



Exploiting GPUs on distributed infrastructures for medical imaging applications with VIP and DIRAC

Sorina Camarasu-Pop, Carole Lartizien, Thomas Grenier, Axel Bonnet, Pascal Wassong, Vanessa Hamar, Fabio Hernandez, Luisa Arrabito, Johan Bregeon, Pierre Gay, et al.

► To cite this version:

Sorina Camarasu-Pop, Carole Lartizien, Thomas Grenier, Axel Bonnet, Pascal Wassong, et al.. Exploiting GPUs on distributed infrastructures for medical imaging applications with VIP and DIRAC. 42nd international convention on information and communication technology, electronics and micro-electronics, May 2019, Opatija, Croatia. pp.190-195, 10.23919/mipro.2019.8757075 . hal-02327867

HAL Id: hal-02327867

<https://hal.science/hal-02327867>

Submitted on 4 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exploiting GPUs on distributed infrastructures for medical imaging applications with VIP and DIRAC

Sorina Camarasu-Pop*, Carole Lartizien*, Thomas Grenier*, Axel Bonnet*, Pascal Wassong*, Vanessa Hamar**, Fabio Hernandez**, Luisa Arrabito***, Johan Bregeon***, Pierre Gay**** and Andrei Tsaregorodtsev*****

* Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, LYON, France

** CNRS, CC-IN2P3, 21 avenue Pierre de Coubertin, CS70202, 69627 Villeurbanne cedex, France

*** Laboratoire Univers et Particules de Montpellier - Université de Montpellier - CNRS/IN2P3, Place Eugène Bataillon - CC 72, 34095 Montpellier Cédex 05 France

**** Université de Bordeaux, France

***** Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France
Sorina.pop@creatis.insa-lyon.fr

Abstract - GPU usage has become mandatory for the processing of (3D) medical data, as well as for efficient machine learning approaches such as deep learning. In this contribution, we present how VIP and DIRAC can be leveraged to run medical image processing applications on distributed computing resources equipped with GPUs.

VIP (Virtual Imaging Platform) is a web portal for the simulation and processing of massive data in medical imaging. VIP users can access applications as a service and significant amounts of computing resources and storage with no required technical skills beyond the use of a web browser. VIP relies on the DIRAC (Distributed Infrastructure with Remote Agent Control) interware for scheduling tasks for execution on distributed infrastructures such as grid, clouds, and local clusters. New applications are regularly integrated into VIP/DIRAC. They all have their own requirements, among which GPU usage is more and more frequent.

This contribution will give an overview of the targeted medical applications and their requirements, as well as technical insights on how VIP and DIRAC allow end users to efficiently exploit GPU resources with no specific knowledge about the underlying distributed infrastructure.

Keywords - medical imaging, distributed computing, GPU

I. INTRODUCTION

Medical imaging applications have always been compute intensive and, in the last few years, GPU usage has become mandatory for the processing of (3D) medical data, as well as for efficient machine learning approaches such as deep learning.

Researchers in the biomedical and life-science communities have had access to distributed infrastructures such as EGI for many years, but they were usually “limited” to CPU-based resources. The scientific

gateways available, as well as other tools allowing researchers to deploy their scientific applications, were therefore tailored to the available computing resources.

With the growing interest in GPUs, we try to answer the question of what it takes to have a medical imaging application available as a web service and executed on distributed computing resources equipped with GPUs.

The paper is organized as follows: Section II introduces the EGI infrastructure, as well as DIRAC and VIP. Section III gives an overview of two medical imaging applications, their implementation and specific needs. Section IV provides technical insight on how these applications can exploit GPU resources through VIP and DIRAC. Finally, Section V is dedicated to conclusions and perspectives.

II. DISTRIBUTED INFRASTRUCTURES AND THEIR TOOLS

A. EGI

Distributed computing has been available for various research fields since the early 2000s through the European DataGrid Project, which became the European Grid Infrastructure (EGI). During the last 20 years, this infrastructure has continuously evolved, proposing a large spectrum of services alongside computing and data resources. Today, EGI provides access to over 850,000 logical CPUs and 650 PB of disk and tape storage. These resources are shared among different research fields, named Virtual Organizations (VOs).

The Biomed VO, gathering researchers from the biomedical and life-science communities, has access to about 65 computing sites world-wide, including more than 130 computing clusters and 5PB of storage distributed across 50 different storage elements (SEs).

Although most computing resources are composed of “classical” CPUs, some sites are also providing GP-GPU resources through the EGI Federated Cloud. The number and heterogeneity of available resources make it difficult for researchers (end users) with no specific knowledge of the underlying infrastructure to use these resources easily and efficiently.

B. DIRAC

The DIRAC Project provides software for building large scale distributed computing systems [4], [7]. Services created using this software allow to integrate computing and storage resources of different types located anywhere in the world in a coherent federation with a single access point from the user perspective. The Project was initially started to exploit available grid computing resources for LHCb, one of the experiments on the Large Hadron Collider (LHC) at CERN, Geneva, as a response to a challenge of processing unprecedented data volumes collected at LHC. The solution developed for LHCb was later adopted by other experiments in the High Energy Physics, Astrophysics and other scientific domains. Services based on the DIRAC software are also offered to the users of large national grid infrastructures, e.g. France-Grilles National Grid Infrastructure (NGI) in France. Since 2014, the DIRAC services are also provided for the users of the European grid infrastructure (EGI). As a result, even small user communities can benefit from an easy access to vast amounts of resources available in these infrastructures.

DIRAC services include both Workload and Data Management tools. The Workload Management Service (WMS) integrates resources provisioned by grid and cloud infrastructures but also by supercomputers, standalone computing farms or even volunteer computing systems. User tasks are submitted to the central Task Queue and then sent for execution to the resources matching the task requirements, taking into account the task priorities and resource usage policies. Special attention is paid to automation of the management of large numbers of tasks that can be organized in complex workflows driven by the availability of the data to be processed. Users access the service using several interfaces – command line tools, Web Portal, RESTful interface or programming API. The work pattern is similar to the one of using a large-scale batch system distributed worldwide. For example the DIRAC WMS of the LHCb experiment is capable of executing of up to 150K user tasks simultaneously, effectively operating a virtual computing center distributed over more than 120 sites over the world with a total capacity of 150K computing cores. The Data Management System (DMS) of DIRAC provides access to different kinds of data storage systems with virtually any access protocol existing in infrastructures supporting scientific research. The File Catalog service keeps track of all the physical

copies of existing files and provides means for user defined metadata which allows efficient selection of datasets for a specific user analysis task. Support for massive data operations is also provided. In the whole, users perceive various computing and storage resources operated by the DIRAC services as a single large computer with a large distributed file system.

The DIRAC software architecture is conceived to be used by various applications in different scientific domains. The modular structure of the software allows for easy extensions by developing additional components specific to workflows of particular user communities using the same software framework.

C. VIP

The Virtual Imaging Platform [1], [8] (VIP¹) is a web portal for medical simulation and image data analysis. It leverages resources available in the Biomed VO of the EGI e-infrastructure to offer an open service to academic researchers worldwide. In January 2019, VIP counts more than 1000 registered users, some 20 applications and 44 publications made by VIP users with results obtained through VIP.

Until recently, applications were manually integrated in VIP as workflows written in the Gwendia language. Workflow languages provide a number of valuable advantages such as the concatenation of multiple applications, as well as different iteration strategies on input list. Nevertheless, they require a certain level of expertise which made porting applications to VIP a costly human effort. In order to ease the integration of new applications, we now rely on a formal application description as proposed by Boutiques². To do so, we developed an importer allowing to automatically generate the application workflow and associated script files based on the Boutiques descriptor.

VIP uses the French national instance of the DIRAC service³ for job submission. Based on the application description, VIP generates a form to be filled-in with inputs from the end user. VIP then prepares the jobs to be submitted and handled by DIRAC. The complexity of the infrastructure is thus hidden to the end user, who only has to provide the inputs for the application.

III. MEDICAL IMAGING APPLICATIONS AND THEIR REQUIREMENTS

A. Multimodality neuroimaging and machine learning

The analysis of multimodality neuroimaging data (MRI, PET) is increasingly used in diagnostic and pre-surgical work-up of patients suffering neuropathologies.

¹ <https://vip.creatis.insa-lyon.fr/>

² <https://github.com/boutiques>

³ <https://dirac.in2p3.fr/DIRAC>

Computer-aided diagnosis (CAD) systems have been introduced to assist clinicians in various tasks such as brain structure or lesion detection and segmentation. The vast majority of the existing CAD systems are built upon machine learning methods developed in supervised settings. Such systems benefit from the available data sets (usually) annotated at voxel-level, and output maps where each voxel is characterized by a score discriminating healthy versus pathological voxels. Supervised learning, however, cannot be effectively applied when the number of pathological cases in the training set is not sufficient to account for the complexity of the task. This is often the case when it comes to detecting some brain pathologies such as epilepsy, when the lesions are subtle and vary largely in terms of shape and texture. It is not trivial to obtain a well-annotated data set to represent such variability. To bypass the problem of insufficient labelled data, we cast epilepsy lesion detection task as an outlier detection problem. Such an approach solely needs a training set of non-pathological images, hence no labelled data is required. The novel CAD architecture that was proposed in [6] consists in learning an oc-SVM model for each voxel in the brain volume using automatically learnt representations based on unsupervised deep neural

networks, as shown on Fig. 1. This system was trained on 75 healthy subjects and tested on 21 epilepsy patients whose MRI exams are considered negative, meaning that the radiologist could not detect the lesion. The scientific details and results of this study are reported in [5] [6].

Fig. 1 depicts the processing pipeline, which includes (i) an image pre-processing step using Matlab (SPM) tools and (ii) the main processing tools, corresponding to feature extraction and classification, developed in Python. The feature extraction, which is compute intensive, uses the Theano library and runs on GPUs.

One of the aims of the project presented here is to build a demonstrator of this CAD system on the VIP platform. This web application should allow users with no specific knowledge on distributed computing to perform different CAD-related tasks, such as:

- Test a given CAD model based on provided reference sample images.
- Process their own data with the entire CAD pipeline, including the preprocessing, feature extraction and classification steps.

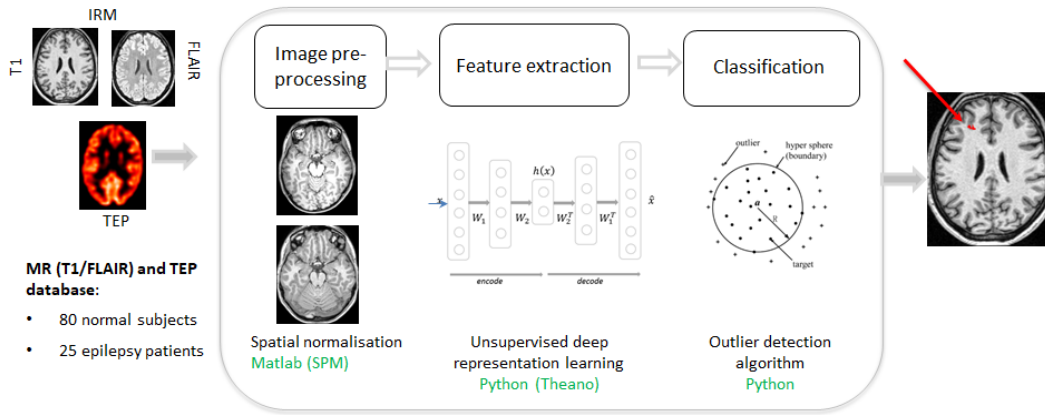


Figure 1. CAD-Epilepsy pipeline

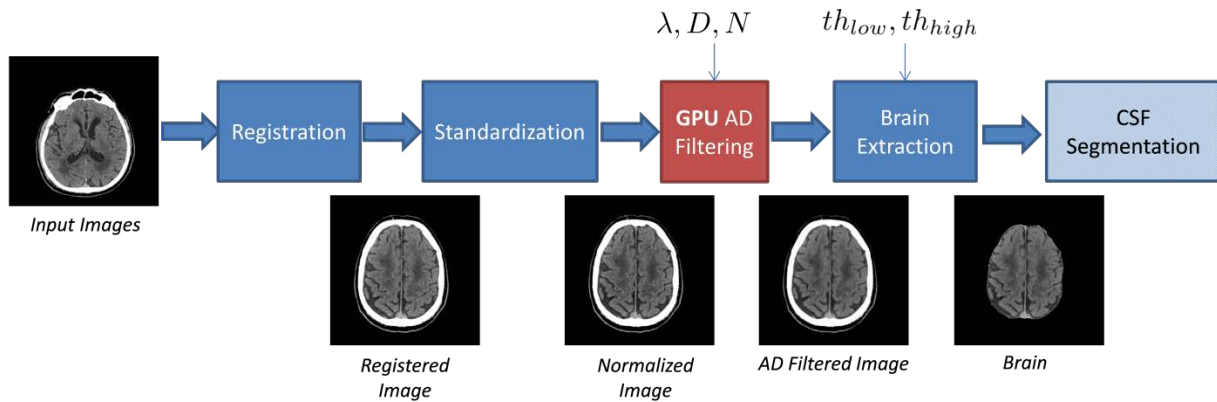


Figure 2. Pipeline, data and important parameters for the CSF segmentation

B. GPU-based filtering for 3D CT Scan

In the context of the study of cerebrospinal fluid (CSF) volume changes across time, we have designed a pipeline able to process robustly 3D CT brain volumes of a given patient. These longitudinal studies consist of 4 or 5 time points each with a 3D CT scan. The pipeline applied on data of a patient is described in Fig. 2. After a registration on a template patient (with a given orientation and resolution), we apply a grey-scale standardization using the first acquisition as baseline, and then we filter the data using an Anisotropic Diffusion approach. Finally, we perform a brain extraction and start the CSF segmentation (these two steps are not described here as they are parts of active research now).

For the preliminary study, the data set consists of 5 patients. The aim is to optimize the parameters' value involved in each step of the pipeline in order to obtain the most suitable parameters for this data set, expecting that they will also provide good results for the hundreds of coming patients.

Basic exhaustive search has been done for the registration step. The standardization step does not require any parameters. Our brain extraction step only needs two thresholds (thlow and thhigh) and we have observed that the most crucial step for the final CSF segmentation is the filtering step.

According to state of the art of filtering approaches and considering the CT scans' noise specificity and our computational time cost constraints, we adopt anisotropic diffusion [3]. This approach aims at smoothing drastically areas without any edges and doesn't process too much edges thus preserves them. It is based on the anisotropic equation:

$$\frac{\partial I(x, t)}{\partial t} = C(x) \nabla_x I(x, t) + \nabla_x C(x) \nabla_x I(x, t)$$

Where $I(x, 0)$ is the input image, $I(x, t)$ is the image at time t , and $C(x)$ is a function that weights the diffusion such that it tends to 0 when x is close to an edge (which implies no smoothing) and around 1 when the neighbourhood of x is on a piecewise constant region. The function used for $C(x)$ is:

$$C(x) = \exp\left(-\frac{\|\nabla I(x, t)\|^2}{D^2}\right)$$

with D the parameter selecting the amount of variation that can be considered as edges.

This equation is solved iteratively. Three parameters must be adjusted carefully to obtain interesting results: D , the time step Δt and the number of iterations N . The iterative process is stable if the time step is chosen below a value calculated from the pixel spacing of the data and the number of dimensions of the data (3 in our case).

Thus, we have to search D and N providing the best results.

This search is performed in an exhaustive way and we opt for the ITK implementation on GPU that is at least 100 times faster (with a Nvidia GeForce GTX 1070Ti) than the CPU implementation (run concurrently on 8 Intel(R) Xeon(R) CPU E5-2403 0 @ 1.80GHz cores).

The proposed work, coupling VIP and DIRAC would allow such parameters optimization. Even if preliminary, Fig. 3 shows results of the whole pipeline using optimal parameters for the given segmentation approach and Fig. 4 shows a segmentation result of a 3D CT Scan.

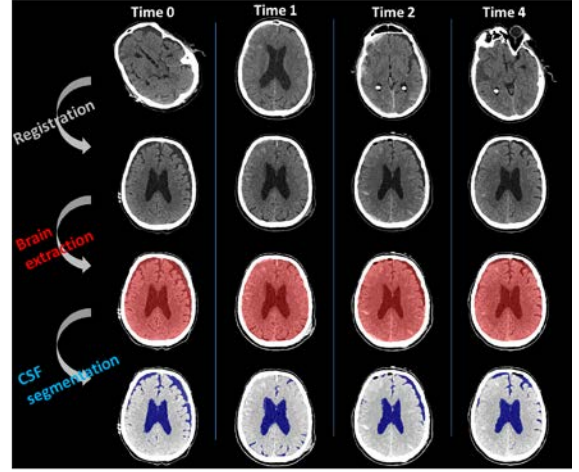


Figure 3. Results summary of the parameters optimized pipeline on a 4 time points study

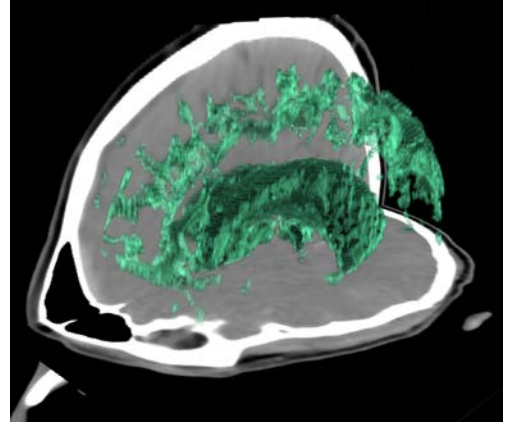


Figure 4. Example of a 3D CSF segmentation

IV. APPLICATION INTEGRATION AND DEPLOYMENT WITH VIP AND DIRAC

A. Containers

Containers have lately become one of the most popular ways of packaging, deploying and executing scientific applications in distributed environments. They have a number of advantages, such as reproducibility,

isolating application processes and ease of use. With containers, end users can easily pack their own applications with the precise versions of the libraries they depend on. Moreover, they can build their applications on their desktops and run hundreds or even thousands of instances on remote CPUs or GPUs.

Docker has become synonymous with container technology because it has been the most successful at popularizing it, but other container technologies exist, such as Singularity and CharlieCloud. As a matter of fact, Singularity has gained enormous popularity in the scientific and HPC world. Singularity containers differ from Docker containers in several important ways, including the handling of namespaces, user privileges, and the images themselves. The excellent news is that “Singularity is good friends with Docker.”⁴, which allows us to build Docker images and then easily convert them to Singularity if needed.

For the deployment of the CAD Epilepsy applications, we deployed Docker images created using an existing Nvidia image with cuda and cuDNN already installed.

B. Boutiques

Boutiques [2] is a cross-platform descriptive command-line framework for applications. It relies on Linux containers to facilitate the application installation and sharing, and it uses a versatile JSON format to describe the command-line template, inputs and outputs.

```
{
  "name": "epilepsy-test",
  "tool-version": "1.0",
  "description": "Run Epilepsy test, using launcher script",
  "command-line": "launch-train.sh [INPUT] [OUTPUT]",
  "schema-version": "0.5",
  "container-image": {
    "type": "docker",
    "image": "feature-extraction"
  },
  "inputs": [
    {
      "id": "image",
      "name": "Input image",
      "type": "String",
      "description": "Image file to test",
      "value-key": "[INPUT]",
      "list": false,
      "optional": false,
      "default-value": ""
    }
  ],
  "output-files": [
    {
      "id": "result",
      "name": "Result image file",
      "description": "Result image file",
      "value-key": "[OUTPUT]",
      "path-template": "[INPUT].tar.gz"
    }
  ]
}
```

Figure 5. Boutiques JSON descriptor to define the command-line, inputs and outputs of the application

Fig. 5 shows a simple JSON descriptor we created for the “Epilepsy test” application. It allows to (i) point to the Docker image “feature-extraction” we created previously,

(ii) define the command-line to be executed and (iii) define its input and output.

C. DIRAC SSH Computing Element

Access to different types of computing resources (standalone batch systems, grids, clouds) is provided by the DIRAC services. In the DIRAC framework an abstract “Computing Element” class encapsulates all the necessary functionality to submit tasks to a computing resource, monitors their progress and gets back the results. Various implementations of this abstract interface are provided, for example: ARCComputingElement for the ARC middleware⁵, CREAMComputingElement for CREAM⁶, etc. SSHComputingElement is the implementation allowing access to a standalone computing farms through an SSH tunnel. It can be specialized to use various types of batch systems, e.g. SLURM, OAR, Torque or GE. For each computing resource specified in DIRAC configuration, the type of Computing Element must be defined together with its general properties - operating system, memory, number of processors, system limits and so on.

In our particular case, the Biomed VO has access to GPU resources at the site savba.sk through an SSH connection. In the DIRAC configuration, a dedicated computing element of the type SSH must be defined. A specific tag can be associated to this computing element in order to restrict its usage to jobs that request GPUs. This prevents GPU resources from being wasted with regular CPU jobs.

This resource was added to the France-Grilles’ DIRAC service configuration, with special care taken to provide confidential configuration parameters that are stored in local protected files in the machine where the SiteDirector, the DIRAC job submission agent, is running.

From the users’ point of view, they only need to add the tag: “RequiredTag = nvidiaGPU” to the job description and the jobs will be executed at the sites where the corresponding resources are available. Since in this case jobs are created and submitted to DIRAC through VIP, the job description, including this specific tag, is entirely handled by VIP service.

Fig. 6 shows the cumulative number of CAD-Epilepsy jobs executed by DIRAC on GPU resources.

⁴ <https://singularity.lbl.gov/docs-docker>

⁵ <http://www.nordugrid.org/arc/>

⁶ <https://cream-guide.readthedocs.io>

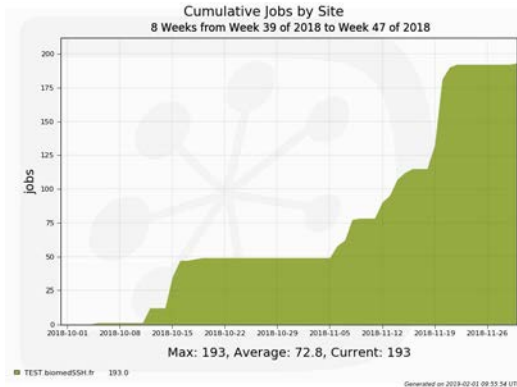


Figure 6. Cumulative CAD-Epilepsy jobs executed by DIRAC on GPU resources

V. CONCLUSIONS AND PERSPECTIVES

Scientific applications have specific needs to which infrastructures and technologies bring multiple solutions, contributing to a very rich and heterogeneous environment. In this paper, we described an end-to-end system enabling medical imaging researchers to use distributed computing resources with GPUs for their specific applications.

VIP, as a web portal and scientific gateway, makes applications available as a service. Boutiques greatly facilitates application integration into VIP, while containers make it possible to easily deploy complex applications on distributed infrastructures. The DIRAC interware allows for an efficient exploitation of distributed computing resources (CPUs or GPUs), despite their heterogeneity.

A first version of the CAD-Epilepsy demonstrator is available for testing, but further work will allow to optimize certain aspects, such as the handling of large data files. The feature extraction tools, which need to execute on GPUs, produce large volumes of data (tens of GB per training model), which are further needed for testing/prediction jobs. The latter are less compute intensive and could be executed on simple CPUs provided that the previously produced data are available. The data needs thus to be transferred as efficiently as possible.

GPU resources are still scarce compared to “classical” CPUs. In order to optimize the use of GPU resources, we plan to further split the current Epilepsy applications into smaller tools that can efficiently run on CPUs from those that need a GPU. By using application workflows at the VIP level, this kind of splitting is transparent for the end user and allows DIRAC to submit the jobs to the correct type of resources.

Last but not least, we would also like to go further into testing the limits of the compatibility and reproducibility

provided by container technology with respect to the ever evolving GPUs cards, drivers and libraries.

ACKNOWLEDGMENTS

Part of this project related to the CAD system for epilepsy lesion detection received funds from the French Foundation for Research in Epilepsy (FFRE) as part of the program ‘INTERFACE 2017’. It was performed within the framework of the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program “Investissements d’Avenir” (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

The authors would like to thank France Grilles, EGI and the Institute of Informatics, Slovak Academy of Sciences for access to GPU resources and technical support.

REFERENCES

- [1] T. Glatard, C. Lartizien, B. Gibaud, R. Ferreira da Silva, G. Forestier, F. Cervenansky, M. Alessandrini, H. Benoit-Cattin, O. Bernard, S. Camarasu-Pop, N. Cerezo, P. Clarysse, A. Gaignard, P. Hugonnard, H. Liebgott, S. Marache, A. Marion, J. Montagnat, J. Tabary, and D. Friboulet, “A virtual imaging platform for multi-modality medical image simulation,” *IEEE Transactions on Medical Imaging*, vol. 32, no. 1, pp. 110–118, 2013.
- [2] T. Glatard, et al., “Boutiques: a flexible framework to integrate command-line applications in computing platforms,” *GigaScience*, vol. 7, no. 5, 03 2018. [Online]. Available: <https://dx.doi.org/10.1093/gigascience/giy016>
- [3] Pietro Perona and Jitendra Malik, “Scale-space and edge detection using anisotropic diffusion”, *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 12, pp. 629-639, 1990.
- [4] A. Tsaregorodtsev, “DIRAC Distributed Computing Services”, *J. Phys.: Conf. Ser.* 396 032107
- [5] Z. Alaverdyan, J. Chai et C. Lartizien (2018). « Unsupervised Feature Learning for Outlier Detection with Stacked Convolutional Autoencoders, Siamese Networks and Wasserstein Autoencoders: Application to Epilepsy Detection”. 4th International Workshop on Deep Learning in Medical Image Analysis (DLMIA) held in conjunction with MICCAI 2018, Granada, Spain.
- [6] Z. Alaverdyan, J. Jung, R. Bouet et C. Lartizien (2018). « Regularized siamese neural network for unsupervised outlier detection on brain multiparametric magnetic resonance imaging: application to epilepsy lesion screening”. International conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam.
- [7] A. Casajus, R. Graciani, S. Paterson and A. Tsaregorodtsev, “DIRAC pilot framework and the DIRAC Workload Management System”, IOP Publishing, 2009
- [8] T. Glatard, et al., “Software architectures to integrate workflow engines in science gateways”, *Future Generation Computer Systems*, vol. 75, pp. 239-255, 2017