread across the network. It calls for out-of-core data primitives, as for instance in the Spark platform. Typically, the data do not fit in the memory of one node and the algorithms try to make operations as local as possible. Another resource-intensive computation pattern is DL. In such settings, even computing the model on a single data is very costly. Common solutions rely on fitting the model on GPUs located on the same computer, using dedicating algorithms – *on-line learning* – to consider only a small fraction of the data that are then streamed directly from storage. Sharing the load across computers is then challenging, as inter-GPU communication introduces a large cost. However, it is often necessary to fit many models for instance varying hyper-parameters or splitting the learning. These mini-batch approaches require frequent synchronizations to keep good learning convergence. Asynchronous approaches expose more parallelism but today solutions are impaired by slower learning rates.

Reinforcement learning and in particular *Deep Reinforcement Learning* further enforce the need for parallelism as the neural network learns through interactions with running simulations of the object to learn. For instance AlphaGo Zero [18] was a leap forward in AI as it outperformed the best human players (and itself) after being trained without using data from human games but solely through unsupervised learning. The approach used combines deep learning and reinforcement learning, often called Deep Reinforcement Learning. The process requires an advanced infrastructure for the training phase. To speed up the learning process and enable a wide exploration of the parameter space, the neural network interacts in parallel with several instances of the simulation. For instance AlphaGo Zero trained during more than 70 hours using 64 GPU workers and 19 CPU parameter servers for playing 4.9 million games of generated self-play, using 1,600 simulations for each Monte Carlo Tree Search.

2.3 Infrastructure and Resource Management

Traditionally, HPC and Cloud/Big Data infrastructures distinguish themselves on two major aspects. First, resource allocation strategies rely on exclusive node access per application (HPC) whereas they leverage time sharing (virtua lization or fair-sharing scheduling) on cloud-based Big Data platforms. Second, supercomputers rely on external storage (dedicated nodes) while Big Data infrastructures assume that data can be stored locally (and replicated) on disks attached to the compute nodes. Last, accelerators and in particular GPUs are becoming the key processing unit for enabling high performance Deep Learning (DL). A new breed of dedicated AI supercomputers is emerging mixing a high performance network, on-ode storage, massive use of GPUs. Resources tend to be managed following cloud-based approaches (container technology). In this context questions of interest include new locality-aware scheduling techniques to handle resource allocation, design of a convergent elastic storage systems to meet semantics and performance requirements for both scientific computations and Big Data analytics. Machine learning may be used to develop new strategies.

3 Work Program

We propose to address the challenges discussed before through 4 work packages, namely use cases, algorithms, frameworks, and infrastructure discussed in the following.

3.1 Use Cases

Goal: analyze the requirements of these applications, which should serve as a reference to validate the outcome of the other work packages. Expected outcome: requirements analysis in terms of open challenges at all levels: algorithms, software infrastructure, frameworks, resource management.

Participants: Zenith (Pl@ntNet), LBT (MD Simulations), ESI-Group (Computational Mechanics), ANL (Coherent Diffraction).

3.1.1 Species Distribution Modelling from Pl@ntNet Data (INRIA)

Context: Pl@ntNet is a participatory platform dedicated to the crowsdourcing of botanical data assisted by the automated identification of plants. It was initiated in 2010 by a French consortium including Inria, Inra, CIRAD and IRD. Nowadays, it is used by millions of people all over the world (*e.g.* 3 million users of the mobile application in 2017 with an average of about 13 sessions per user). An impact study concluded that about 12% of uses are done in the context of professional activities in agriculture, education, trade or tourism. Besides, the collected data is used by a growing number of researchers in ecology, agronomy or computer science.
Scenario: Today, only 3% of the data collected through the Pl@ntNet platform is shared with the scientific community because of the human validation bottleneck (even collaborative). A fully automated validation process would allow exploiting much more plant occurrences and, in fine, setting up a biodiversity indicator at an unprecedented scale. However, achieving this objective requires selecting, optimizing, running and upgrading very large convolutional neural networks to be trained iteratively on tens of millions of web images.
Data: At the time of writing, the Pl@ntNet data-store accounts for about 30 million of observations, each observation being composed of 1 to 5 plant images and additional meta-data (geo-location, date, author, device, etc.). This represents about 10 Tb of raw storage nowadays. Because of the strong growth of the number of sessions (x2 every year), it is estimated that the number of observations should reach 100 millions in 2021.
Requirements analysis: Training a convolutional neural network with millions of images can take up to one week on a single node equipped with four recent GPUs. To select the best performing architecture and optimize the hyper-parameters, it is necessary to train tens of such networks. Overall, training a classifier such as the one required for the recognition of the world's flora is a highly intensive computational task that has to be necessary distributed on a large HPC infrastructure. Moreover, the models will have to be regularly upgraded thanks to the newly collected data. Speeding-up this large-scale deep learning workflow has to be thought at all levels: algorithms, software infrastructure, frameworks, resource management, etc. The most critical bottlenecks and potential gains, however, are still not well understood.

3.1.2 Molecular Dynamics Simulations of Biological Assemblies (LBT)

Context: Numerical simulations on massively parallel architectures are used routinely to study the dynamics of biomolecules on the atomic scale. They produce large amounts of data representing the time trajectories of molecular configurations. These data points sample a high-dimension configuration space (dimension tens of thousands or more). Today's approaches for extracting knowledge from these massive data are still a challenge. Advanced data analytics could be transformative in the field if it could identify some inherent structure that is not captured by current methods.

Scenario: Laboratoire de Biochimie Theorique (LBT) is involved in simulation and analysis of complex biological systems and has been a long-standing partner of the Data-Move INRIA team. LBT is involved in the study of ion channels, typically membrane-inserted proteins that allow signals to be transmitted by transport of small charged molecules such as ions. HPC has lead to a steep increase in the generated trajectory datasets and LBT disposes of an extensive set of such simulations on ion channels. A combination high performance simulation, data analytics and machine learning is needed to fully exploit these data.

Data: Over the last 10 years, we have acquired many trajectories on pentameric ion channels that can be analyzed as a coherent body of samples. Trajectories comprise a broad variety of lengths, from ca. 20 nanoseconds to up to 10 microseconds, with snapshots typically saved in about 10s of picosecond steps. They originate from various sources: national supercomputer centre runs, Anton-generated data and one large PRACE-dataset. The system size is around 150 000 atoms on average, comprising the protein, the membrane, solvent and electrolyte.

Requirements analysis: Preliminary exploration carried out by the INRIA Data-Move team in concert with LBT showed untapped potential in Map/Reduce type approaches to tackle the scale-up in the trajectory analysis problem. Yet, enabling the analyses only represents the first step. Another challenge is to uncover the relationships and causalities: what caused a given change characterized by trajectory analysis? Can the system behavior be predicted? ML may provide an alternative to scientists' intuition to address these fundamental questions. The analysis of molecular data can be seen as a dimension reduction problem. Large volumes of molecular data in high dimension should be interpreted in a reduced space compatible with scientists' intuition of molecular processes. Many linear dimensionality reduction techniques have been applied to this problem, with varying degrees of success. More recently, nonlinear dimensionality reduction techniques such as diffusion maps have shown that the intrinsic dimension of the data varies locally, with a structure that can be interpreted as high-dimension basins connected by low-dimension pathways. So far these nonlinear methods have been isolated experiments; they have not yet gained a wide adoption, reflecting the fact that the appropriate target of ML algorithms in this domain remains an open question.

3.1.3 Computational Mechanics: Coupling Real Time Simulation and Learning (ESI-Group)

Context: Numerical simulation is nowadays present in most of scientific and engineering domains, making possible the virtual evaluation of systems responses and alleviating the number of experimental tests on the real system that the numerical model represents. The so-called *digital twins* take profits from the use of model order reduction (MOR) techniques and data assimilation to emulate the physical system from the accurate solution of the mathematical model expected describing it and identify online the model parameters in order to make model predictions as much as close the experimental measurements. Nevertheless, significant deviations between the predicted and observed responses can be noticed, limiting their use in many applications. This deviation can be decomposed in a biased contribution that can be associated to noise, and in an unbiased one that expresses hidden physics not taken into account in the physical model.

The hybrid twin that we propose, aims at taking advantage of the collected data in order to build on-the-fly a data-driven model that compensate the gap between the prediction and the measurement in order to ensure accurate predictions, making possible real-time control. The three main components of an hybrid-twin are: (i) a physically based model that represents the known physics, (ii) a data-driven correction built on the flight to fill the gap between predictions and measurements, and based exclusively on the collected data, making use of timely machine learning techniques and (iii) a control system.

Scenario: ESI-Group use case targets a composite forming process, namely the resin transfer molding (RTM). This process consists in injecting a polymer matrix into a mold in which a fibrous perform was placed. The proposed hybrid twin allows: (i) the inverse identification of the fibrous preform permeability online and in real-time and (ii) real-time control to adapt to eventual unexpected perturbations. For that purpose, we developed: (i) a physically based parametric reduced order model emulating the filling process for any possible perform permeability using the so-called Proper Generalized Decomposition – PGD – that allows performing in real time the inverse identification of the preform permeability, (ii) a data-driven based model that, using dictionary learning or deep learning based techniques, collect and assimilate data to adapt the predictions to evolving unexpected scenarios.

Data: The hybrid-twin requires the construction, enrichment and efficient manipulation of the process data-base (the so-called dictionary) to allow an efficient data-based construction of the divergence model.

Requirement analysis:

- Dictionary learning / deep learning techniques analysis to choose / develop the most accurate tool to perform real time construction of a data-driven correction model;
- Data completion and data-assimilation techniques
- Data-driven model constructor development based on the previously chosen techniques;

3.1.4 Machine Learning of Coherent Diffraction Data (ANL)

Context: X-ray coherent diffractive imaging (CDI) is a remarkable tool for seeing nanoscale materials. Classifying features such as defects in those materials, however, is difficult because state-of-the-art iterative algorithms for CDI phase retrieval are compute-intensive, and classification is done manually. The goal is to replace current compute- and human-driven methods with data-driven ones. The proposal studies whether atomic-scale defects such as dislocations as well as lattice positions in real space can be learned from diffraction patterns in reciprocal space without being defined analytically and without running expensive computations to reconstruct the real-space image.

Scenario: Argonne National Lab (ANL) workflow has been designed to extend to a variety of features in various X-ray imaging modes, but here we will focus on a specific problem by (a) classifying three types of dislocations and (b) deriving the atomic lattice positions in X-ray CDI of nanocrystals. The diffraction images required for training are generated from large-scale (billion atom) LAMMPS molecular dynamics (MD) simulations coupled with Debyer simulations that compute diffraction images from the MD models. A residual neural network (ResNet) is used for the dislocation classifier. For the atomic lattice positons, a reinforcement learning (RL) network generates crystal structures that LAMMPS uses to evaluate energy potential.

ANL uses Keras DL framework. An idea is to use Keras on top of the data processing frameworks (Damaris, FlowVR, Decaf) that will be investigated or proposed in this IPL project, to validate them.

Data: The training data sets will be derived entirely from molecular dynamics (MD) simulations obtained by minimizing the energy of machine-generated atomistic structures. Atomistic structures will be generated for perfect crystals, twinned crystals and crystals with edge and screw dislocations. The shapes of the crystals and the type, number and lengths of the defects will be randomized to sample a wide variety of configurations representative of real world structures. From each atomistic snapshot, the computed CDI pattern will be extracted by the Debyer simulation code. The training set will consist entirely of synthetically generated CDI pattern-label pairs. Experimental datasets will be available from ANL's science collaborators. Both modeling and experimentalists are available courtesy of an ongoing ANL internal project called AICDI (Artificial Intelligence for Coherent Diffraction Imaging) where Tom Peterka, invovled in this IPL, is the co-PI.

Requirements analysis: Speeding-up of the deep learning workflows has to be thought at all levels: algorithms, software infrastructure, frameworks, resource management, etc. To understand bottlenecks and inefficiencies, we will setup and conduct experiments for the overall workflow of this use case using existing frameworks for deep learning and data processing, analyze the logs and use the whole workflow as a baseline to compare and validate the outcome of the other work packages (algorithms, frameworks, resource management techniques).

3.2 Algorithms

Goal: explore how innovative ML-based algorithmic approaches can be combined with HPC based acceleration techniques.

Participants:HiePACS, Sierra

3.2.1 Stochastic Algorithms

Many of the common machine learning frameworks are based on some form of empirical risk minimization, where a certain error on the observed training set is minimized incrementally. The classical algorithm is *stochastic gradient descent* [17] with its recent accelerated versions based on variance reduction [15, 10].

These algorithms rely on accessing the data by blocks of observations. A key open problem when implementing these in an HPC framework is the optimal access to data with the minimal amount of communication and idle time. Here, relying on optimized frameworks for numerical linear algebra (which is a subcase of machine learning with square loss) provides a first avenue of progress.

3.2.2 Linear Algebra and Tensors

The design and implementation of efficient linear algebra algorithms fully exploiting the different processing units of the modern manycore nodes (CPU, CPU+GPU, ...) have been widely studied during the last decades. Novel algorithms for LU and QR factorizations, matrix-matrix and matrix-vector multiplications try to overcome the possible bottlenecks of new computer architectures (memory path, various levels of caches, inter processor or node network) by using various computing techniques loop blocking, matrix tiling, task-based programming over runtime systems to name a few. More recently, to decrease the memory footprint and to speedup those algorithms one introduced hierarchical matrices (HODLR, H-matrix, ...) to compress the data or randomized linear algebra (random projection, column or row selection, ...). Furthermore to tackle solution in higher dimensions, tensor calculations have been investigated that foster the study of new numerical schemes and associated high performance algorithms.

In machine learning, matrices and tensors come in two flavors: (a) data in form of tensors, (b) tensors and matrices created for other machine learning purposes.

Tensorial data come up in a variety of disciplines where often the dimensions have a physical meaning (e.g., space and time) such as in computer vision. Simple operations such as factorization are then challenging because of the combinatorial explosion in the order of the tensor [8].

Moreover, a classical line of work [9] has received increased attention [3] as they allow principled estimation for unsupervised learning for a wide variety of models. Starting from matrix data, higher-order tensors are built, whose factorization leads to the desired solution. Given that these tensors have particular structures, this opens up new interactions with scientific computing architectures.

3.3 Frameworks

Goal: explore new software architectures to combine Big Data tools and HPC.

3.3.1 Combining In Situ and In Transit Processing with Big Data Processing

Participants: KerData, Zenith, DataMove, ANL, LBT.

In the HPC area, the need to get fast insights from massive amounts of data generated by extreme-scale computations was addressed through a different means: *in situ* and *in transit* processing approaches were developed, allowing data to be visualized and processed in an interactive way, as they are produced, as opposed to traditional approach consisting of transferring the data off-site after the end of the computation, for offline analysis. In the Big Data analytics area, this challenge was addressed in a different way, leading to the emergence of *stream processing* approaches, allowing to leverage Big Data analytics tools on on data in-motion.

It becomes clear today that unifying the approaches developed in the two areas can be an efficient means addresses the Big Data challenges, which are now relevant for both HPC and BDA. We will explore the design of innovative architectures for data processing by exploring as follows.

1. We will explore design alternatives for an enhanced architecture that combines the strengths of various in situ processing frameworks for in situ processing designed by the IPL partners, e.g., Damaris (KerData), FlowVR (DataMove), Decaf (ANL); DataMove, KerData, LBT and ANL have planned to explore this direction in the framework of the INSITU ANR project (submitted to the 2017 call, now on a waiting list);

2. We will design a unified architecture which will combine in situ/in transit processing techniques with stream processing techniques and allow advanced tools for Big Data processing to be applied to HPC simulation data in real time.

The first direction will enable the design of an enhanced in situ processing framework; the second direction direction will bridge in situ/in transit processing with stream processing and enable the usage of standard Big Data analytics tools in this enhanced framework to analyze HPC simulation data.

Overall, the resulted unified framework will allow the processing of vast amounts of data at real-time for the considered use cases.

3.3.2 High Performance Deep Learning

Participants: RealOpt, Storm, Tau, SequeL, Zenith, TADaaM, DataMove, LBT, ESI-Group

There are several ways HPC can be leveraged by Deep Learning Applications.

- At the kernel level, more sophisticated algorithms to work on tensors can be developed, as proposed in Section 3.2.
- At the algorithm level, one way to obtain a parallel DL algorithm is to have several resources (GPUs or TPUs) working on the same level of the DL network. In this context, the problem is to have enough work to perform to keep the GPUs busy, what requires to consider large mini-batches, as proposed in [13]. Unfortunately, in general, considering larger mini-batches affects the convergence and the quality of the resulting network, and sophisticated techniques are required to adapt the mini-batch size to the learning stage (small at the beginning when the weights change a lot, larger at the end when everything has almost converged). We consider that this way to parallelize DL algorithm does not require a strong interaction between learning / HPC and BigData researchers, since the parallel algorithm is rather trivial, provided that difficult questions such as convergence and quality have been solved and that those questions are typically related to DL only.
- At the algorithm level, an other way to obtain parallelism is to have several computing resources working simultaneously on several convolution layers simultaneously. This is typically what TensorFlow [1] offers, by representing the DL network as a directed graph whose nodes represent convolution kernels and edges represent data dependences between convolution kernels. This is on this specific point that we propose to orient our work.

At the moment, the resource allocation described in [1] is rather simple. It is based on an estimation of the cost of each kernel on each type of resource. The convolution kernels are allocated to the different nodes using a traversal of the graph where dependencies are satisfied. At each step, a convolution kernel is allocated to the resource that is expected to complete it first, depending on its expecting cost and the expected transfer cost of its input data.

We strongly believe that this resource allocation algorithm can be improved along several directions. First, the resource allocation algorithm does not take into account the specificities of the application. Indeed, it is for instance very close that the by default StarPU scheduling algorithm [4] that is used for general task graphs. It has been proved that for specific applications such as linear algebra kernels, injecting some static knowledge based on a more sophisticated scheduling algorithm can strongly improved the performance of greedy algorithm [2].

Deep Reinforcement Learning also needs to execute multiple simulations during the learning process. Handling infrastructures such as the one used for AlphaGo Zero [18] is challenging and critical. We envision needs to train neural networks with even more complex simulations requiring each one to run in parallel on hundreds or thousands of compute cores, in addition to the compute resources needed for the neural network itself. This type of scenario is also relevant in scientific computing where DL can be coupled in situ with running larges scale simulations as in the use cases of ESI-Group, ANL and LBT.

We will address these issues first estimating the limits of current scheduling and resource allocation strategies such as Tensorflow [1], PyTorch [16], Caffe [14] and Theano [6], both in practice (what fraction of GPU time is lost using these strategies), before proposing new strategies based on specifically designed scheduling and resource allocation algorithms.

3.4 Infrastructure

Goal: develop novel infrastructure and resource management approaches for converged architectures.

Participants: TADaaM, KerData, Zenith, DataMove, ANL, ATOS/Bull.

A key aspect for performance of executing BigData applications onto HPC systems is to carefully manage the available resources: the storage system, the local memory, the local disk and buffers as well as the network. This is even more important when mixing different applications and workloads (i.e. HPC and BigData).

We will study the problem at different levels and granularity. First, it is necessary to provide ways for the applications to express their needs and describe their behavior. In relation with the other tasks of this IPL we need to know their usage of the different resources (compute, communication, storage, memory). Then, this input will be used by the batch scheduler to allocate the resources to the applications and workflow. We also want to be able to co-schedule applications which resource usage does not interfere too much. For instance, an application performing mostly I/O could be co-allocated with an application with an application requiring important compute resources.

At a lower level, we want to study how to schedule and partition access to local or intermediate storage (i.e. NVRAM, burst buffers, etc.) for the different applications. The problem is how to partition, at runtime, the storage and memory space both spatially and temporally between different applications depending on their behavior. On the same aspect, as a we are going to execute complex workflows, storage and memory need will evolved during execution. Therefore, we work on providing a mechanism for elastic storage and memory provisionning and allocation. This means that, depending on the state of the workflow the amount of resource dedicated to each application will vary. For this, we will need to modify the lower level of the software stack and especially the different runtime systems to enable cooperation of resource usage instead of the usual best-effort approach.

At a longer term, we will also work toward a convergent storage system. We want to provide an abstracted view of the different resources of the memory and storage hierarchy to enable each kind of application to see a unique system managing their data. Such system would meet the performance and the semantics required by each type of applications by enabling different kinds of data access (e.g. POSIX I/O vs. Database access) depending on the application requirements. A preliminary step in this direction has been taken based on the concept of transactional blobs illustrated by the Týr approach developed at KerData (PhD thesis of Pierre Matri). A collaboration with ANL is envisioned in the framework of this IPL (ANL-funded postdoc of Pierre Matri).

4 Expected Support

This IPL proposal is a 48 month project. We list below the expected human support for the first two years. It includes 3 PhDs, 1 postdoc and one engineer. For enabling our meetings, partner visits, a laptop for hired personal and 4 international missions per year, we will need an annual operating budget of 25Keuros. Needs for the 2 last years are sketched at the end of this section but will be further refined based on the progress performed during the first two years.

Scheduling Strategies for High Performance Deep Learning (2018-2021) Recently, several frameworks such as TensorFlow [1] and PyTorch [16] emerged and represent the DL network as a directed graph whose nodes represent convolution operations and edges represent data dependences between them. The goal of this **PhD thesis** is to work on how to allocate the convolution operations and how to schedule them to achieve a better efficiency, typically in the context of platforms consisting of heterogeneous resources such as GPUs and multicore nodes. We will combine static optimization and knowledge extraction to be used by the runtime (StarPU). In the context of DL, the same graph of convolution layers is used many times on different input data along the execution of the DL algorithm, what is close to the context of steady state scheduling [7], that has been proved to be more tractable than general scheduling. Another opportunity is to develop high level simulation techniques, that could be used in particular to detect bottlenecks with respect to a DL network and to a parallel architecture. This possibility is especially interesting in the context of DL, since it may help to redesign the network itself to cope with bottlenecks.

The PhD student will be localized in Bordeaux co-advised by Olivier Beaumont (RealOpt) and Alexis Joly (Zenith). The student will also benefit from the expertise of Samuel Thibault on on-line scheduling (Storm) and Guillaume Charpiat on deep learning (Tau). We will target the PlantNet use case [5].

Bridging in situ/in transit processing with Big Data analytics (2018-2021). The thesis goal is to propose an approach enabling HPC-Big Data convergence at the data processing level, by exploring solutions to build a unified framework for extreme-scale data processing. The framework will explore how the in situ/in transit data processing approaches originated in the HPC area can be used with Big Data processing techniques emerged in the BDA area (batch-based, streaming-based and hybrid). It will leverage previously-validated approaches including the Damaris framework for scalable, asynchronous I/O and in situ and in transit visualization and processing developed at KerData[6] and the KerA approach for low-latency storage for stream processing (currently under development at KerData, in collaboration with UPM and Huawei Munich). The resulted framework aims to support precise predictions in real-time and fast decision making, to be illustrated with the motivating use cases.

The PhD student will be based in Rennes, co-advised by Gabriel Antoniu, Alexandru Costan (KerData) and Patrick Valduriez (Zenith). This thesis

[6] https://project.inria.fr/damaris/

will involve a collaboration with Datamove and ANL and will specifically target the requirements of two of the IPL use cases: Pl@ntNet and Machine Learning of Coherent Diffraction data (ANL).

Analysis of Massive Molecular Trajectories through Deep Learning (2019-2022). A **PhD student** will work on a strategy to analyze high-dimension, massive data produced by molecular simulations to extract an intrinsic manifold that can be extrapolated to drive the detailed large scale simulation further in configuration space. Two limiting cases to be explored are 1) learning a parameterization of the local, low-dimension manifold and performing a geometric extrapolation, and 2) learning a reduced (coarse-grained) form of the system, as well as its dynamics, in order to predict its future states faster than the detailed dynamics can. In both cases, the reduced states extrapolated in time or configuration space will be used for steering a large scale simulations, thus accelerating sampling. This works will build on previous work between Bruno Raffin and Marc Baaden on interactive steering and in situ analytics of large scale molecular dynamics simulations [11, 12].

The student will be based in the Tau group and co-advised by Guillaume Charpiat (Tau) for ML, Bruno Raffin (DataMove) for HPC, and Jérôme Hénin (LBT) for molecular applications.

Parallel Scikit-Learn (2019-2020). An engineer will work on using the Dask distribution manager by Scikit-learn for the parallelization of its Machine Learning algorithms. He will also design a Python interface on top of the StarPU HPC runtime system, either as a replacement for Dask inside scikit-learn, or as a backend to Dask. We hope to reach convergence between the notions of tasks as expressed by Dask and tasks as expressed by StarPU. As a result, StarPU be usable as a runtime for Scikit-Learn. This will bring to StarPU new application perspectives for research on task and data scheduling in the context of big data and machine learning.

The engineer will be located in the Scikit-learn team at Parietal, co-advised by Gael Varoquaux (Parietal) and Samuel Thibault(Storm).

High Performance Deep Reinforcement Learning (2019-2020). The goal of **this postdoc** is to develop solutions for enabling efficient deep reinforcement learning at scale. This work will combine parallel dataflow approaches capable of supporting advanced adaptative interactions between the neural network and the simulations, for instance favoring asynchronous stochastic processes as they enable to expose more parallelism. This Postdoc is expected to start the second year of the IPL for a duration of 24 months. He/she will be **co-advised by Bruno Raffin (DataMove) and Philippe Preux (SequeL)**. SequeL is specialized on reinforcement learning, developing algorithms and applications that require an increasing amount of compute resources. For instance around 2005, CrazyStone revolutionized the world of go player: in the early 2010, CrazyStone required weeks of training on Grid 5K. Today SequeL is working on the acquisition of the ability to dialogue in natural languages which takes weeks of training on a couple of GPUs[7]. DataMove has experience with large scale dataflow infrastructures through the FlowVR [12] or Melissa frameworks running multiple simulations on more than 30K cores for instance [19].

The resources needed for the two last years of the IPL will likely include engineering support for the integration and validation of complex application scenarios. We are also interested in the distribution of data in the context of a converged system, in which some nodes of an HPC system are provided with a nonvolatile storage. These questions have already begun to be looked at from the point of view of the commission and de-commission of the nodes as part of Nathanel Cherriére's thesis in KerData and should be looked at from an algorithmic point of view in a postdoc funded within the framework of a Bordeaux IdEx project. An associated problem related to replicated storage is to adapt the schedulers (e.g. OAR) to take into account the locality of the data at the time of the allocation of computing resources. This topic will be subject to resource requests in 2020. The algorithm axis may also need to be supported. The collaboration between HPC and ML teams is just starting through this IPL. We will pay attention to novel emerging topics that may emerge during the first two years of join work.

5 Organization

Bruno Raffin is the PI for this IPL and will oversee the activity of the project. He will be assisted by an advisory board composed of Gabriel Antoniu, Emmanuel Jeannot, Patrick Valduriez, Gael Varoquaux, Philippe Preux, as well as Jean Roman and Frédéric Desprez as DGD-S representatives. Given the large size of this IPL, not all teams are represented to keep a reactive IPL management structure. The board members have been chosen to keep a good geographic and thematic coverage. External partners not being financially support are not included. The advisory board will meet every 6 months during the general meetings (see bellow). Its role will be to review the IPL activity an eventually provide guidance for the IPL management and the envisioned actions.

General meetings will be held every 6 months, often associated with coding/paper/proposal sprints focused on specific issues and goals. General conf-calls will take place every month. Young scientists will visit partner teams for week-long stays or even month long stays. Coding sprints will be organized when needed. They are extremely efficients to unlock issues, boost code development and tighten the collaboration. Grid'5000 will be a key experimental platform for this IPL and we will establish a tight collaboration with the Grid'5000 team (external partner of this IPL). New Grid'5000 equipments (fall 2017) will well fit our needs: a high performance cluster with on-node storage dedicated to BigData/HPC experiments at Grenoble, Lille platform augmented with last generation GPUs).

[7] https://www.guesswhat.ai

6 Outreach

Beyond joint publications in top-level international venues of the HPC and Big Data areas, we will intend to leverage the results of the project in the following ways:

- Submissions to ANR and H2020 projects (e.g., future calls like FETHPC-02-2019 includes the subtopic *Supercomputing for Extreme Data and emerging HPC use modes*)
- Co-organize national and international open events: workshops, tutorials, panels, summer schools, hackatons, etc.
- Create a framework for regular interactions through existing workshops (e.g., within JLESC) and potentially new ones, involving the national HPC, Big Data processing and machine learning communities involved in the corresponding challenge of Inria's strategic plan, inspired by the international BDEC workshop series model.
- Enforce/develop new research directions for participating teams, dedicated to open issues related to the HPC-Big Data convergence process.
- Use the expertise gained in this project to provide strong and relevant contributions to international and European forums such as ETP4HPC and BDVA, which influence the future agendas of the European Commission in the HPC and Big Data areas.
- Participate to define the future architecture needs, in particular AI machines, in the prospect of new acquisitions at the INRIA, G5K, national or European level.

7 References

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 12, 13, 14

[2] E. Agullo, O. Beaumont, L. Eyraud-Dubois, and S. Kumar. Are static schedules so bad? a case study on cholesky factorization. In *Parallel and Distributed Processing Symposium, 2016 IEEE International*, pages 1021–1030. IEEE, 2016. 12

[3] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014. 10

[4] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier. Starpu: a unified platform for task scheduling on heterogeneous multicore architectures. *Concurrency and Computation: Practice and Experience*, 23(2):187–198, 2011. 12

[5] D. Barthélémy, N. Boujemaa, D. Mathieu, M. Jean-François, A. Joly, and E. Mouysset. The pl@ ntnet project: plant computational identification and collaborative information system. 2011. 14

[6] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012. 13

[7] O. Beaumont, A. Legrand, L. Marchal, and Y. Robert. Steady-state scheduling on heterogeneous clusters. *International Journal of Foundations of Computer Science*, 16(02):163–194, 2005. 14

[8] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009. 10

[9] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010. 10

[10] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 10

[11] M. Dreher, M. Piuzzi, A. Turki, M. Chavent, M. Baaden, N. Férey, S. ebastien Limet, B. Raffin, and S. Robert. Interactive Molecular Dynamics: Scaling up to Large Systems. In *International Conference on Computational Science (ICCS)*, June 2013. 15

[12] M. Dreher and B. Raffin. A Flexible Framework for Asynchronous In Situ and In Transit Analytics for Scientific Simulations. In *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Chicago, United States, May 2014. IEEE Computer Science Press. 15, 16

[13] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 12

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 13

[15] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013. 10

[16] N. Ketkar. Introduction to pytorch. In *Deep Learning with Python*, pages 195–208. Springer, 2017. 13, 14

[17] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statistics*, 22:400–407, 1951. 10

[18] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017. 5, 12

[19] T. Terraz, A. Ribes, Y. Fournier, B. Iooss, and B. Raffin. Melissa: Large scale in transit sensitivity analysis avoiding intermediate files. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC'17)*, Denver, 2017. 16