Received 30 April 2018; | Revised 09 November 2019; | Accepted xxx
DOI: xxx/xxxx

ARTICLE TYPE

Multimedia Processing using Deep Learning Technologies, HPC Cloud Resources and Big Data Volumes

Sidi Ahmed Mahmoudi*[1] | Mohammed Amin Belarbi[1] | Saïd Mahmoudi[1] | Ghalem Belalem[2] | Pierre Manneback[1]

[1]Computer Science Unit, Faculty of engineering, UMONS, Mons, Belgium
[2]Computer Science Department, Faculty of Exact and Applied Science, University of Oran 1, Oran, Algeria

Correspondence
*Sidi Ahmed Mahmoudi, UMons, Place du Parc 20, B-7000 Mons, Belgium. Email: sidi.mahmoudi@umons.ac.be

Present Address
University of Mons, Computer Science Unit, 9 rue de Houdain, B-7000 Mons, Belgium

Summary

The last few years have been marked by the presence of very large sets of images and videos in our everyday lives. These multimedia objects have a very fast frequency of creation and sharing since images and videos can come from different devices such as smartphones, satellites, cameras or drones. They are generally used to illustrate objects in different situations (public areas, train stations, hospitals, political and sport events and competitions, etc.). As consequence, image and video processing algorithms have got increasing importance for several computer vision applications that should be adapted for managing large-scale volumes and exploiting high performance computing resources (local or cloud). In this work, we propose a cloud-based toolbox (platform) for computer vision applications. This platform integrates a toolbox of image and video processing algorithms that can: 1) exploit HPC cloud resources, 2) execute applications in real time 3) manage large-scale database using Big Data technologies. The related libraries and hardware drivers are automatically integrated and configured in order to offer to users an access to the different applications without the need to download, install and configure software or hardware. Experiments were conducted using three kinds of applications: 1) image and video processing applications. 2) Deep learning techniques for images classification and multi-object localization 3) images indexation and retrieval. These experiments demonstrated the interest of our platform for sharing, in an efficient way, our scientific contributions and annotated databases in order to improve the quality and performance of computer vision applications.

KEYWORDS:
Multimedia processing, High performance computing, Cloud computing, Big Data, Deep Learning, Images indexation and retrieval.

1 | INTRODUCTION

In computer science domain, we noticed in the last few years a great evolution of processors, which is mainly due to the change of central processing unit (CPUs) architectures. This changement offered the possibility to benefit from the new processors performance as a result of increasing the computing units' number within processors. This multiplication of computing units is present in different types of calculators such as grids, clusters, Graphic processing units (GPUs), Tensor Processing Units (TPUs), Multi-CPU or/and Multi-GPU machines, etc.

In literature, one can find different image and video processing applications that tend to exploit the high computing power of these new calculators. This exploitation is mainly performed by the development of parallel implementations that exploit the high number of processing units.

We note that multimedia processing methods, and mainly features extraction algorithms, fit naturally with parallelization within Multi-CPU or/and Multi-GPU architectures since they consist of a common calculation over many pixels. The process of features extraction presents a tremendous step for each multimedia processing application since it allows to define parameters and features for several tasks such as: image (or video) classification, object localization, shape recognition, images segmentation, event detection, event localization, etc. The methods of features extraction can use two kinds of approaches: 1) Classic image processing algorithms that consist on analyzing images pixels in order to extract the main image or video features. 2) Convolutional neuronal networks (CNN) that allow to extract features based of the different convolutional layers of Deep neural networks.

In this context, several solutions have been proposed, recently, for exploiting the above-mentioned materials. Although they benefit from the high potential of multi-core processors (Multi-CPU or/and Multi-GPU, grid, etc.), the exploitation of cloud resources is not efficient since the number of reserved computing units is not depending to algorithms requirements. On the other hand, the use, configuration and exploitation of multimedia processing solutions (that can use classic or/and deep learning approaches) in not so easy. In this case, users must dispose locally of the required material (CPUs, GPU, etc.) and need to download, install and configure the related modules and libraries that are required for execution. Moreover, the proposed solutions in literature are not adapted for processing large scale image databases that require a parallel and distributed treatment in the same time.

Therefore, we propose a cloud-based platform that integrates multimedia features extraction toolbox exploiting both classic image processing methods and convolutional neuronal networks. The proposed platform integrates also an application of image indexation and retrieval within large-scale databases using the Hadoop framework. Within our platform, each user can select the required algorithm, load its data and retrieve results with a responsive environment similar to desktop either if the required application exploits parallel (GPU) or heterogeneous (Multi-CPU/Multi-GPU) platforms. The selection of cloud resources is performed in an efficient way based on analyzing the requirement of selected algorithms and users' input data. This selection allows to reduce the computation time, the consumed energy and the cost of cloud reservation.

The integrated methods are provided in a secure way where each required library or hardware driver is automatically installed and configured. This offers an easier access to our algorithms without the need to download, install and configure software and hardware. Otherwise, the different applications, integrated in the cloud platform, can be accessed from multiple users thanks to the use of Docker[1] containers and images. In terms of security, the connections of users are secured within SSL[2] protocol, the exchanged data are secured within SFTP[3] protocol and encrypted within an efficient method of data compression. The exchanged data are duplicated within Hadoop[4] framework in order to keep a backup of our data in case of crash.

The remainder of the paper is organized as follows: Section 2 presents the related works, while the third Section is devoted to detail our cloud-based platform for multimedia processing and retrieval, using classic and deep learning approaches, within large-scale data bases. Experimental results are given in Section 4. Finally, conclusions and future works are presented in the last Section.

2 | RELATED WORKS

In literature, we can categorize two kinds of works related to our approach of large-scale multimedia processing using cloud resources: 1. Computer vision cloud platforms. 2. Big Data frameworks for multimedia processing.

2.1 | Computer vision cloud platforms

During the last years, the development of computer vision frameworks has been influenced by the high emergence of cloud computing platforms, which are hosted by famous companies such as Amazon Web Services, Google Cloud Platform, Microsoft Azure, etc. These platforms helped developers to produce several cloud interfaces with a high abstraction of the complexity behind computer vision applications. Indeed, they provide an easier access of high computing power without the need to adapt the related software and hardware[5]. In this context, CloudCV[6] presented an example of a cloud-based and distributed computer vision platform that offer the access to state-of-the-art computer vision applications as a cloud service through a Web Interface. Image processing On Line (IPOL)[7] platform provides image processing and analysis algorithms, described accurately with source codes. This work allowed researchers to select and execute image processing algorithms using a web interface. Recently, we have developed a cloud-based platform for computer aided medical diagnosis related to scoliosis and osteoporosis diseases[8,9].

2.2 | Big Data frameworks for multimedia processing

The challenge of Big Data has become a very important task that needs to be managed within cloud resources. This is mainly due to the exponential grow of data in social networks and more particularly multimedia information. In this context, Yan et al.[10] proposed a cloud platform devoted to large-scale image processing based on Hadoop[4]. Authors analyzed the performance of the cloud architecture using a large set of image processing algorithms. As result, they reported some issues with data distribution and cluster resource related to the use of Hadoop.
On the other hand, the management and treatment of massive data requires an efficient exploitation of HPC (High Performance Computing) architectures. This task is not so easy and becomes more challenging when the HPC resources are hosted in the cloud. Recently, Netto et al.[11] presented in a survey the challenges of Cloud HPC for scientific, research and business applications. They noted different aspects that need to be considered for performance optimization when using HPC resources in the cloud. The main reported criteria are : the scheduling that affects the performance of jobs, platform selection, performance scalability, elasticity for dynamic resource adding and finally predictors for future resource consumption.

There are also some image processing works that exploit cloud architectures such as HIPI (Hadoop Image Processing Interface)[12], which is based on the Apache Hadoop MapReduce parallel programming framework. HIPI provides for users an efficient solution for storing large sets of images on Hadoop Distributed Files Systems (HDFS) for a further distributed treatment. The OpenCV library [1] is integrated within the HIPI interface for image treatment algorithms. Yamamoto and Kaneko[13] proposed and evaluated an approach for video processing in cloud and distributed environments using MapReduce on Hadoop.
Despite the efficiency and accuracy of the above-mentioned implementations, none of them can provide a cloud-based treatment of large-scale images using high performance computing resources. Our contribution can be summarized with three points:

1. Development of a complete solution that allows to exploit High Performance Computing (HPC) resources for the treatment of large-scale images (Big Data). Our solution applied parallel and distributed treatment of multimedia objects where parallel treatments are performed within GPU's computing units and distributed treatments are performed within distributed nodes (CPUs or GPUs).

2. Efficient deployment of our solution on the cloud where the resources are selected depending on the type and complexity of algorithms and managed data. This allowed to provide a Software as Service (SAAS) application with high performance (accelerated) and reduced costs (of resources location). Moreover, our cloud-based application can be easily migrated to other cloud platforms thanks to the use of Docker images (Dockerfile and docker-compose).

3. Secure management of users' sessions where data users are encrypted, saved and duplicated in order to keep a backup in case of crash. Each user can have access to the history of previous computations in order to compare results and adjust parameters.

In addition to the efficient applications deployment in the cloud, this paper presents the interest of our platform for improving performance as a result of efficient selection and management of cloud resources.

3 | CLOUD-BASED PLATFORM FOR LARGE-SCALE IMAGE PROCESSING

To ensure high performance of our cloud-based multimedia processing platform, we developed a tool that can be described within three subsections: cloud-based platform, parallel and distributed processing of images, web server management.

3.1 | Cloud-based platform

We propose a cloud framework hosting several multimedia processing applications. We used for this aim three virtual machines (VMs), selected accordingly to the complexity of the deployed applications. The role of each virtual machine is described below:

- **VM_01: server machine**: the first VM is used as a web server for communicating with the other virtual machines using SFTP and SSH protocols. Notice that SFTP is used to send data between machines, and SSH is used to execute the applications (deployed on the other VMs). In order to secure users information within our platform, we created in the server VM a Docker image, based on Hadoop, from which we launched three (03) containers representing one master node and two slave nodes. The master node (container) is used to generate, for each user, an encryption key using the RSA algorithm[14]. A backup of this encryption key is automatically copied on the two slave nodes using Hadoop. Indeed, if the master node fails, one of the slave masters will be designated as master. This process is maintained by

[1] OpenCV. https://opencv.org/

the framework Hadoop. Notice that VM_01 represents the master machine that stores all the applications data (if users accepts), where a backup is stored in the VM_02. On the other hand, a complexity estimation algorithm is integrated in this VM in order to affect computing resources efficiently so that highly intensive algorithms are executed on GPUs and low intensive algorithms are executed on CPUs. Our complexity estimation is based on five parameters: parallel fraction, computation per-pixel, computation per-image, dependency factor. For more details about this estimation, we refer authors to our previous publication [15]. Fig. 2 summarizes the role of the server VM.

- **VM_02: backup machine**: the second VM is used to save a backup of applications and data (encrypted or not) in order to keep the platform working either in case of crash. This VM represents the backup machine of the master node (VM_01).

- **VM_03: multimedia processing machine**: the third VM is used to run image and video processing applications that can use classic and deep neural approaches. In this case, we use the Docker framework that allows to deploy the applications without the need to install operating systems. Notice that Docker is an open source platform released in 2013 and used for the creation, deployment and management of applications. Docker is mainly based on images and containers where images allow to define the precise software packages (applications, libraries, configurations, etc.). Images can be also created by combining or modifying other standard images downloaded from public repositories. On the other hand, containers present instances of images that can be executed from each user (one user can execute one container). Docker containers are isolated and are run on single operating systems which makes them so lightweight than virtual machines. To summarize, Docker containers present an open source software platform of development. Its main advantage is the ability to package applications in containers, which allows them to be portable among any system running the Linux operating system (OS). In our case, we have generated two docker images for each integrated application (in the platform):

  - CPU-based docker image: including sequential (CPU) versions of the required algorithms and libraries such as, OpenCV, Tensorflow [2], Keras [3], etc.
  - GPU-based docker image: including parallel (GPU) versions of the required algorithms and libraries such as, OpenCV, Tensorflow, Keras, etc.

  Notice that the parallel versions are based on the API CUDA, where we included our own CUDA implementations for image and video processing (edge and corner detection) [15]. The GPU implementations used within Deep Learning and multimedia retrieval applications are imported from the GPU module of OpenCV and tensorflow libraries, which is also based on the API CUDA. The resource selection (CPUs or GPUs) is based on our complexity estimation algorithm, which is integrated in the VM_01. Moreover, the use of Docker enables multi-user exploitation of the integrated applications. Indeed, multiple users can execute the same cloud-based method simultaneously (Fig. 1). For example, two different tenants/users can be executed on the same VM or different VMs. Our platform can be easily extended for using more virtual machines by deploying the docker images. As shown in Fig. 1 , our platform can support many users. From one image docker, we can create several docker containers (one container for each user). So, if we have N users, the platform will generate N containers in order to execute applications. At the end of each container execution, the results will be saved before the destruction of the corresponding container. Notice that this virtual machine (VM_03) disposes of two processors (CPU and GPU) where the resource is selected accordingly to the algorithm complexity and the size of processed data. In case of saturation, other virtual machine can be reserved and managed with the same docker images.

The general architecture of the proposed cloud-based platform is summarized in Fig. 2 . This figure illustrates the platform with the three VMs. The heart of this platform is the virtual machine that runs the web server since this machine interacts with the client and the other VMs.

3.2 | Parallel and distributed storage and processing

Within our platform, we propose parallel and distributed management of multimedia objects for both storage and processing. In terms of data storage, all the data are encrypted using the protocol RSA [16]. In this context, Hadoop allows to replicate data with the HDFS system. RSA is a cryptosystem for public-key encryption, used for securing sensitive data, particularly when being sent over an insecure network such as the Internet. Our use of Hadoop is due to our need for tasks distribution between nodes and for managing backups in case of system failure. The choice of HDFS is also motivated by our interest for managing massive volumes in batch mode (multimedia indexation and deep learning data training), where HDFS is more suitable than Spark. These features makes from HDFS and efficient system for managing massive data.

[2]TensorFlow. https://www.tensorflow.org/learn
[3]Keras: The Python Deep Learning library. https://keras.io/
[3]https://www.docker.com/resources/what-container

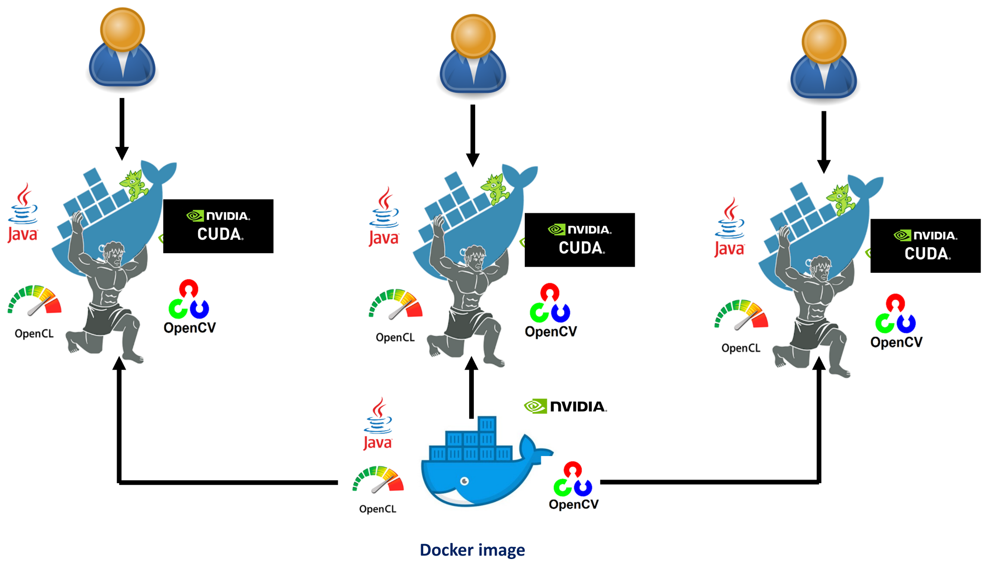


FIGURE 1 Multi-user exploitation of applications within Docker containers

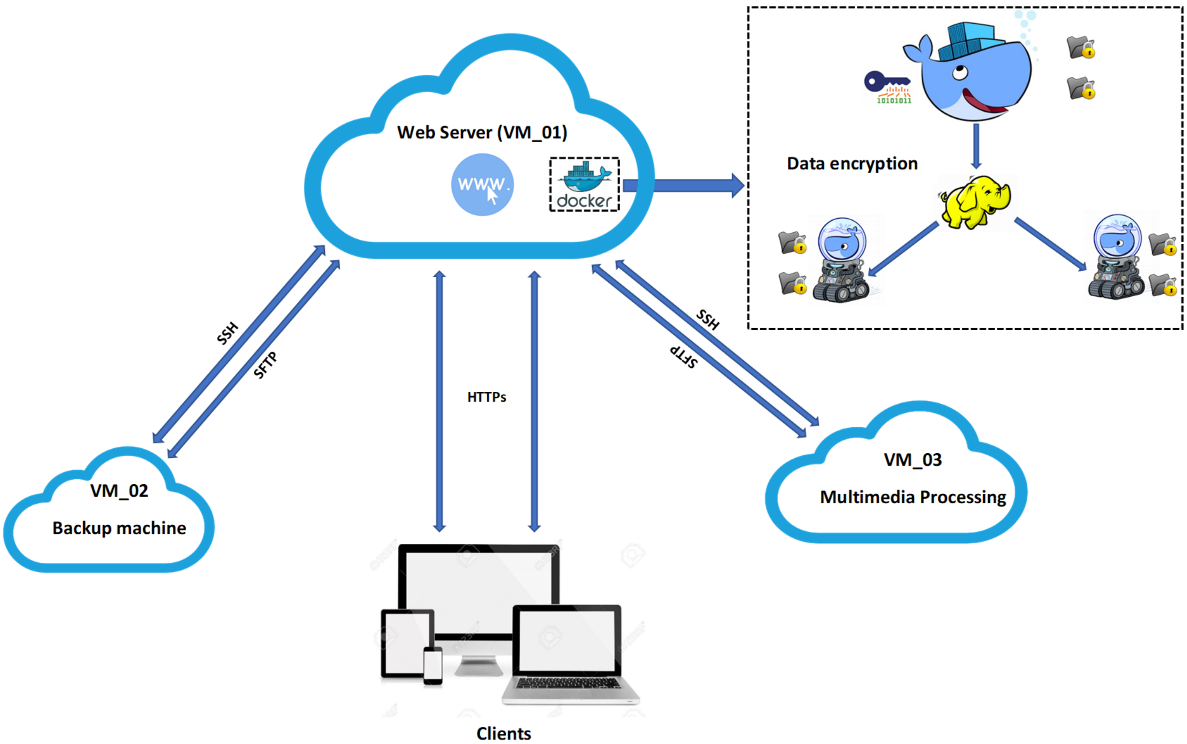


FIGURE 2 The general architecture of our platform

Thanks to this implementation, the data can be recovered in case of crash within the slave machine (VM_02). We note that the HDFS is a distributed file system, which offers high-performance access to data across Hadoop clusters. Like other Hadoop-related technologies, HDFS has become a key tool for managing massive volumes and supporting Big Data analytics applications. Notice that the operation of encryption is executed in the master machine (VM_01) when the user accepts to store his data in the platform.

In terms of processing, we used the framework HIPI [12] that represents an image processing library designed to work with the Apache Hadoop MapReduce parallel programming framework. HIPI offers efficient and high-throughput image processing with MapReduce style parallel programs typically executed on a cluster. It provides a solution for storing large collection of images on the Hadoop Distributed File System (HDFS) and makes them available for distributed processing. The choice of HDFS service is due to our need for tasks distribution between nodes and for managing backups in case of system failure. This choice is also motivated by our interest for managing massive volumes in batch mode (multimedia indexation and deep learning data training), where HDFS is more suitable than Spark. Moreover, HIPI integrates the OpenCV module, a popular open-source library that contains many computer vision algorithms [4]. In our case, we adapted the HIPI framework for applying deep learning techniques [17] in a distributed mode where the training process can be performed faster. The general presentation of a program that use MapReduce/HIPI frameworks is shown in Fig.3 .

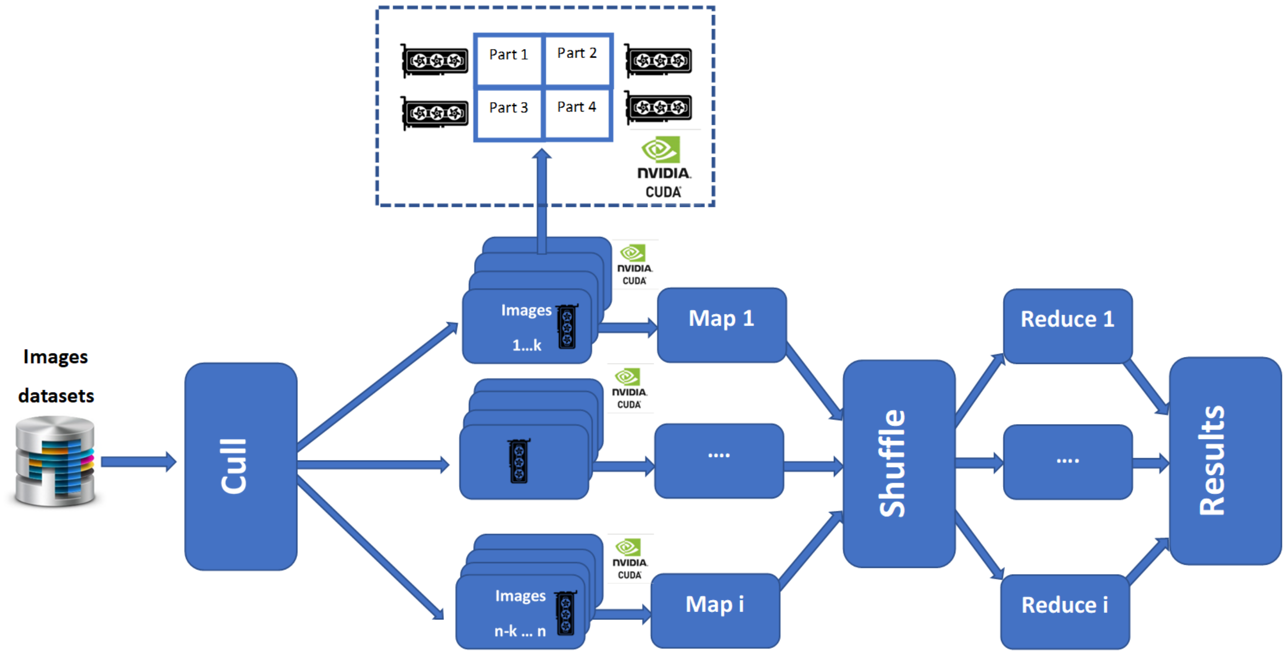


FIGURE 3 The organization of MapReduce/HIPI framework using multiple GPUs. Inspired from [12]

We use the framework HIPI to store images within distributed nodes in order to apply a distributed treatment per after. In addition to this distribution, the algorithms applied on each image (from the distributed nodes) are implemented in parallel [18 15] for exploiting the high computing power of GPUs. This process allowed to manage Big Data image volumes in a fast way thanks to the efficient exploitation of graphic cards. Notice that for each algorithm, we provide its CPU and GPU (using the API CUDA[5]) versions in order exploit the appropriate resource for treatment. Our platform integrates three kind of algorithms:

1. Classic image processing: such as image smoothing, image denoising, edge detection, points of interest extraction (Harris, SIFT, SURF), etc. The CPU and GPU implementations of these methods are provided from the library OpenCV, which was compiled with the CUDA module.
2. Motion tracking: mainly represented by optical flow estimation, which allows to extract motion vectors and directions. This algorithm was developed for exploiting CPUs, GPUs and either Multi-GPU platforms. For more details about this implementation, we refer authors to our previous publications [19 20].

[4] Open Source Computer Vision Library. OpenCV. https://opencv.org/
[5] CUDA Zone. https://developer.nvidia.com/cuda-zone

3. Deep learning algorithms: represented by the employment of classification and localization architectures that allow to classify images and localize objects within images. The CPU and GPU implementations of these methods are provided from the library TensorFlow, which was compiled with the CUDA module.

3.3 | Web server management

The website of our platform is developed with the Bootstrap framework that allows to have a responsive multi-platform website, which can be executed even on mobile devices (smartphone, tablet, etc.). To ensure high security, we use the framework Symfony [21] for users sessions management. We use the protocol HTTPS, which is based on the certificate of Let's encrypt[6] that represents a free, automated, and open certificate authority brought by the non-profit Internet Security Research Group (ISRG).

4 | EXPERIMENTAL RESULTS

Experimentation has been conducted using three VMs that dispose of the following hardware:

- VM_01 : 1 CPU Intel XEON E5 , 5.33GHz, 32 GB of RAM, 1 TB of storage
- VM_02 : 1 CPU Intel XEON E5 , 5.33GHz, 32 GB of RAM, 1 TB of storage
- VM_03 : 1 CPU Intel XEON E5 , 5.33GHz, 32 GB of RAM, 1 TB of storage, 4 NVIDIA GPU GeForce GTX 980 with 4 GB of RAM.

In order to test and validate the above-mentioned large-scale image processing cloud platform, we have integrated three kinds of applications and algorithms:

- Classic image and video processing toolbox ;
- Deep learning models for images classification and object localization ;
- Images retrieval and indexation method.

4.1 | Image and video processing toolbox

We started by integrating an image and video processing toolbox in our platform. This toolbox allows users and developers to test, exploit and combine several image processing algorithms such as image denoising, features extraction, SIFT [22] and SURF [23] [24] [25] descriptors, etc. Within our cloud platform, these algorithms can be turned in real time. The integrated algorithms are executed using a Docker image in order to offer to users the possibility to group the application parameters (input parameters, algorithm type and hardware type) in a container. With this, the container can execute the algorithm and send the result to the website.

On the one hand, the integrated algorithms are implemented in parallel using the CUDA API for exploiting GPUs. In case of processing multiple images, they are distributed between the different nodes in order to apply a distributed storage and processing within Hadoop framework. Thanks to this configuration, the change of material does not affect our configuration since we have just to install Docker and download the appropriate images. Moreover, this configuration allows us to have a multiple user access simultaneously. Notice that the integrated algorithms can exploit CPU or GPU depending on the users preferences and complexity estimation of algorithms. The image processing algorithms are also integrated and adapted for processing videos instead of images. Moreover, we have integrated some motion tracking algorithms such as background subtraction, face, people and cars detection and tracking in real time, etc.

4.2 | Deep learning models for images classification and object localization

Deep learning presents and important branch of machine learning and artificial intelligence. Deep learning architectures are based on neural networks and can be applied in different domains such as computer vision, natural language processing, speech recognition, temporal series,

[6] Let's encrypt. https://letsencrypt.org/

robotics, etc. Deep learning algorithms are generally represented by neural networks with multiple layers that allow to transform input annotated data into a representative model of data. In the domain of computer vision, input data (first layer) are presented by a matrix of pixels. The other layers allow to combine pixels values in order to detect specific features such as corners, edges, faces, etc. Notice that convolutional neural networks (CNNs) are particularly used in the domain of computer vision, where images features are calculated within the application of different convolutions. The process of Deep learning is based on two phases: 1) model preparation using a set of annotated data in order to define the most convenient model. 2) after defining the model, the test or inference consists of testing (prediction or classification) the model with new data that have never been presented to the neural network.

In this work, we have worked with deep neural architectures that are used for images classifications and objects localization [26]. Notice that training step is launched within the thrid VM at each increase or change of our annotated images databases, while the inference step is launched from users. The both steps are executed on GPU since they are known by their high intensity. For this aim, we tested the most used classification (VGG16, VGG19, Inception V3, Xception, ResNet, DenseNet, MobileNet) and localization (Yolo V2, Yolo V3 [27]) architectures, showing an accuracy going up to 99% and less that 10% for the loss. The classification architectures were tested using the GHIM database [7] (10000 images and 10 classes) and the localization architecture was tested with COCO database [8]. The high scores of accuracy are obtained as a result of exploiting multi-layer neural networks. We note that object localization architecture (using Yolo algorithm) was deployed using the same process . This algorithm allows to define the precise localization of detected objects.

Fig.4 illustrates some integrated algorithms that have been defined in Section 4.1 and 4.2. The figure shows that users can select the material: CPU or GPU. Notice that the use of GPU offers betters performance as a result of the parallel exploitation of GPU's computing units. The address of the toolbox platform is : www.multimedia-processing.fr. Users are invited to register, connect and test our toolbox algorithms.

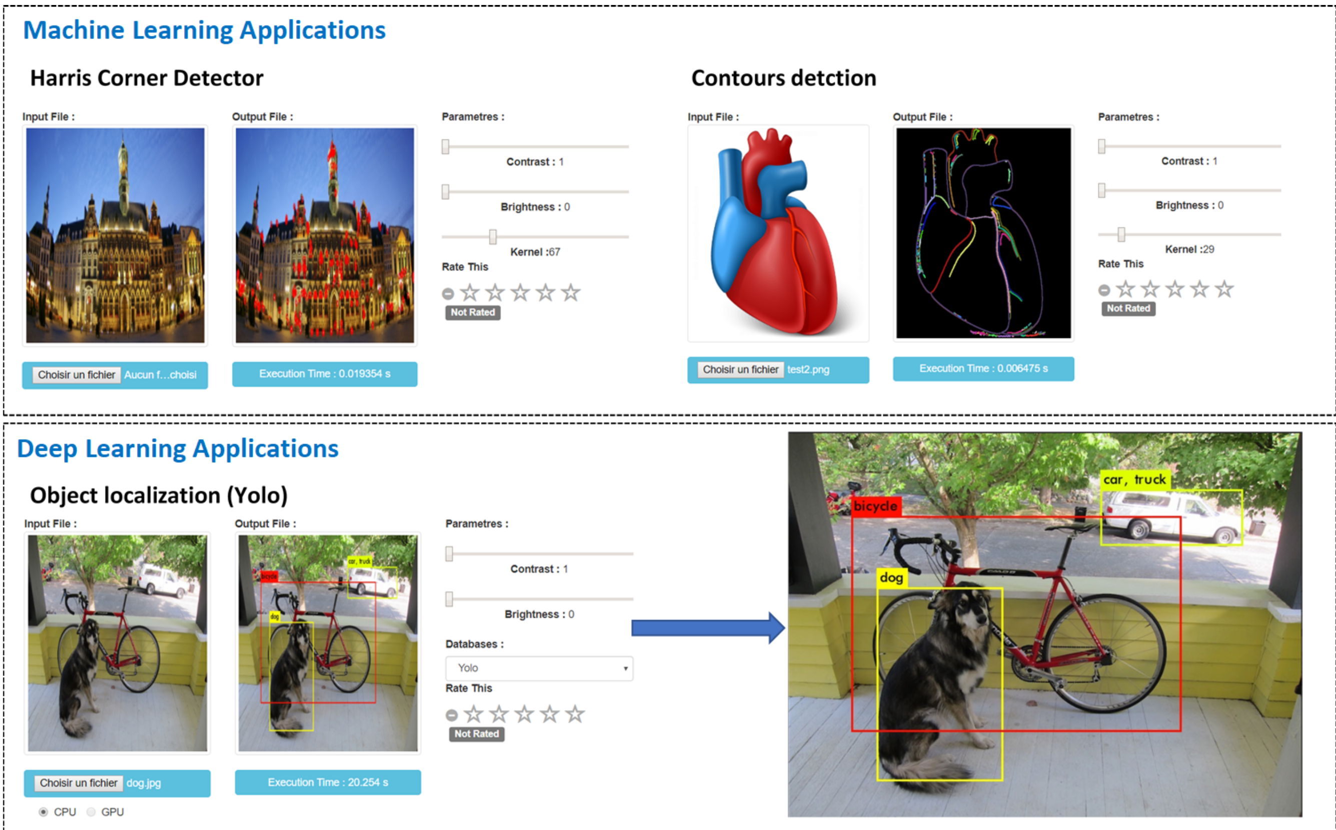


FIGURE 4 Illustration of our cloud-based image processing and Deep learning algorithms

[7] GHIM database. http://www.ci.gxnu.edu.cn/cbir/Dataset.aspx
[8] COCO databse. http://cocodataset.org

4.3 | Images indexation and retrieval

We have also integrated a method for indexing images within a system that allows users to perform a research in a set of images. We have used the Hadoop framework in order to replicate the data in the backup VM (VM_02), and also to obtain a fast research time by using the function Map and Reduce. These functions apply a parallel computing (on GPU) in order to accelerate the process of indexing and research. In fact, we ensure scalability when using several CPU and GPU resources, horizontally, thanks to the use of HIPI framework. Notice that our test platform is using one resource only. Our experiments where done using four GPUs. The general scheme of our content-based images retrieval method is shown in Fig. 5 .

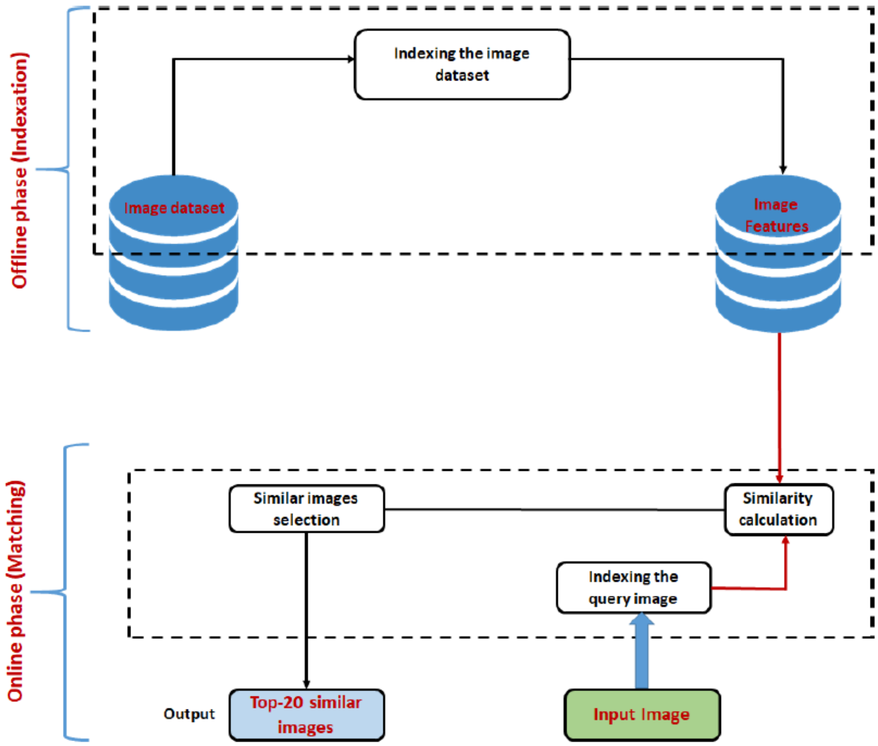


FIGURE 5 The general architecture of Content Based Image Retrieval (CBRIR)

The proposed approach of multimedia retrieval is based on four steps:

1. A feature extraction process is applied by using both classic descriptors (SIFT and SURF) and deep learning feature maps.
2. The application of PCA as dimensionality reduction method in order to reduce the dimension of these features.
3. The generation of the binary tree where the extracted features are replicated in each node
4. The implementation of the previous methods with the framework HIPI in order to have a distributed storage of features and parallel computing.

Notice that we have adapted the framework HIPI for exploiting GPUs in order to accelerate the process of feature extraction (SIFT and SURF descriptors) by the parallel exploitation of GPU's computing units. Indeed, the OpenCV library (used by HIPI) was compiled with the GPU module based on the API CUDA. We used both HIPI with CPU and HIPI with GPU (compiled within CUDA and OpenCV GPU). Thus, SIFT and SURF descriptors are executed on GPU using the API CUDA. The use of GPU's computing units in parallel allowed to accelerate the process of indexation and research with a speedup ranging from 1,3x to 2,6x compared to CPU version (HIPI using CPUs only).

Fig.6 , illustrates the comparison of features extraction time (of large-scale volumes) between the proposed parallel and distributed (using HIPI and GPUs) and the distributed solution, which is not using GPUs (using CPUs). The use of GPU with HIPI allowed to obtain speedups ranging from 1,3× 2,6×.

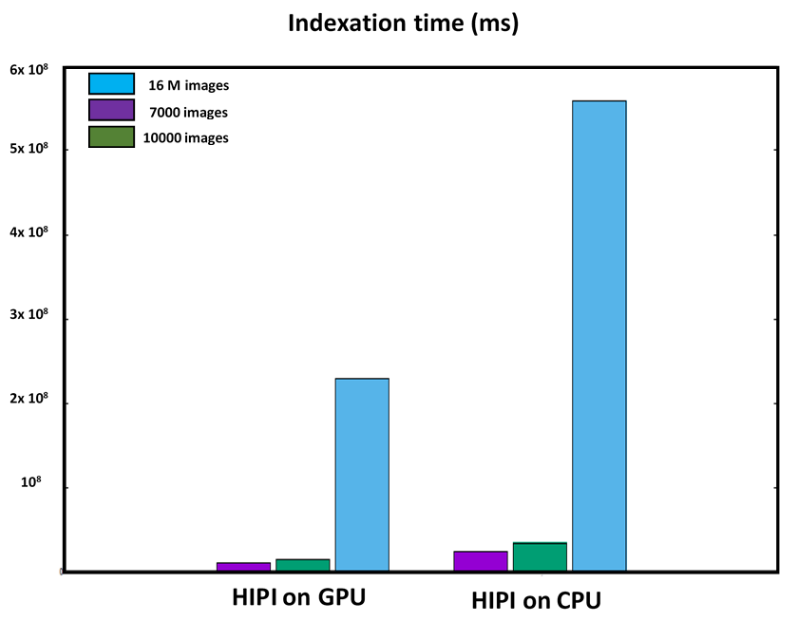


FIGURE 6 Performance comparison of Features extraction (HIPI vs HIPI + GPU)

On the other hand, Fig. 7 illustrates the comparison of research (retrieval) time (of large-scale volumes) between the proposed parallel and distributed (using HIPI and GPUs) solution and the distributed solution, which is not using GPUs. Here also, the use of GPU with HIPI allowed to obtain speedups ranging from 1,3× to 1,5×. Less accelerations are obtained for the research step since the GPU is not so suited for dependent computations such as those present in our case, where we need to compute the distance between different feature maps.

The address of the toolbox platform is: www.multimedia-processing.fr. Users are invited to register, connect and test our images retrieval algorithm. In terms of performance, Table 1 summarizes the interest of using our cloud platform for the different use cases (classic image processing, images classification and object localization using Deep Learning architectures, images indexation and retrieval). This table shows that the use of GPU allows to highly improve the performance either if the resource is on the cloud. Moreover, the platform is adapted for exploiting multiple GPUs if needed.

5 | CONCLUSIONS

In this paper, we proposed a real time cloud-based platform for computer vision applications. This platform integrates a toolbox of image and video processing, using Machine and Deep learning algorithms, that can be run in real time and process large scale databases. The related libraries and hardware drivers are automatically integrated and configured in order to offer to users an access to the different algorithms without the need to download, install and configure software or hardware. The platform was evaluated by three kinds of algorithms: 1. image and video processing toolbox. 2. Images classification and object localization using Deep Learning architectures. 3. Images indexation and retrieval. This evaluation demonstrated that our platform can deal with heterogeneous data (images, video, etc.) and is suitable for Big Data volumes. The platform allowed to improve algorithms performance thanks to the use of GPU's computing units. The cloud deployment of our applications is based on allocating resources (in the cloud) accordingly to the kind of application, processed data and users preferences. As results, our platform provided an easier access to different image and video processing applications (using Machine and Deep Learning algorithms), ensuring fast executions, low costs of cloud resources location and low energy consumption. Moreover, our cloud-based application can be easily migrated to other cloud platforms

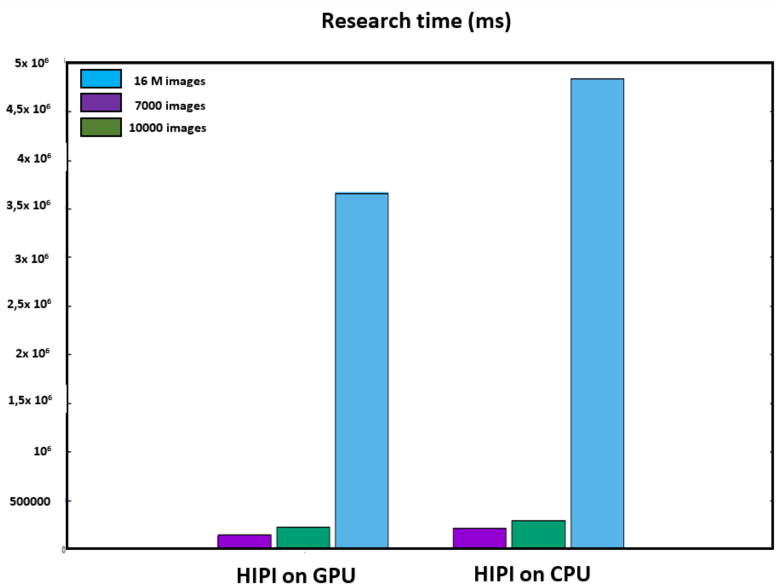


FIGURE 7 Performance comparison of retrieval process (HIPI vs HIPI + GPU)

thanks to the use of Dokcer images (Dockerfile and Docker-compose). We note that our framework can be so useful for researchers in the domain of multimedia processing using artificial intelligence algorithms. These applications are known by their high intensity of computations and also the complicated configuration since they use libraries (OpenCV, Tensorflow, caffe, keras) that could exploit CPUs or/and GPUs using massive data (Big Data).

As future work, we plan to improve our parallel and distributed approach by exploiting: 1) Multiple GPUs simultaneously for image processing algorithms and 2) Distributed deep learning techniques for exploiting multiples resources for data training. We plan also to improve the security of our cloud platform by providing a full isolation of docker containers such as described in [28 29 30].

ACKNOWLEDGMENTS

The authors would like to thank our colleague Mr Adriano Guttadauria for technical support and help for configuring and administrating the cloud architecture.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. *Linux Journal.* 2014;2014(239).

2. A Freier., P Karltonet.. The Secure Sockets Layer (SSL) Protocol Version 3.0.. *Internet Engineering Task Force (IETF).* 2011;.

3. Srivastav Akanksha, Bhartee Ajeet. Design of Secure File Transfer over Internet. *International Journal of Advanced Research in Computer and Communication Engineering.* 2016;5:471–473.

4. Shvachko K., Kuang H., Radia S., Chansler R.. The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST).* 2010;:1-10.

5. Calasanz R. B.-I.. Towards the scientific cloud workflow architecture. *5th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE'2017).* 2017;.

Category	Algorithm	Selected unit	Acceleration
Features extraction	Gaussian filtering	**GPU**	**112,4** ↗
	Edges detection	**GPU**	**18,03** ↗
	Corners detection	**GPU**	**23,55** ↗
	SIFT descriptor	**GPU**	**180,14** ↗
	SURF descriptor	**GPU**	**85,62** ↗
	Optical flow estimation	**GPU**	**7,41** ↗
Images classification	VGG16	**GPU**	**33,25** ↗
	MobileNet	**GPU**	**39,97** ↗
	Inception V3	**GPU**	**27,09** ↗
	ResNet50	**GPU**	**46,14** ↗
Multi-object localization	Yolo V2	**GPU**	**16,67** ↗
	Yolo V3	**GPU**	**12,83** ↗
Images classification	VGG16	**GPU**	**33,25** ↗
	MobileNet	**GPU**	**39,97** ↗
	Inception V3	**GPU**	**27,09** ↗
	ResNet50	**GPU**	**46,14** ↗
Images indexation & retrieval classic approach	Using SIFT and SURF descriptors	**GPU**	**2,63** ↗
Images indexation & retrieval Deep Learning approach	Using InceptionV3, DenseNet121 and MobileNet models	**GPU**	**48,88** ↗

TABLE 1 Performance comparison (CPU vs. GPU) of the integrated algorithms (test image resolution : 1920 $\times$ 1080)

6. Agrawal Harsh, Mathialagan Clint Solomon, Goyal Yash, et al. CloudCV: Large-Scale Distributed Computer Vision as a Cloud Service. *Mobile Cloud Visual Media Computing: From Interaction to Service.* 2015;:265–290.

7. Limare Nicolas, Morel Jean-Michel. The IPOL Initiative: Publishing and Testing Algorithms on Line for Reproducible Research in Image Processing. *Procedia Computer Science.* 2011;4:716 - 725. Proceedings of the International Conference on Computational Science, ICCS 2011.

8. Mahmoudi Sidi Ahmed, Adoui Mohammed El, Belarbi Mohammed Amin, Larhmam Mohammed Amine, Lecron Fabian. Cloud-based Platform for Computer Vision Applications. *Proceedings of the 2017 International Conference on Smart Digital Environment.* 2017;:195–200.

9. Mahmoudi Sidi Ahmed, Belarbi Mohammed Amin, Adoui Mohammed El, Larhmam Mohammed Amine, Lecron Fabian. Real Time Web-based Toolbox for Computer Vision. *Journal of Science and Technology of the Arts. Vol. 02(10).* 2018;.

10. Yan Yuzhong, Huang Lei. Large-Scale Image Processing Research Cloud. *The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization.* 2014;:88–93.

11. Netto Marco A. S., Calheiros Rodrigo N., Rodrigues Eduardo R., Cunha Renato L. F., Buyya Rajkumar. HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges. *ACM Comput. Surv.*. 2018;51(1):8:1–8:29.

12. Sweeney Chris, Liu Liu, Arietta Sean, Lawrence Jason. HIPI : A Hadoop Image Processing Interface for Image-based MapReduce Tasks. In: ; 2011.

13. Yamamoto Muneto, Kaneko Kunihiko. Parallel Image Database Processing With MapReduce and Performance Evaluation IN Pseudo Distributed Mode. *International Journal of Electronic Commerce Studies.* 2012;3.

14. Somani U., Lakhani K., Mundra M.. Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. *2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010).* 2010;:211-216.

15. Mahmoudi S. A., Manneback P.. Efficient exploitation of heterogeneous platforms for images features extraction. *2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA).* 2012;:91-96.

16. Kalpana Parsi, Singaraju Sudha. Data security in cloud computing using RSA algorithm. *International Journal of research in computer and communication technology, IJRCCT, ISSN*. 2012;:2278–5841.

17. Dean Jeffrey, Corrado Greg, Monga Rajat, et al. Large scale distributed deep networks. *Advances in neural information processing systems.* 2012;:1223-1231.

18. Mahmoudi Sidi Ahmed, Ozkan Erencan, Manneback Pierre, Tosun Suleyman, Jeannot E, Zilinskas J. Taking advantage of heterogeneous platforms in image and video processing. *High-Performance Computing on Complex Environments.* 2014;95:429.

19. Mahmoudi Sidi Ahmed, Kierzynka Michal, Manneback Pierre, Kurowski Krzysztof. Real-time motion tracking using optical flow on multiple GPUs. *Bulletin of the Polish Academy of Sciences: Technical Sciences.* 2014;62(1):139–150.

20. Larhmam Mohamed Amine, Mahmoudi Sidi Ahmed, Benjelloun Mohammed, Mahmoudi Saïd, Manneback Pierre. A Portable Multi-CPU/Multi-GPU Based Vertebra Localization in Sagittal MR Images. In: Campilho Aurélio, Kamel Mohamed, eds. *Image Analysis and Recognition*, :209–218Springer International Publishing; 2014; Cham.

21. Zaninotto François, Potencier Fabien. The definitive guide to Symfony. 2007;.

22. Lowe David G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision.* 2004;60(2):91–110.

23. Bay Herbert, Tuytelaars Tinne, Van Gool Luc. Surf: Speeded up robust features. *European conference on computer vision.* 2006;:404–417.

24. Belarbi Mohammed Amin, Mahmoudi Saïd, Belalem Ghalem. Indexing video by the content. *Information Systems Design and Intelligent Applications.* 2016;:21–33.

25. Belarbi Mohammed Amin, Mahmoudi Saïd, Belalem Ghalem. PCA as dimensionality reduction for large-scale image retrieval systems. *International Journal of Ambient Computing and Intelligence (IJACI).* 2017;8(4):45–58.

26. Durand Thibaut, Mordan Taylor, Thome Nicolas, Cord Matthieu. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017;:642–651.

27. Redmon Joseph, Farhadi Ali. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. 2018;.

28. Priedhorsky Reid, Randles Tim. Charliecloud: Unprivileged Containers for User-defined Software Stacks in HPC. In: SC '17:36:1–36:10ACM; 2017; New York, NY, USA.

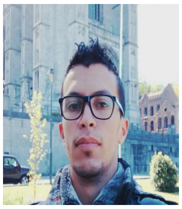
29. Azab A.. Enabling Docker Containers for High-Performance and Many-Task Computing. In: :279-285; 2017.

30. Arnautov Sergei, Trach Bohdan, Gregor Franz, al. . SCONE: Secure Linux Containers with Intel SGX. In: ; 2016.

31. Mahmoudi S.A., Belarbi M.A., Mahmoudi S., Belalem G.. Towards a smart selection of resources in the cloud for low-energy multimedia processing. *Concurrency Computation: Practice and Experience. 2018;30:e4372*. 2018;30(12):1–13.

AUTHOR BIOGRAPHY

Sidi Ahmed Mahmoudi. received the graduate engineering degree in computer science from the University of Tlemcen, Algeria, the master degree in multimedia processing from the Faculty of Engineering in Tours, France, and the PhD degree from the University of Mons, Belgium, in 2006, 2008, and 2013, respectively. Currently, Sidi Ahmed Mahmoudi is working as PhD research associate at the University of Mons, Belgium. His research interests are focused on the efficient exploitation of local and cloud computing resources (multi-CPU/ multi-GPU) architectures for multimedia processing and machine learning algorithms within Big Data volumes. Sidi Ahmed Mahmoudi has participated in several national (ARC-OLIMP, Numédiart, Slowdio, Comptoux, CLEO and FEDER IDEES) projects and European actions (COST IC 805 and COST NESUS IC1305). He is author or co-author in more than 50 international publications.



Mohammed Amin Belarbi. is a PhD Student from the Mathematics department, Faculty of Exact Sciences and Computer Science, University of Mostaganem, Algeria and from computer science department, Faculty of Engineering, University of Mons, He started his PhD the in University of Mostaganem, Algeria and University of Mons in 2014 (Co-Tutelle). He holds a master's degree in engineering information system in 2013 and Computer Science degree in 2011 from University of Mostaganem. His research interests are Multimedia Retrieval, Indexing videos and images.

Saïd Mahmoudi. graduated from the Computer Science Department, Faculty of Sciences, University of Oran, Algeria. He received his MS in Computer Science from the LIFL Laboratory, University of Lille1, France in 1999. He obtained his PhD in Computer Science at the University of Lille1, France in December 2003. Between 2003 and 2005, he was an Associate Lecturer at the University of Lille3, France. Since September 2005, he is Associate Professor at the Faculty of Engineering of the University of Mons, Belgium. His research interests include inernet of things, images processing, computer aided medical diagnosis, 2D and 3D retrieval and indexing and annotation.

Ghalem Belalem. graduated from University of Oran, Algeria, where he received PhD degree in computer science in 2007. His current research interests are distributed systems, grid and cloud computing, placement of replicas and consistency management in large scale systems and mobile environment.

Pierre Manneback. Pierre Manneback obtained his Ph. D. in Applied Mathematics in 1985 at the University of Namur, Belgium. He worked since 1990 at University of Mons, Faculty of Engineering, where he is Professor. His research is focused on parallel and distributed computing. He is member of several scientific committees of international meetings in distributed computing. He has published about 50 papers and communications to international conference in the field of scientific computation and parallel computing. He is particularly interested in deploying high demanding applications on new parallel architecture, either local (multicore, GPU) or distant (Clouds). He is member of the Board of CETIC (Applied Research Center in ICT, Charleroi, Belgium). He teaches courses in Parallel and Distributed Computing, Methodology of Programming, Real Time Computing and Web Technologies.

How to cite this article: S. A. Mahmoudi, M. A. Belarbi, S. Mahmoudi, G. Belalem, and P. Manneback (2019), Multimedia processing, High performance computing, Cloud computing, Big Data, Deep Learning, *Concurrency and Computation: Practice and Experience*, *2019;00:1–15*.