Payload Data Ground Segment in Cloud for Earth Observation Satellites with the Entice Middleware

Jonathan Becedas, María del Mar Núñez and David Gonzalez

Department of Satellite Systems
Elecnor Deimos
Puertollano, Ciudad Real 13500, Spain
{jonathan.becedas & maria-del-mar.nunez & david.gonzalez}@deimos-space.com

***Abstract* – In this work we implemented in cloud the Payload Data Ground Segment (PDGS) of the gs4EO commercial software, created by Deimos Elecnor Group. This pilot implementation is named EOD Pilot (Earth Observation Data Pilot). The advantages of the EOD Pilot with regards to a traditional PDGS are the following: optimized ingestion and processing of raw data incoming from ground stations, on-demand adaptation of the infrastructure for processing incoming data, provisioning of an optimized storage service for cataloguing, reduction of the delivery time of products to end users, data rate optimization between the Virtual Machines (VMs), VM startup and auto-scaling optimization. The implementation was done by using the ENTICE middleware, maintaining all the functionalities of the system and improving efficiency, agility and extensibility. The final results of system are shown.**

Index Terms –gs4EO, EOD Pilot, ENTICE, Cloud Computing, Earth Observation.

I. Introduction

During the last years, private organizations are gaining more prominence in the space domain. The new advances and increased performance of commercial satellites enhance the benefits of space investment in areas affecting our lifestyles; some examples are global positioning, navigation, timing, communication systems, global financial transactions, scientific research, environmental monitoring and forecasting, observing natural disasters, aviation, ground transportation and civil security systems.

Earth Observation (EO) satellites directly impact in society. The EO industry provides the sustained global, regional and local data to develop solutions to local, regional and global problems, addressing problems at a global scale. EO is used in the definition of global goals and frameworks, and adopted as the global standard to inform policy makers and to reliably report and monitor [1].

Therefore, Earth Observation missions are not only being exploited by public organizations, the industry is becoming a relevant actor nowadays. This irruption of private investments in EO increased the competition, reduced the costs of the missions and contributed [to develop more technology. Besides the synergies between private and public administrations in this field created programmes such as the European Copernicus Programme, which made EO more accessible worldwide improving the life of the citizens.

Nevertheless, in spite of the evolution of the spatial EO market, the infrastructures commonly used to manage EO data recorded with satellites, also known as Payload Data Ground Segments (PDGS), make use of traditional technology. Curent systems were designed to be monolithic and localized in a specific location. Common PDGS technology presents the following limitations:

a) Lack of flexiblility and scalability to operate in function of the demand of services.
b) Risk of oversizing or undersizing the infrastructure to offer services when highly variable demand exists.
c) High cost of acquiring, processing, manage and distribute recent images of the Earth.
d) Lack of direct and fast access to the information generated.

Cloud computing technology can be a solution to solve the previous problems because the virtualization of the infrastructure can provide flexibility and scalability by adjusting the resources to the demand, reduce the costs of creating an infrastructure and provides global access [2]. Nevertheless, cloud computing is far from being the perfect solution:

a) The use of public clouds is more expensive than expected.
b) The scalability cannot be done in fast and the times to provision and deploy resources are high.
c) The use of virtual machines requires more resources than actually needed.
d) Applications usually have to be developed for the specific public cloud to be used, existing difficulties to change from one cloud provider to a different one, also known as vendor lock-in.
e) Security is one of the major issues which reduce the growth of cloud computing due to its open environment with very limited user side control [3] [4].

In this work, we made use of the ENTICE middleware [5] [6], to implement the PDGS used to manage the data of the Deimos-2 satellite in cloud with the objective of reducing the drawbacks presented by current cloud computing technology. This pilot implementation was named as the EOD Pilot (Earth Observation Data Pilot).

With this implementation we demonstrated that a commercial PDGS can be implemented in cloud to respond to flexible demands; achieve increased utilization efficiency by centralizing storage, memory, processing and bandwidth at lower operational cost; guarantee Quality of Service (QoS); enable real scalability; shorten the time to user by parallelizing computing resources, and use the required resources in an optimum manner. The final results of the EOD Pilot are presented.

II. Earth Observation Data Processing and Distribution Pilot

A. *Payload Data Ground Segment in the EOD pilot*

The main functionality of the EOD pilot was to process real data (images) of the Deimos-2 satellite. The processing of the data was done in five different stages from the raw data acquired:

- 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 EOD pilot was constituted of the following components:

- monitor4EO: it ingested the raw data in the system and managed the system.
- process4EO: it processed the satellite data.
- archive4EO: in this module, the processed images were stored and catalogued for their distribution.
- user4EO: it was a web service providing access to the generated products.
- Shared Storage: it was a storage module shared by all the modules of the architecture.

The monitor4EO, the process4EO, the archive4EO and the user4EO were implemented in four different virtual machines. The architecture of the EOD pilot is depicted in Fig. 1.

More information about the functionalities of the different software modules can be found in [6] and [7].

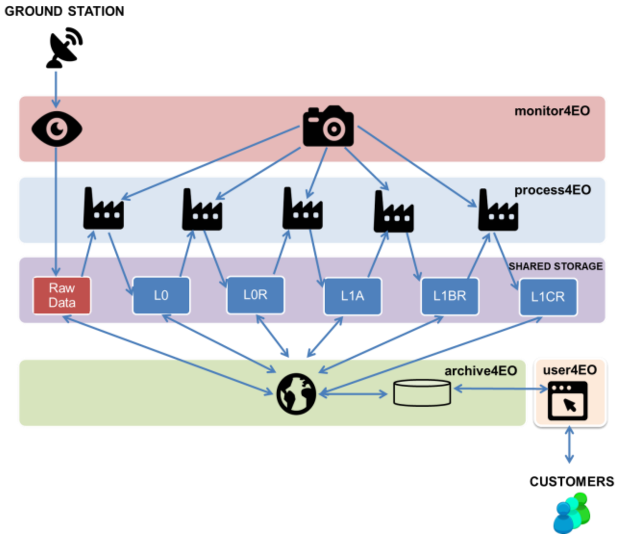

Fig. 1 Size of the VMIs - Not optimized vs Optimized

B. *ENTICE middleware*

In the EOD Pilot, ENTICE was used as a middleware between the federated cloud infrastructure and the software described above.

ENTICE provided optimization mechanisms through the use of fragmentation and replication of images, and a Pareto Multi-Objective Optimization solver. ENTICE has the following functionalities:

1) Preparation of lightweight VMI through functional descriptions: 60% smaller compared to regular user-created VMIs, reduce VM delivery time of VMI by 30%, down to around 10 seconds.

2) Distributed lightweight VMI storage: Reduce time for cross data center deployment by over 20%.

3) Autonomous multi-objective repository optimization: Preserve performance while decreasing costs and storage requirements by at least 25%.

4) Elastic resource provisioning: Improve the QoS elasticity.

The ENTICE technology was completely decoupled from the EOD application and its runtime environment, but supported it through optimized VM image creation, assembly, migration and storage [8]. ENTICE stored metadata about the virtual machine images (VMIs) and their fragments in a distributed knowledge base to be used for interoperability, integration, reasoning and optimization purposes. It decomposed the monolithic services into microservices to distribute system.

Therefore, ENTICE was composed of four services: Virtual Machine Optimizer (VMO), Knowledge Base (KB), Distributed Storage (DS) and Graphical User Interface (GUI). Each service was running in its own process. Consequently, the EOD pilot services were independently deployable and scalable.

ENTICE takes advantage of automation in order to create the VMIs of the application. The platform implemented the automation functionality. The EOD pilot machines were built using Packer and Ansible templates. Packer created the image of the virtual machines and Ansible installed the EOD software in the virtual machines. The process of VMI creation and software installation is described in [7] in detail.

III. Experiment Setup

A. *Testing Infrastructure*

The ENTICE federated cloud was used to optimize the virtual images of the EOD Pilot. This infrastructure was virtualized by making use of OpenNebula 5.2. Three nodes provided the computing capability. The number of cores per node differed depending on the version of the processor, the average was 48 cores. Each node had at least 128GB of RAM. The underlying virtualization technique was KVM (hardware virtualisation), and the storage was provided by both a Ceph distributed storage cluster and a QCOW2 store. Additionally, an Amazon S3 based storage was part of the infrastructure (via RadosGW). The resources used are detailed in Table I.

TABLE I

Physical Resources Used In The ENTICE Environment

Physical resources			
Number of nodes	3		
Nodes	Processor	Cores	Memory GB
	AMD Opteron™ Processor 6262 HE @ 1.6 GHz	64	256
	AMD Opteron™ Processor 6164 HE @ 1.7 GHz	48	128
	AMD Opteron™ Processor 6376 HE @ 2.3 GHz	32	128
Storage	Type		Amount
	Ceph based distributed storage		31.5 TB
	QCOW2 storage		18.1 TB
Virtualization Technology			
Hypervisor	KVM		
Virtualization environment	OpenNebula		
User access	ECONE		
Web API	Yes		

To deploy the EOD pilot an isolated infrastructure was used in order to avoid the interference of other workloads in the system. This infrastructure was virtualized by making use of OpenNebula 4.14.2. The hardware details are described in Table II.

TABLE II

Physical Resources used In The Deimos Infrastructure

Physical resources			
Number of nodes	3		
Nodes	Processor	Memory GB	Storage GB
	Intel Core i7 – 2600 3.4 GHz	8	160
	Intel Core i7 – 2600 3.4 GHz	16	250
	Intel Core 2 6300 1.86 GHz	4	250
Storage	Type		Amount
	NFS		3 TB
Virtualization Technology			
Hypervisor	KVM		
Virtualization environment	OpenNebula		
User access	ECONE		
Web API	Yes		

B. *Experiment Description*

In papers [6] [7] an optimization of the process4EO was evaluated. It was demonstrated that the functionality of the EOD pilot remained the same in the runtime and that the parallelization of image processing could be done increasing scalability while reducing VMI size, VMI creation time, VMI delivery time and VMI deployment time.

In this experiment, the complete EOD pilot is optimized, (i.e. the monitor 4EO, the process4EO, the archive4EO and the user4EO) and tested.

The raw data used in this experiment remains the same of previous works to facilitate comparisons. Then, the raw data has 3090MB size; it contains information from four multispectral bands (R, G, B and NIR) and one panchromatic band; and the recorded area of the land surface is a rectangle of 8.86 x 16.59 km^2.

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

The aim of this experiment is to demonstrate the improved performance of the complete system by optimizing all the modules of the EOD Pilot and validate the use of the ENTICE middleware.

VII. Experiment Results

Four virtual machine images were created, one for every software module of the EOD Pilot: monitor4EO, process4EO, archive4EO and user4EO. The creation was done twice to evaluate the time required to create the machine both by following a manual process and by using an automated process. The size of the process4EO VMI was 1.99GB, the size of the monitor4EO VMI was 4.62GB, the size of the archive4EO VMI was 3.27GB and the size of the user4EO VMI was 1.40GB. TABLE I shows the results of the creation time required to create the VMIs with both the manual and the automated process. The automatic creation of the process4EO, the monitor4EO, the archive4EO and the user4EO was 51.54%, 52.40%, 44.32% and 36.61% fastest respectively. The user4EO took less time to be created because it only contained a service to access and visualize the catalogue of the processed images.

TABLE I
CREATION TIME (MINUTES)

	process4EO	monitor4EO	archive4EO	user4EO
Manual	42.65	70.17	51.40	27.97
Automated	20.67	33.40	28.62	17.73
% Reduction	51.54	52.40	44.32	36.61

The created VMIs with the automated process were delivered and deployed in the cloud infrastructure to measure the time required to be delivered and deployed.

Later, those VMIs were optimized with the ENTICE optimizer, delivered and deployed. The size of every VMI, the time required to deliver them and the time required to deploy them were measured to establish a comparison with the size, delivery and deployment of the non-optimized VMIs. Fig. 2 shows the size of the non-optimized VMIs and their size after the optimization carried out with ENTICE. The optimization of the process4EO has improved from previous works [6] and [7], the process4EO passed from 1.99GB to 1.06GB, which was a 46.73% of size reduction; in the monitor4EO the reduction was 35.93%, in the archive4EO 38.87% and in the user4EO 48.57%.

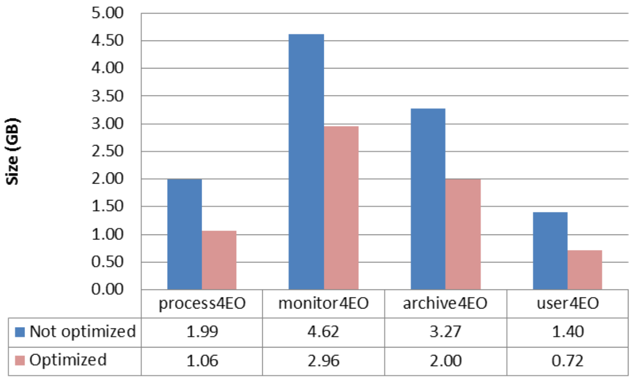

Fig. 2 Size of the VMIs - Not optimized vs Optimized

The delivery time of the non-optimized images compared with the delivery time of the optimized images is depicted in Fig. 3. The delivery time of the VMIs was reduced in all cases: 45.38% in the process4EO, 23.27% in the monitor4EO, 60.59% in the archive4EO and 45.96% in the user4EO.

The deployment time of the non-optimized images compared with the deployment time of the optimized images is depicted in Fig. 4. The deployment time was also reduced in all cases: 16.81% in process4EO, 39.92% in monitor4EO, 47.32% in archive4EO and 4.96% in user4EO.

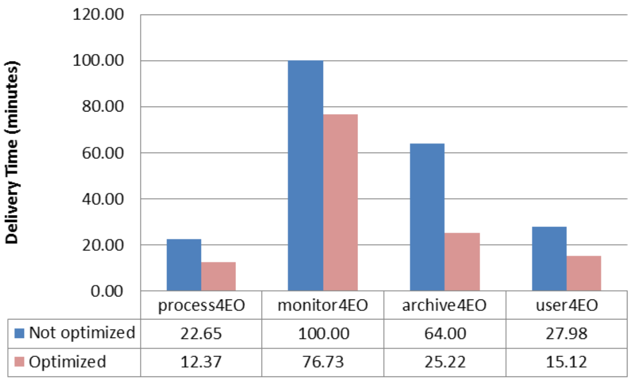

Fig. 3 VMIs delivery time (minutes) - Not optimized vs Optimized.

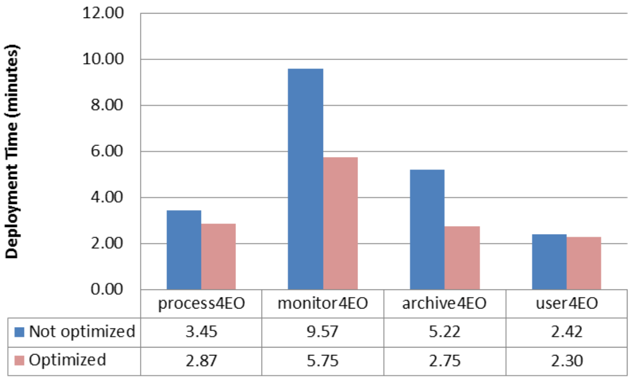

Fig. 4 VMIs deployment time (minutes) - Not optimized vs Optimized.

The size reduction, the delivery time reduction and the deployment time reduction of the optimized VMIs with regards to the not optimized VMIs in percentage are depicted in TABLE II.

TABLE II
REDUCTION IN METRICS (%)

	process4EO	monitor4EO	archive4EO	user4EO
Size	46.73	35.93	38.87	48.57
Delivery	45.39	23.27	60.59	45.96
Deployment	16.81	39.92	47.32	4.96

These results demonstrate that the optimization provides benefits in storage, delivery and deployment of the system. Besides, TABLE III compares the expected improvement of the ENTICE objective performances with the real improvement of the actual experiment. The size of the optimized VMIs did not reach 60% reduction; however the reduced size was between 35.93% and 48.57%, which was notorious. The VMI creation time of the VMIs was faster than the 25% expected; the use of the automated process reduced the creation time between 36.61% and 52.40%. The delivery time of the optimized VMIs was also faster than the 30% expected (60.59% faster in the delivery of the archive4EO). Finally, the deployment time was also faster than expected in the archive4EO and in the monitor4EO. However the deployment of the user4EO, although it improved, did not substantially change.

The results shown in TABLE III validate the use of ENTICE and their offered functionalities, demonstrating relevant reduction in VMI creation time, VMI size, VMI delivery time and VMI deployment time, which contributes to obtain a more efficient use of the needed cloud resources.

TABLE III
ENTICE OBJECTIVE VS REAL RESULTS

Metrics	**Expected Reduction**	**Real Reduction**
VMI Size	Up to 60%	Between 35.93% and 48.57%
VMI Creation Time	Up to 25%	Between 36.61% and 52.40%
VMI Delivery Time	Up to 30%	Between 23.27% and 60.59%
VMI Deployment Time	Up to 20%	Between 4.96% and 47.32%

VIII. CONCLUSIONS

This work culminates the integration of a Payload Data Ground Segment in cloud with the support of the ENTICE technology. The complete system was implemented in cloud and optimized with this technology.

The results demonstrated that the virtualization of a commercial system can be optimized in terms of size, creation time, delivery time and deployment time of the VMIs. ENTICE contributed to reduce the size of the EOD Pilot modules between 35.93% and 48.57%, the creation time between 36.61% and 52.40%, the delivery time between 23.27% and 60.59% and the deployment time between 4.96% and 47.32%. In the case of the creation time and the delivery time, they were faster than expected.

Therefore, the implementation of an Earth Observation Satellite PDGS can be flexible and scalable in function of the demand, as demonstrated in [6]; the delivery and deployment of the virtual machines can be done 60% and almost 50% faster respectively; the risk of oversizing or undersizing the infrastructure can be eliminated; the costs of using cloud resources can be reduced by optimizing the system, as done with the ENTICE technology; the user can be globally accessed with the application, and the vendor lock-in with public cloud infrastructure providers can be eliminated.

ACKNOWLEDGMENT

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

REFERENCES

[1] J. Hiestermann and S.L. Ferreira. "Cloud-based agricultural solution: A case study of near real-time regional agricultural crop growth information in South Africa," in *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 37th International Symposium on Remote Sensing of Environment*, Tshwane, South Africa, 2017, pp. 79-82.

[2] J. Becedas, R. Pérez and G. González. "Testing and validation of cloud infrastructures for Earth observation services with satellite constellations." *International Journal of Remote Sensing*, vol. 36 (19-20), pp. 5289-5307, May 2015.

[3] P. Patidar and A. Bhardwaj. "Network Security through SSL in Cloud Computing Environment." *International Journal of Computer Science and Information Technologies, vol. 2 (6), pp. 2800-2803,* 2011.

[4] E.S.A. Ahmed and A. Rashid. "A survey of big data cloud computing security." *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 3 (1), pp. 78-85, Dec. 2014.

[5] G. Kecskemeti, A. C. Marosi and A. Kertesz. "The ENTICE approach to decompose monolithic services into microservices," in *High Performance Computing & Simulation (HPCS), 2016 International Conference on. IEEE*, Innsbruck, Austria, 2016, pp. 591-596.

[6] J. Becedas, M. M. Núñez, D. González, “Optimization of an Experimental Cloud Infrastructure for Earth Observation Services” in *Recent Advances in Quantitative Remote Sensing,* Torrent (Valencia), Spain, Sep. 2017

[7] J. Becedas, M. M. Núñez, D. González. “Optimization of an Earth Observation Data Processing and Distribution System” in *Multi-purposeful Application of Geospatial Data*, 1st ed. Editors: R.B. Rustamov, S. N. Hasanova, M. H. Zeynalova, Ed. InTechOpen, ISBN 978-953-51-5823-3 (in press).

[8] D. Kimovski, N. Saurabh, V. Stankovski and R. Prodan. "Multi-objective middleware for distributed VMI repositories in federated cloud environment." *Scalable Computing: Practice and Experience*, vol. 17 (4), pp. 299-312, Aug. 2016.