

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Optimization of an Earth Observation Data Processing and Distribution System

Jonathan Becedas, María del Mar Núñez and
David González

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.71423>

Abstract

Conventional Earth Observation Payload Data Ground Segments (PDGS) continuously receive variable requests for data processing and distribution. However, their architecture was conceived to be on the premises of satellite operators and, for instance, has intrinsic limitations to offer variable services. In the current chapter, we introduce cloud computing technology to be considered as an alternative to offer variable services. For that purpose, a cloud infrastructure based on OpenNebula and the PDGS used in the Deimos-2 mission was adapted with the objective of optimizing it using the ENTICE open source middleware. Preliminary results with a realistic satellite recording scenario are presented.

Keywords: Earth Observation, distributed systems, cloud computing, ENTICE project, gs4EO

1. Introduction

Traditionally, Earth Observation systems have been operated by governments and public organizations; the primary investors being US, China, Russia, Japan and Europe mainly because of worldwide common objectives such as climate change, sustainable development and objectives at national level.

However, from 2015 to 2016, the Earth Observation from space paradigm is changing with the globalization of the market, the evolution of the information and communication technologies and the high investment of private entities in the field.

This boost of commercial interest in Earth Observation can be explained because of the parallel evolution of three main pillars, as stated by Denis et al. in [1]:

1. Increased performance of commercial satellites with defence needs in the range of very high resolution products, i.e. resolutions between 0.25 and 1 m.
2. The development of hybrid procurement schemes between private and public customers.
3. Appearance of the New Space scheme started in Silicon Valley, which attracted the interest of investors and contributed to the creation and entrance of new actors in the space sector.

To these, we would add the dedicated budget of new countries, such as Kazakhstan, Venezuela and Vietnam, in EO; increased budget in new EO programmes for India, China and South Korea [2] and fast evolution of information and communication technologies, which facilitated the creation of new applications requiring availability of lots of information in the shortest time possible. This contributed to the evolution of the space sector in two manners: (a) the evolution of the sensors to provide highest performance at a lower cost and (b) the launch of more satellites to cover the demand of information. This last explains the increase in the launch of satellites during the last years and interest of satellite operators to operate satellite constellations in order to reduce the revisit time and offer more coverage of the land surface. A proof of this is the number of EO satellites launched between 2006 and 2015: 163 satellites over 50 kg were launched for civil and commercial applications, generating \$18.4 billion in manufacturing market revenues, whereas 419 satellites are expected to be launched over the next decade (2016–2025), generating \$35.5 billion in manufacturing revenues. In terms of EO data sales, the market reached \$1.7 billion in 2015 and it is expected to reach \$3 billion in 2025. This is \$12.2 billion total revenue in the decade 2006–2015 and \$24 billion in the decade 2016–2025 [3]. The amount of generated data is used, for instance, to accumulate spatial and temporal records of the world itself, of the events and changes that occur in it in a diverse number of applications: security, maritime, agriculture, energy and emergency, among others [4].

However, the infrastructures used to manage EO data are still based on traditional EO systems, which (because of their previous ambit of application) make use of on-site traditional infrastructures or data centers. Their architecture was designed to be monolithic in a localized single infrastructure.

Now, the process of recording data from Earth observations generates massive amounts of spatiotemporal geospatial information that has to be intensively processed for a variable and increasing demand. This is a handicap for traditional data centers since they are not designed to manage variable amounts of data. They were designed and sized to operate a certain data volume. They are then limited in terms of flexibility and scalability [5]. The storage of increasing amounts of data over time is also a challenge, since the recordings are also maintained by their owners over time as well [6].

Traditional Earth Observation Payload Data Ground Segments (PDGS) present the following limitations to cover the demands of the current EO market:

- i. Traditional infrastructures are not flexible or easily scalable to operate.
- ii. There is a risk of oversizing/undersizing the infrastructure to offer services when highly variable demand exists.
- iii. They make the cost of acquiring recent images of the Earth very high.

- iv. The customers cannot access directly neither fast to the information they need because this has to be processed and ad-hoc distributed.

However, the use of cloud computing technology can eliminate the previous drawbacks to improve EO services because it is elastic, scalable, it works on demand through virtualization of resources, offers virtually unlimited storage and computation capability, it is worldwide connected and it is based on a pay per use model [7, 8].

Nevertheless, the current cloud computing technology still presents some limitations:

- i. The virtual machine images (VMIs) are not optimized, being highly oversized, impacting in the costs of using the infrastructure and in the dynamic resources provisioning.
- ii. The deployment of virtual machines (VM) in cloud is not in real time. The deployment normally takes between 10 and 20 minutes, which directly affects to the flexibility and dynamic scalability of the system.
- iii. Although the pay per use model should intrinsically have reduced costs, since the customer only pays for what he uses, the costs of using cloud computing are still high.
- iv. There are some major worldwide champions in the offer of cloud services such as Amazon, Google, Microsoft and IBM, which make difficult the migration of a system from a cloud infrastructure to another different cloud infrastructure, existing vendor lock-in. This limits the democratization of these services and makes an entrance barrier for new cloud providers.

Within the ENTICE H2020 project (project no. 644179), we intend to demonstrate that processing the data recorded from Earth observations in a cloud environment with the middleware ENTICE optimizes the efficiency and overcomes the critical barriers of cloud computing and data processing needs. Among other advantages, ENTICE provides independence from a specific infrastructure provider and facilitates the distribution of VMs in distributed infrastructures.

In this work, we present the implementation of the Earth Observation Data (EOD) pilot, which mainly consists of the implementation in cloud of the already commercial Ground Segment for Earth Observation (gs4EO) suit, commercialized by Deimos [9], which is currently operational in the Deimos-2 satellite mission [10].

For this purpose, we simulate a real scenario with the Deimos-2 satellite running in a federated cloud infrastructure, in which we obtain real performance metrics and present real system requirements for normal operations with the satellite. Through this experimentation, we demonstrate the EOD concept as a solution for the new EO market paradigm.

2. Earth Observation Data Processing and Distribution Pilot

2.1. ENTICE environment

In order to facilitate the implementation in cloud, the EOD pilot makes use of the ENTICE middleware [11], which facilitates autoscaling and flexibility to the ingestion of satellite imagery, its

processing and distribution to end users with variable demands. Kecskemeti et al. [12] introduced the ENTICE approach to solve these problems. The ENTICE environment consists of a ubiquitous repository-based technology, which provides optimised virtual machine (VM) image creation, assembly, migration and storage for federated clouds. The webpage of ENTICE can be found in [13].

ENTICE facilitates the implementation of cloud applications by simplifying the creation of lightweight virtual machine images (VMIs) by means of functional descriptors. These functional descriptors define at high and functional levels the VMIs and contribute to define the system Service Level Agreement (SLA) to facilitate the optimization of the VMIs in terms of performance, costs, size and quality of service (QoS) needed. Then, the VMIs are automatically decomposed and distributed to meet the application runtime requirements. In addition, ENTICE facilitates elastic autoscaling. The benefits of using ENTICE are the following:

- Reduction of up to 80% storage.
- 95% elastic Quality of Service.
- VMIs creation 25% faster.
- Reduction on the costs of deployment.
- VMIs optimization up to 60%.
- VMIs delivery 30% faster.
- Scalability and elasticity.
- Elimination of cloud infrastructure vendor lock-in.

In the EOD pilot, ENTICE is used as middleware between the federated infrastructure described in Section 3.1 and the gs4EO application software.

2.2. EOD pilot description

The Earth Observation Data Processing and Distribution Pilot (EOD) consists of the implementation of the Elecnor Deimos' geo-data processing, storage and distribution platform of Deimos-2 satellite using cloud technologies. The main functionalities of the system are the following:

- **Acquisition of raw data:** When the imagery data are ingested from the satellite into the ground station, the system is notified and the ingestion component automatically ingests the raw data into the cloud for its processing.
- **Processing of data:** Once the data are ingested, it is processed in the product processors. There are several processing levels to provide different products.
- **Archiving and cataloguing geo-images:** The different products obtained from the processing of raw data are archived and catalogued in order to provide these images or high added value services to end users.
- **Offering user services:** This is the front-end of the system. It allows end users to select the product that they want to visualize or to download.

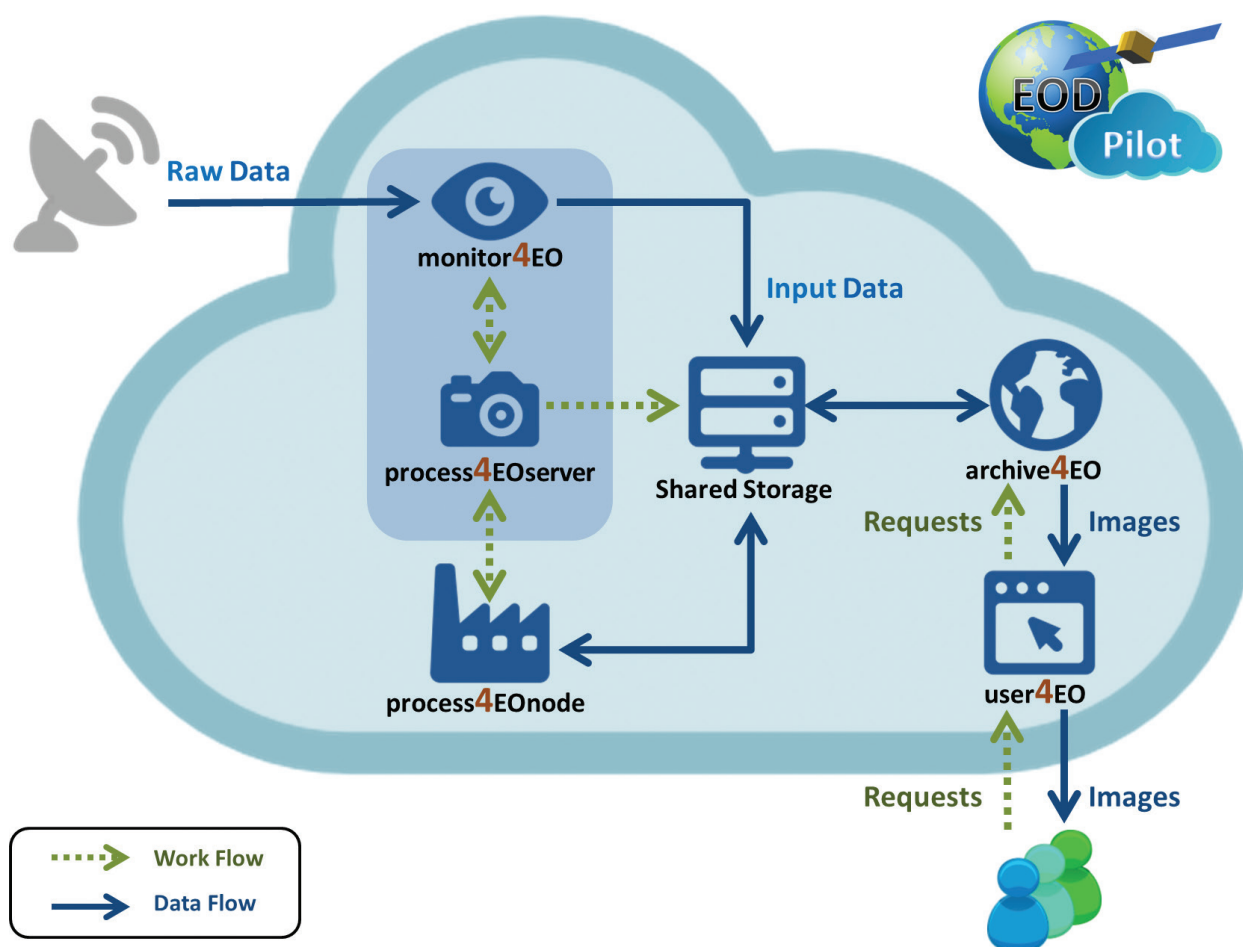


Figure 1. Earth Observation Data Processing and Distribution pilot (EOD)'s architecture.

2.2.1. EOD architecture

The main objectives of the EOD pilot is to process real data of Deimos-2 satellite in a realistic scenario of normal operation and the validation of the processing chain module as part of the cloud infrastructure. Ramos and Becedas [14] proposed an original architecture of the gs4EO suit to be implemented in cloud. Based on that work, the architecture for the EOD pilot has been redesigned and implemented, see **Figure 1**.

The architecture is composed of the following components:

- **monitor4EO:** It is a ground station monitor, which ingests the available raw data from the ground stations to the cloud system. It contains an Orchestrator, which manages the tasks of the different modules.
- **process4EO server:** It is the Orchestrator, which is the component that manages the tasks to be done by all the modules of the architecture computed in the cloud infrastructure. The Orchestrator has the following functions:
 - To identify which outputs shall be generated by the processors.
 - To generate the Job Orders. They contain all the necessary information that the processors need. Furthermore, these eXtensive Markup Language (XML) files include the interfaces

and addresses of the folders in which the input information to the processors is located and the folders in which the outputs of the processors have to be sent. They also include the format in which the processors generate their output.

- To find data in the ground stations (pooling) to be ingested in a shared storage unit in the cloud for its distribution to the processing chain.
- To control the processing chain by communicating with the product processors.
- To manage the archive and catalogue.
- **process4EO node:** Constituted of different software modules, which are in charge of the processing of the raw data and the products of previous levels to produce image products. **Figure 2** depicts the pipeline of the image processing process. The four most important operations are the following:
 - Calibration: (L0 and L0R processing levels) to convert the pixel elements from instrument digital counts into radiance units.
 - Geometric correction: (L1A processing level) to eliminate distortions due to misalignments of the sensors in the focal plane geometry.
 - Geolocation: (L1BR processing level) to compute the geodetic coordinates of the input pixels.
 - Orthorectification: (L1C processing level) to produce orthophotos with vertical projection, free of distortions.

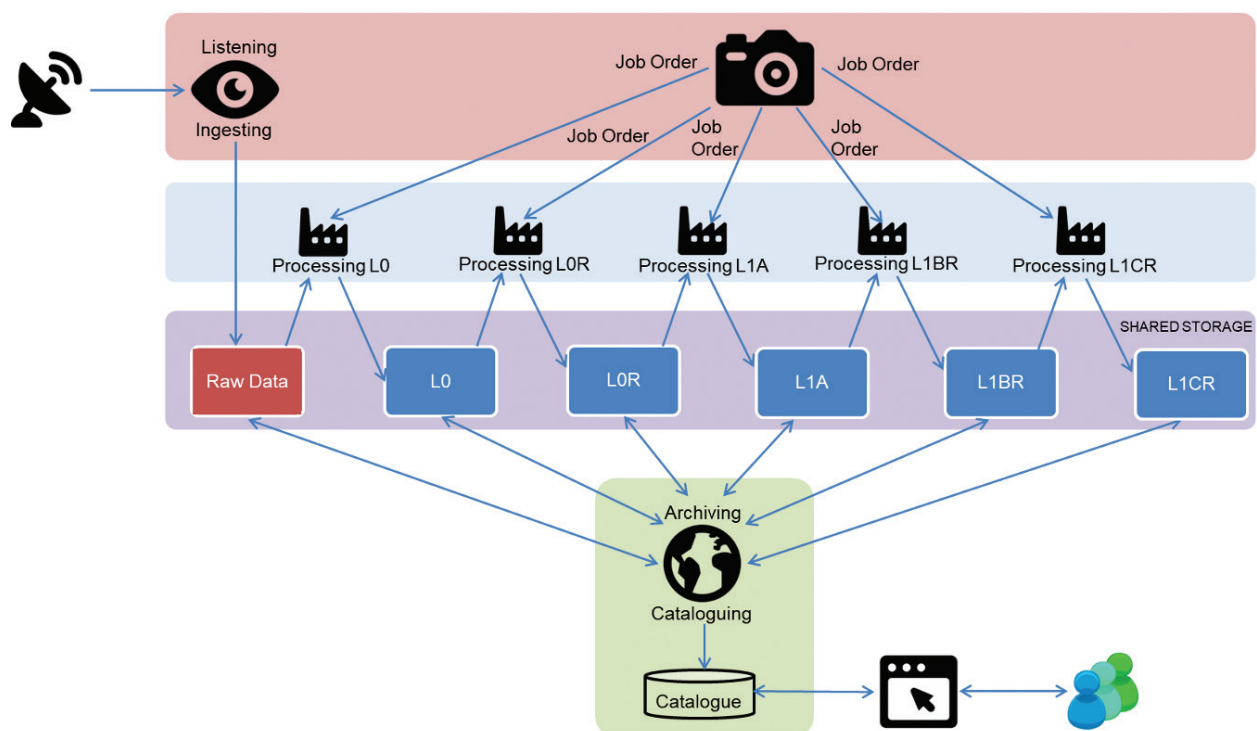


Figure 2. EOD's pipeline.

- **archive4EO:** In this module, the processed images are stored and catalogued for their distribution. It offers a Catalogue Service for the Web (CSW) interface.
- **user4EO:** It is a web service in which the end users can access to the products.
- **Shared storage:** It is a storage module shared by all the modules of the architecture in which all the inputs and outputs of the different modules of the architecture are stored.

3. Experiment setup

3.1. Testing infrastructure

The testing infrastructure used in the experiment is formed by hardware deployed in three different locations and managed in a federated manner: DMU infrastructure (in Deimos UK in United Kingdom), DMS infrastructure (in Deimos Space in Spain) and DME infrastructure (in Deimos Engenharia in Portugal). The hardware resources deployed in every location are described in **Table 1**. The ENTICE middleware was installed in the DMU infrastructure, which is acting as master. It also contains an object store with interface to Amazon Simple Storage Service (Amazon S3) for cloud bursting. DMS and DME infrastructures are slaves of DMU infrastructure and contain object stores also with interfaces to Amazon S3. A block diagram describing the interrelations of the testing infrastructure is depicted in **Figure 3**. The virtualization of the infrastructure was done with OpenNebula. Kernel-based Virtual Machine (KVM) was used as hypervisor. The creation of the virtual machines was done with Packer, whereas the automatic deployment of the virtual machines was done with Ansible. **Figure 4** shows a diagram describing the logic process of automatic generation of the virtual machines that constitute the EOD software. The image building process takes advantage of

| Location | Name | Model | CPU | RAM (GB) | HD (GB) | OS |
|----------|---------------|------------------|-------------------------------------|----------|---------|--------------------|
| DMU | Node-1 | Dell Optiplex790 | Intel Core i7-2600 3.4 GHz | 8 | 160 | CentOS 7.2.1511 |
| | Node-2 | Dell Optiplex790 | Intel Core i7-2600 3.4 GHz | 16 | 250 | CentOS 7.2.1511 |
| | OpenNebula-fe | Dell Optiplex745 | Intel Core 2 6300 1.86 GHz | 4 | 250 | CentOS 7.2.1511 |
| DMS | Node-2 | Dell | Intel 8 Core 2.37 GHz | 16 | 2048 | CentOS 7.2.1511 |
| | Node1 | Dell | Intel 2 Core 3 GHz | 6 | 230 | CentOS 7.2.1511 |
| DME | Node1 | HP | AMD Athlon 64 X2 Dual Core 3800+ | 4 | 256 | CentOS 7.2.1511 |

Table 1. Hardware resources in the testing infrastructure.

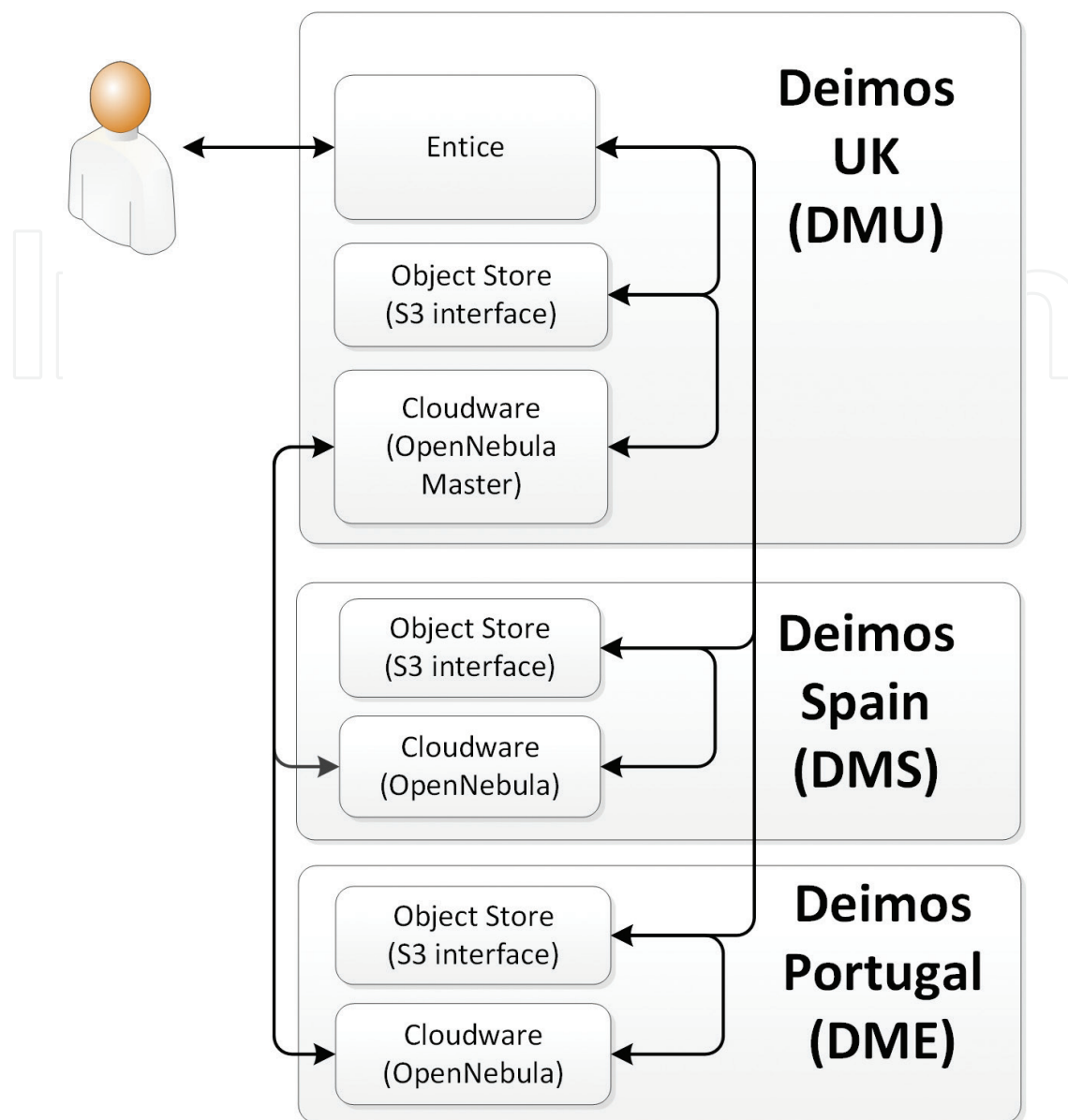


Figure 3. Block diagram of the testing infrastructure.

the functionalities provided by Packer and Ansible to build KVM images. The virtual images are based on CentOS 6 Linux distribution and are stored in qcow2 format. This automation step comprises several files:

- **Execution script:** This script, developed in Python, launches the creation of the machine image with Packer. It receives a JSON file with all the variables that will be used in the building process, e.g. the user configuration, software repositories, Kickstart file and Ansible playbook, and configures all the required fields in the Kickstart file. It can build all the types of VMIs required to deploy the EOD software: archive4EO, monitor4EO and process4EO. The type of virtual machine to generate is specified in the content of the configuration file.

- Packer template: It is a JSON file that provides all the information to create the virtual machine in Packer. It contains the format, the instructions and the parameters on how to build a VMI using KVM. The provisioners define the scripts or recipes in Ansible for configuring the machine and installing the applications.
- Ansible playbook: These files are “recipes” to install the EOD software in the virtual machines. This is a YAML file with the commands expressed in a simplified language, describing a configuration or a process. It contains the information to configure the system, install the EOD software and the functionalities to work in the cloud environment (contextualization).

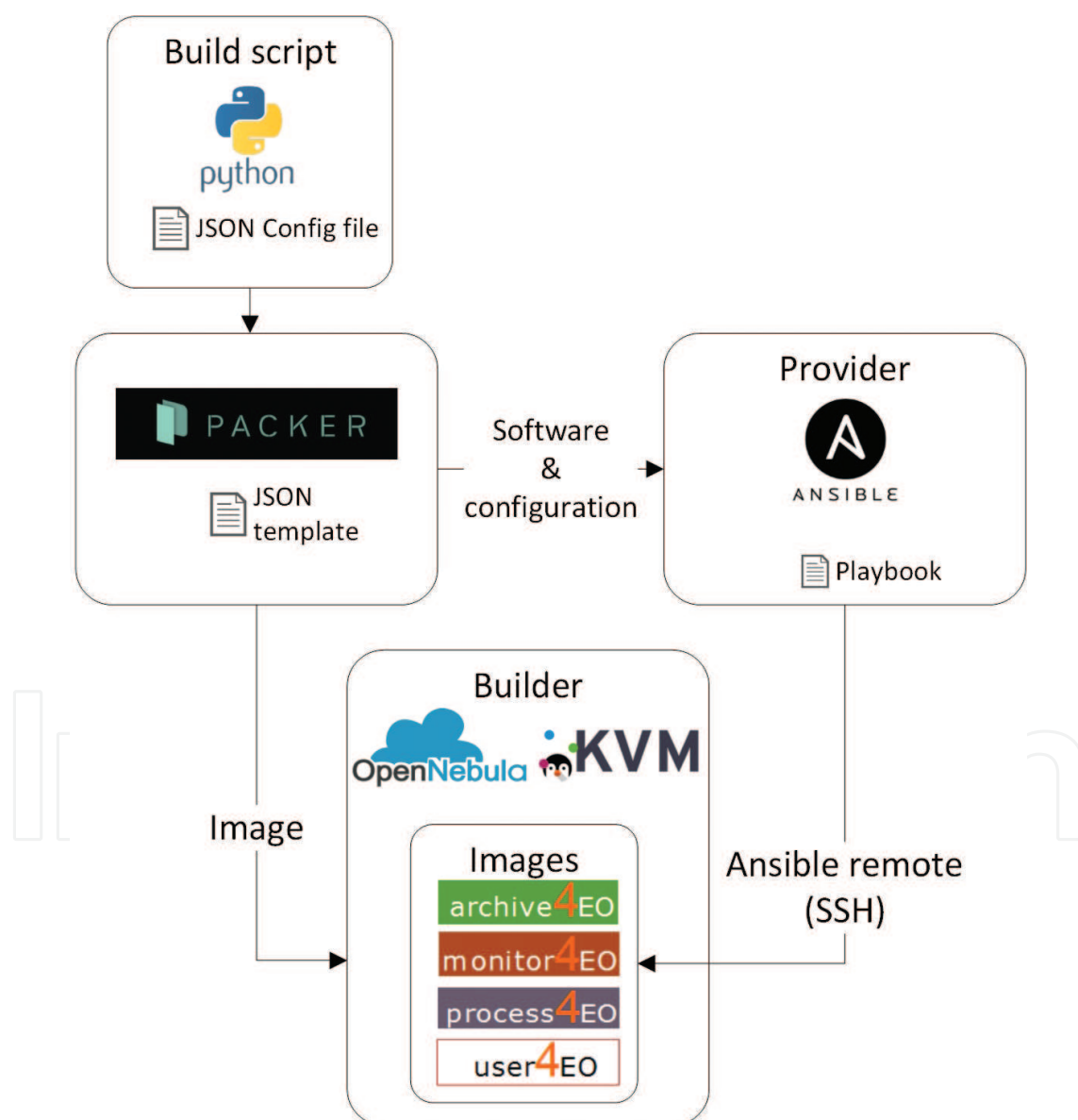


Figure 4. Diagram of the automatic generation of the EOD virtual machines.

The Python script receives the configuration file and launches the Packer command after configuring some parameters in the Kickstart file. The Packer command takes the template and runs all the builds within it in order to generate a set of artefacts and build the image in KVM. Once the image is built, Packer launches all the provisioners (Ansible) contained in the template. Ansible carries out several steps: it configures all the repositories, installs all the dependencies and software packages of the EOD modules, configures the EOD software and installs a context package to deploy the VMI in OpenNebula.

The recording of the experiment data was done with Jmeter™ [15] and Nagios® [16]. Jmeter™ is installed in the Node and Nagios® in a virtual machine inside the federated cloud. It is used for the monitoring of the cloud resources and status and to extract the experimental data.

3.2. Experiment description

The aim of this experiment is to demonstrate the feasibility of implementing the EOD system in cloud and how its behavior improves after the optimization done by ENTICE over the process4EO node.

The experiment is that of a realistic recording with Deimos-2 satellite in which a real acquisition is ingested into the EOD pilot. Then, the processing of the raw data is carried out with the EOD pilot before and after the optimization process. The results are compared to evaluate the functionality of the optimized system with regard to the nonoptimized system and validate the implementation of the gs4EO modules in cloud.

VMI size, VMI creation time, VMI delivery time and VMI deployment time are the evaluated metrics selected to compare the performance of the system before and after the optimization process.

The following are the evaluated metrics to demonstrate that the functionality of the system remains the same after the optimization: processing time, imagery products size, CPU use per process and memory use per process.

The raw data used in the experiment have 3 MB size, four multispectral bands (R, G, B and NIR) and one panchromatic. The recorded area of the land surface is a rectangle of $8.86 \times 16.59 \text{ km}^2$.

The raw data are managed and processed to automatically obtain the following products:

- L0: raw data decoded.
- L0R: transformation of L0 into image.
- L1A: geolocated and radiometric calibrated image.
- L1BR: resampled image and more precise geolocation.
- L1CR: orthorectification.

The virtual resources used in the experiment were the following: a virtual machine with 300 GB, a RAM of 10 GB, four CPUs of 32 bits, a shared storage with 99 GB and an additional storage volume with 50 GB. This hardware was used for both experiments (EOD before and after optimization) in order to facilitate comparison.

4. Experiment results

First, the virtual machine images of the EOD pilot were created, delivered and deployed in the cloud. Then, the virtual machine of the proces4EO was optimized and its VMI was again created, delivered and deployed. The time spent in every step is depicted in **Table 2**.

In these results, one can see the increase in the performance of the system before the runtime, i.e. up to the deployment of the system: this is a reduction of 30% in VMI size, a reduction of 37.3% in the VMI creation time, a reduction of 34.53% in the VMI delivery time and a reduction of 54.05% in the deployment time.

Next, the raw data recorded with the satellite were ingested in both the original EOD pilot and the optimized EOD pilot. The response of both optimized and nonoptimized systems were measured in the runtime. The processing time of the satellite imagery in the original EOD pilot and the EOD pilot with the optimization of the processing chain is shown in **Figures 5 and 6** respectively. It can be noticed that the processing time of the different levels is similar in both experiments, so as to the time to process the raw data up to the orthorectification level (L1CR): 33.95 and 35.75 s in the nonoptimized and optimized systems, respectively. This difference is not substantial and can be produced by some OpenNebula processes, or the cloud has used

| | VMI size (GB) | VMI creation time (hh:mm:ss) | VMI delivery time (hh:mm:ss) | VMI deployment time (hh:mm:ss) |
|-----------------|---------------|------------------------------|------------------------------|--------------------------------|
| Nonoptimized VM | 2 | 00:19:42 | 00:20:25 | 0:06:47 |
| Optimized VM | 1.4 | 00:12:21 | 00:13:22 | 0:03:07 |
| Reduction (%) | 30 | 37.31 | 34.53 | 54.05 |

Table 2. Metrics of the optimized and nonoptimized EOD pilot.

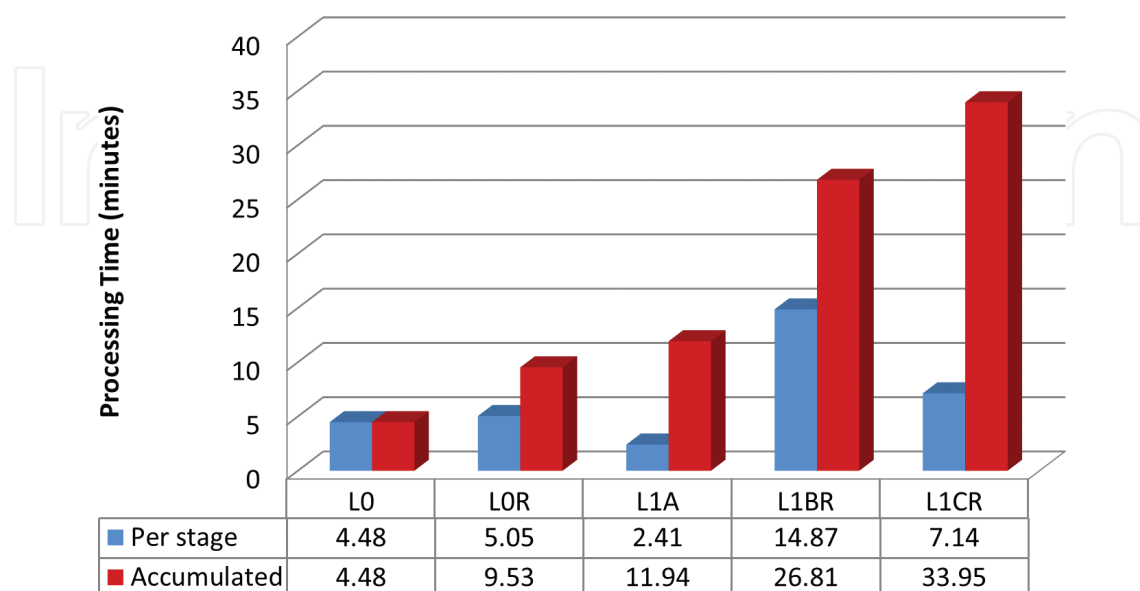


Figure 5. Processing time of the satellite imagery with nonoptimized EOD system.

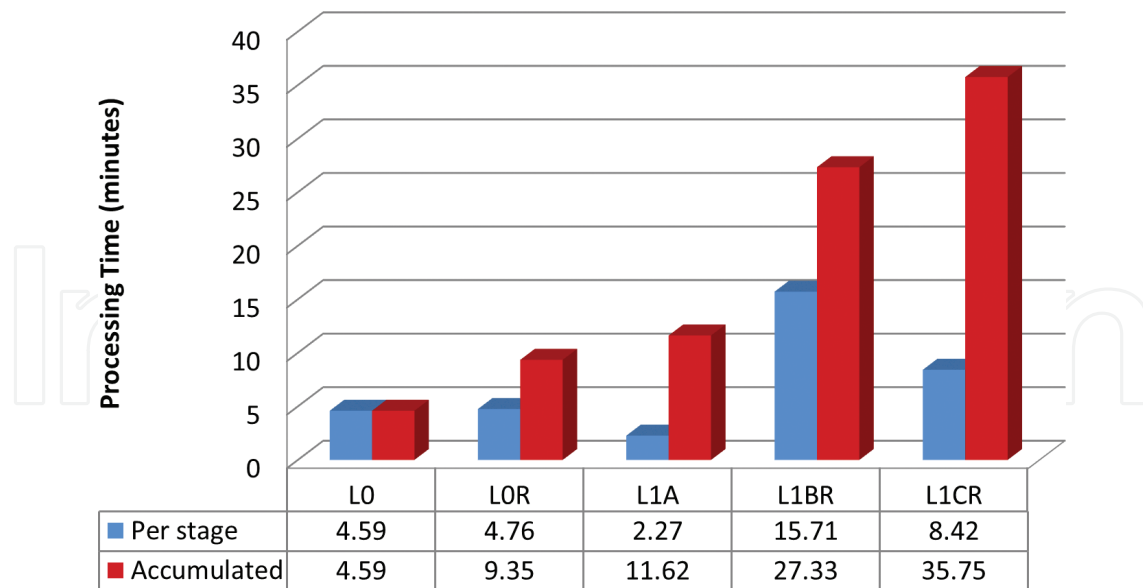


Figure 6. Processing time of the satellite imagery with optimized EOD system.

| Data type | Raw data | L0 | L0R | L1A | L1BR | L1CR | Total Products |
|--|----------|-----|-----|-----|------|------|----------------|
| Size of the products obtained with the non-optimized system (MB) | 3090 | 764 | 789 | 749 | 1140 | 1130 | 4572 |
| Size of the products obtained with the optimized system (MB) | 3090 | 764 | 789 | 749 | 1140 | 1130 | 4572 |

Table 3. Imagery product sizes obtained with both the nonoptimized and the optimized EOD system.

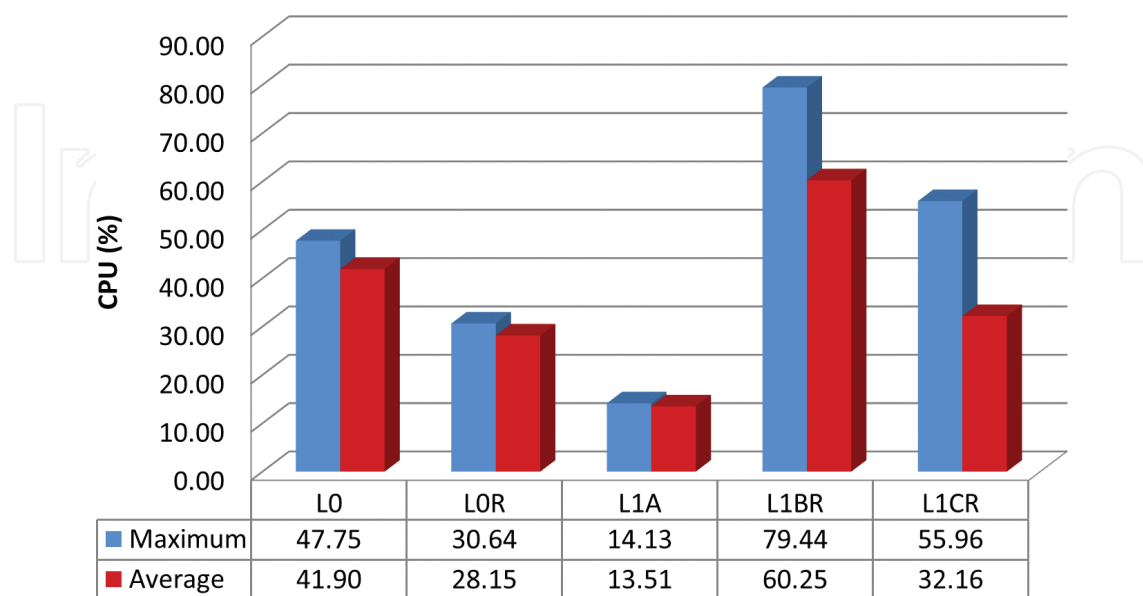


Figure 7. CPU use per process in the nonoptimized EOD system.

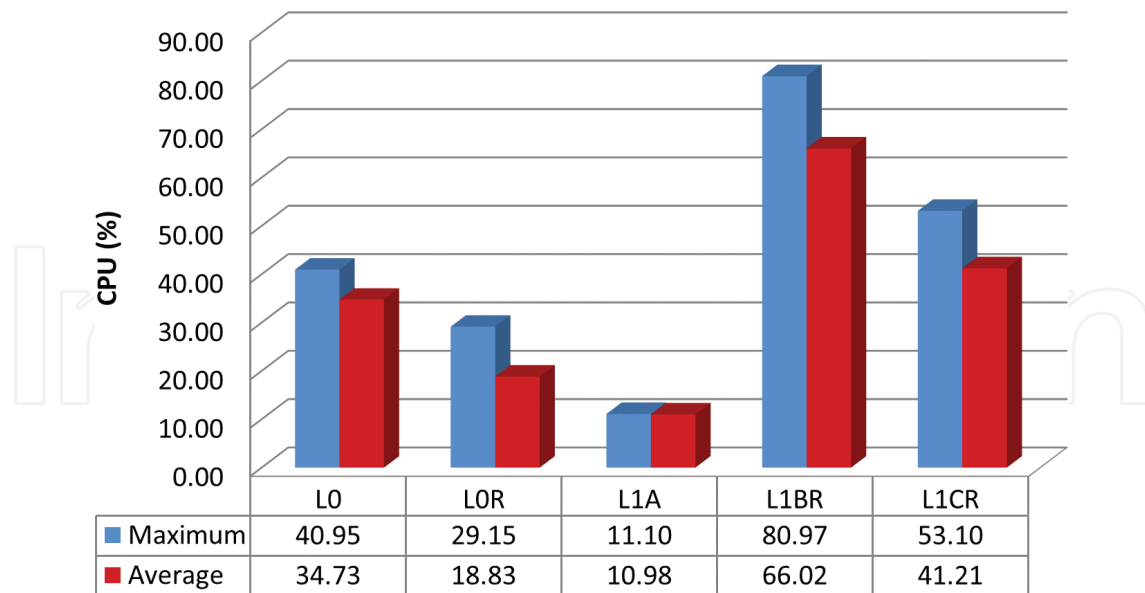


Figure 8. CPU use per process in the optimized EOD system.

some resources while executing the experiments. In addition, the size of the different imagery products in both experiments is depicted in **Table 3**. Notice that the size of the different products remains the same in both experiments. These demonstrate that the functionality of the system is intact after the optimization process, while the optimization provides benefits in storage, creation, delivery and deployment of the system.

Furthermore, the CPU and memory used in both experiments are similar for all the processing stages: in **Figure 7**, the CPU used in the processing of the satellite imagery with the non-optimized system is shown; in **Figure 8**, the CPU used in the optimized system is depicted.

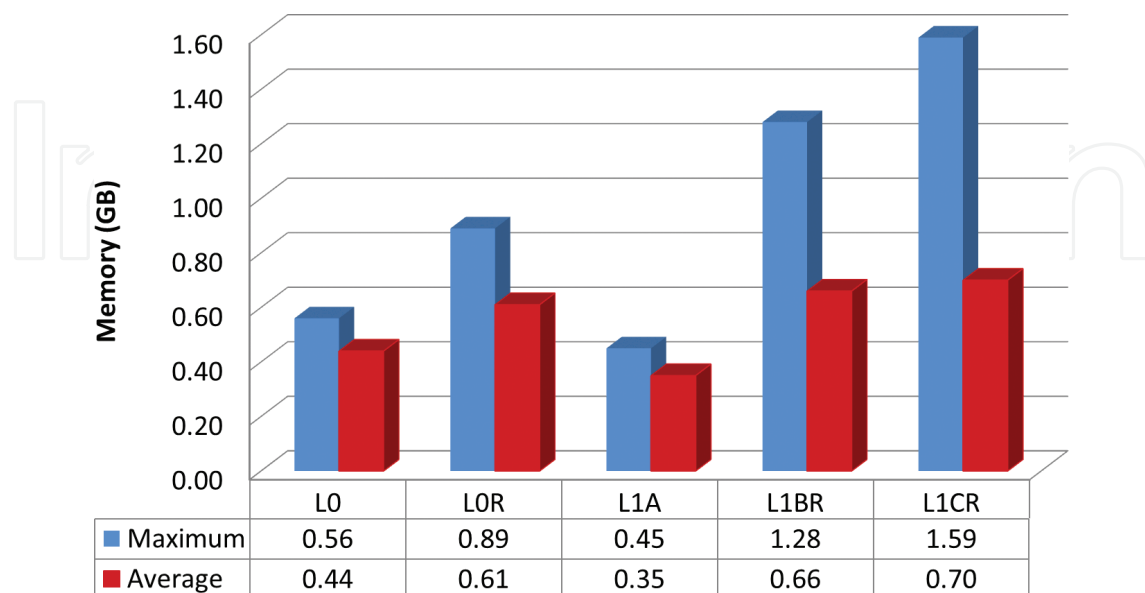


Figure 9. Memory use per process in the nonoptimized EOD system.

Besides, the memory used by the optimized system was lower: the memory use per process in the nonoptimized system can be seen in **Figure 9**, while the memory used in the optimized system can be seen in **Figure 10**.

These results obtained with the EOD pilot can be related with the new paradigms of the Earth Observation market stated in [1]. **Table 4** describes how an approach of a PDGS system similar to the EOD pilot could cover the main requirements of the new EO market.

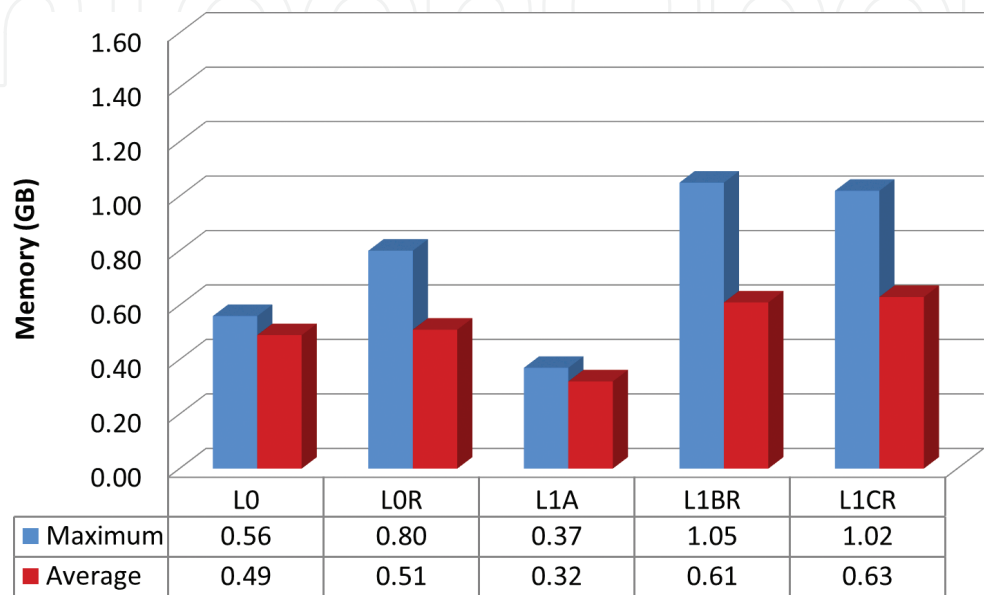


Figure 10. Memory use per process in the optimized EOD system.

| New paradigm requirements | EOD pilot |
|---|--|
| Costs optimization | Cost reduction by means of reduced storage of optimized VMIs, reduced creation time, reduced delivery time and reduced deployment time |
| Multi sensors ground processing systems | Ground stations, ground control centers and data processing centers would take advantage of a rapid, agile, resilient and secure interconnected computer system in cloud |
| Vertical integration | Global distributed infrastructure connecting all the stakeholders in an operational environment |
| Scalability | Elastically autoscale applications on cloud resources based on their fluctuating load with optimized VM interoperability across cloud infrastructures and without provider lock-in |

Table 4. New paradigm requirements vs. EOD pilot approach.

5. Conclusions and future work

In this work, the successful implementation of the EOD pilot in an experimental cloud infrastructure with the middleware ENTICE was demonstrated. The pilot was tested and

promising results were obtained. These results indicated that real scenarios of satellite imagery managing and processing can be carried out in cloud with many advantages with respect to traditional infrastructures. Furthermore, an optimization of the EOD pilot was carried out, demonstrating a reduction of 30% in VMI size, 37.3% in the VMI creation time, 34.53% in the VMI delivery time and 54.05% in the deployment time, while maintaining the functionality of the system intact. This indicates that a PDGS system implemented in cloud in a similar manner to that of the EOD pilot can fulfill the requirements of the new Earth observation market paradigm. Specifically, these EOD pilot results demonstrate that the deployment of an optimized PDGS system in cloud can reduce the costs of storage and reduce the time to user by reducing the creation time, the delivery time and the deployment time of the system. Besides, ground stations can take the advantage of rapid, agile, resilient and secure interconnected system when are cloud-based. In addition, the global operational environment provided by a cloud infrastructure facilitates both global acquisition and distribution of data, improving the market efficiency. Finally, the system improves its scalability without vendor lock-in, covering the needs of recent on demand markets.

In future research, different realistic scenarios with variable demand of services will be tested. With these scenarios, we will evaluate the elastic behaviour in the ingestion of raw data in the system, the processing and the distribution of imagery products to users. Furthermore, a complete optimization of the system will be tested to evaluate the complete repository storage size reduction, which was not evaluated in this work. In addition, new metrics will be measured to validate the implementation of the system for its commercial implementation in the next future.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 644179.

Author details

Jonathan Becedas*, María del Mar Núñez and David González

*Address all correspondence to: jonathan.becedas@deimos-space.com

Elecnor Deimos Satellite Systems, Puertollano, Spain

References

- [1] Denis G, Claverie A, Pasxo X, Darnis JP, de Maupeou B, Lafaye M, Morel E. Towards disruptions in Earth observation? New Earth Observation systems and markets evolution: Possible scenarios and impacts. *Acta Astronautica*, 2017;137:415-433. DOI: 10.1016/j.actaastro.2017.04.034

- [2] Keith A. Emerging programs, markets drives earth observation growth. *Earth Imaging Journal* [Internet]. 21 February, 2015. Available from: <http://eijournal.com/print/articles/emerging-programs-markets-drive-earth-observation-growth> [Accessed: 2017-08-24]
- [3] Keith A, Larrea Brito N, Manjunath S. *Satellite-Based Earth Observation: Market Prospects to 2025*. 9th ed. Euroconsult, Paris, France; September 2016
- [4] Samsó L, Quirce J, Becedas J. Distributed Satellite Systems: The Next Big Thing is Already Here. In: *Small Satellite Systems and Services Symposium (4S'2014)*; 26-30-05-2014, Majorca, Spain: ESA; 2014. p. 1-14
- [5] Farres J. Cloud Computing and Content Delivery Network Use within Earth Observation Ground Segments: Experiences and Lessons Learnt [Internet]. 2012. Available from: https://earth.esa.int/documents/1656065/1664726/11-Cloud_computing_content_delivery.pdf [Accessed: 2017-08-30]
- [6] Yang C, Goodchild M, Huang Q, Nebert D, Raskin R, Xu Y, Bambacus M, Fay D. Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth*. 2011;4(4):305-329. DOI: 10.1080/17538947.2011.587547
- [7] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. *Communications of the ACM*. 2010;53(4):50-58. DOI: 10.1145/1721654.1721672
- [8] Botta A, Donato W, Persico V, Pescapé A. Integration of cloud computing and internet of things: A survey. *Future Generation Computer Systems*. 2016;56:684-700. DOI: 10.1016/j.future.2015.09.021
- [9] gs4EO. Complete Suite of Ground Segment Products [Internet]. 2017. Available from: <http://www.elecnor-deimos.com/wp-content/uploads/2017/07/g4EO-EN-1.pdf> [Accessed: 2017-08-30]
- [10] Deimos-2 Minisatellite Mission [Internet]. 2017. Available from: <https://directory.eoportal.org/web/eoportal/satellite-missions/d/deimos-2> [Accessed: 2017-08-30]
- [11] Prodan R, Fahringer T, Kmovski D, Kecskemeti G, Csaba A, Stankovski V, Becedas J, Ramos JJ, Sheridan C, Whigham D, Rodrigo C. Use cases towards a decentralized repository for transparent and efficient virtual machine operations. In: *25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP'2017)*; 6-08-03-2017; St. Petersburg, Russia. IEEE; 2017. p. 478-485
- [12] Kecskemeti G, Csaba A, Kertesz A. The ENTICE approach to decompose monolithic services into microservices. In: *International Conference on High Performance Computing & Simulation (HPCS'2016)*; 18-22-07-2016; Innsbruck, Austria. IEEE, 2016. p. 591-596
- [13] ENTICE – dEcentralized repositories for traNsparent and efficienT vIRtual maCHine opERations [Internet]. 2017. Available from: <http://www.cloudwatchhub.eu/entice-decentralized-repositories-transparent-and-efficient-virtual-machine-operations> [Accessed: 2017-08-30]

- [14] Ramos JJ, Becedas J. Deimos' gs4EO over ENTICE: A cost-effective cloud-based solution to deploy and operate flexible big EO data systems with optimized performance. In: Proceedings of the 2016 Conference on Big Data from Space (BiDS'16); Santa Cruz de Tenerife, Spain. ESA; 2016. p. 107-110
- [15] Apache Jmeter™ [Internet]. Available from: <http://jmeter.apache.org/> [Accessed: 2017-09-01]
- [16] Nagios. The Industry Standard in IT Infrastructure and monitoring [Internet]. Available from: <https://www.nagios.com/> [Accessed: 2017-09-01]

