Use Cases Towards a Decentralized Repository for Transparent and Efficient Virtual Machine Operations

Radu Prodan, Thomas Fahringer, Dragi Kimovski
University of Innsbruck, Austria
Email: {radu,tf,dragi}@dps.uibk.ac.at

Gabor Kecskemeti
Liverpool John Moores University, UK
Email: g.kecskemeti@ljmu.ac.uk

Attila Csaba Marosi
MTA SZTAKI, Budapest, Hungary
Email: marosi.attila@sztaki.mta.hu

Vlado Stankovski
University of Ljubljana, Slovenia
Email: vlado.stankovski@fgg.uni-lj.si

Jonathan Becedas
Satellite Systems, Spain
Email: jonathan.becedas@elecnor-deimos.es

José Julio Ramos
Deimos UK Ltd., UK
Email: jose-julio.ramos@deimos-space.co.uk

Craig Sheridan, Darren Whigham
Flexiant Ltd., Edinburgh, UK
Email: {csheridan,dwhigham}@flexiops.com

Carlos Rodrigo Rubia Marcos
Wellness Telecom S.L., Spain
Email: crubia@wtelecom.es

***Abstract*—Virtualization is a key enabling technology in Cloud computing that allows users to run multiple virtual machines (VMs) with their own application environment on top of physical hardware. It permits scaling up and down of applications by elastic on-demand provisioning of VMs in response to their variable load to achieve increased utilization efficiency at a lower operational cost, while guaranteeing the desired level of Quality of Service (QoS) to the end-users. Typically, VMs are created using provider-specific templates that are stored in proprietary repositories, leading to provider lock-in and hampering portability or simultaneous usage of multiple federated Clouds. In this context, optimization at the level of the virtual machine image is needed both by the applications and by the underlying Cloud providers for improved resource usage, operational costs, elasticity, storage use, and other desired QoS-related features.**

To overcome those issues, the ENTICE project [1] researches and creates a novel VM repository and operational environment for federated Cloud infrastructures. There exists a large variety of industrial applications that can strongly benefit by the ENTICE environment. In this paper we present an interesting selection of complementary use cases that drive the definition of the essential requirements for the ENTICE environment, and more importantly, validate the introduced innovations.

***Index Terms*—Virtual Machine images, Distributed VMI repository, Cloud federation.**

I. Introduction

Virtualization is a key technology in *Cloud computing* that allows users to run multiple *virtual machines (VMs)* with their own application environment on top of physical hardware. Virtualization enables scaling up and down of applications by elastic on-demand provisioning of VMs in response to their variable load to achieve increased utilization efficiency at a lower operational cost, while guaranteeing the desired level of *Quality of Service* (QoS, such as response time) to the end-users. Typically, VMs are created using provider-specific templates (so-called *VM images (VMIs*) that are stored in proprietary *repositories*, leading to provider lock-in and hampering portability or simultaneous usage of multiple federated Clouds. In this context, *optimization* at the level of the VMIs is needed both by the applications and by the underlying Cloud providers for improved resource usage, operational costs, elasticity, storage use, and other desired QoS-related features. We identify in this project *five critical barriers* that prevent many users from industry, business and academia to effectively use Cloud resources and visualized environments for their computing and data processing needs:

- manual, error-prone and time consuming VMI creation,
- monolithic VMIs with large deployment and migration overheads,
- proprietary unoptimised VM repositories,
- inelastic resource provisioning, and
- lack of information for effective VMI optimization.

The ENTICE project [1] researches and creates a novel VM repository and operational environment for federated Cloud infrastructures aiming to:

- simplify the creation of lightweight and highly optimized VMIs tuned for functional descriptions of applications;
- automatically decompose and distribute VMIs based on multi-objective optimization goals (performance, economic cost, storage size, and QoS needs) and a knowledge base and reasoning infrastructure to meet application run-time requirements; and
- elastically auto-scale applications based on their fluctuating load with optimized VM interoperability across Cloud infrastructures without provider lock-in,

in order to finally fulfill the promises that virtualization technology has failed to deliver so far, as presented in Section II.

There exists a large variety of industrial applications that can strongly benefit by the ENTICE environment. In the ENTICE project, we gathered an interesting selection of complementary *use cases* from to drive the requirements and to validate the researched environment, as follows:

- *Earth observation data (EOD)* [2] from one industrial partner called *Deimos Castilla La Mancha S.L. (DEIMOS)*, presented in Section III;
- *Image Repo* from one small and medium enterprise (SME) called *Flexiant Limited (FLEX)*, presented in Section IV;
- *Cloud service provider (CSP)* from another SME called *Wellness Telecom S.L. (WT)* presented in Section V;

II. Research Problems

Based on these considerations and the three use cases representing the focus of this paper, we research and develop an environment for decentralized and efficient VM operations named ENTICE, addressing the following research problems, also by showing how they surface in the WT use case.

A. Manual VM image creation.

VM image creation is still a largely manual process usually performed by experts with know-how in minimizing its management operations. Without expertise in operating system (OS) maintenance, the created VM images are likely to incur large overheads (e.g. in size, type or configuration) that limit the elasticity and auto-scaling capabilities of the application on the leased infrastructure. For example, VM images created by non-expert users commonly contain unnecessary software such as unused libraries, documentation and manual documents, or even complete desktop environment distributions (CDE, KDE, Gnome) that slow down the overall performance. For example, an average technician at WT needs approximately one hour to create a new VM image. If a VM image can be based on previously developed and configured images, its creation time can be reduced to around ten minutes which is still too long considering the highly dynamic demand for new images.

B. Monolithic (non-decomposable) VM images.

Currently, VM images are often treated as black-boxes and thus, are the smallest deployable components in Infrastructure-as-a-Service (IaaS) systems causing large deployment, instantiation, and migration overheads in the order of minutes or tens of minutes. Although VM images often consist of complementary and distinct parts (e.g. OS images, user software) with different functionality and usage frequency, there is currently no system that treats them as VM building blocks and exploit this feature to reduce management overheads (e.g. through caching and optimized distribution of commonly used components). For example, WT offers a platform for custom Software-as-a-Service (SaaS) solutions for external parties which usually build on custom technologies and create monolithic VM images. To save space, customers only store basic images that require additional manual work (e.g. reconfiguration) for every deployment. As a result, the custom SaaS offerings incur either significant VM preparation time, or resource penalties due to insufficient manual optimization.

C. Proprietary unoptimised VM repositories.

VM images are currently stored by Cloud providers in proprietary centralized repositories without considering application characteristics and their runtime requirements, causing high deployment and instantiation overheads. Moreover, users are expected to manually manage the VM image storage which is tedious, error-prone and time-consuming process especially if working with multiple Cloud providers. For example, every Cloud provider, including WT, is highly interested in attracting new customers from other providers. Unfortunately, current users must be familiar with WT's repository interfaces in order to use them, which is an unsurpassable barrier in deploying new images and exploiting the resources of WT. In other situations when WT's resources cannot fulfill certain resource demands, it is desired not to lose the customers, but to transfer and deploy part of their VMs on a different Cloud infrastructure, an operation hindered by similar problems. While VMI format converters do exist, they are ad-hoc solutions that do not preserve the performance and optimizations (Problem 1).

D. Inelastic resource provisioning.

Currently, scaling up and down applications on the Cloud involves latencies in the orders of minutes [3] for on-demand starting, stopping or migrating their VMs which is a too high overhead for applications with stringent realtime requirements. Most existing industrial applications, including the ENTICE use cases, have an *inelastic resource demand*. The percentage of QoS improvement as the result of scaling up or down the application is often lower than the percentage of change in resource provisioning for a certain VM allocation, which results either in low resource utilization, or in a higher number of QoS breaches. At WT for example, applications need to be manually scaled up or down based on the user load. Shutting down, manually assigning additional resources, and then powering up a VM again can take between $10 - 15$ minutes causing a high service interruption during this time.

E. Lack of information to support automated preparation and optimized deployment.

To support the overall lifecycle of applications deployed as VM images and to facilitate automatic optimization, setup and management in federated Cloud environments, complex information and reasoning mechanisms are needed including:

- functional and non-functional descriptions of the software components, including information on their computational, memory, storage, and communication complexity, and all the necessary dependencies, libraries and environmental variables needed for deployment on a VM;
- functional and non-functional QoS requirements such as speed, response time, cost, and storage space;
- dynamic information about the underlying federated Cloud infrastructure resources;
- information on the potentially new and resulting *Service Level Agreements (SLA)*, and other relevant information.

For example, WT needs to improve the software engineering process and user acceptance of its Cloud platform through automated and optimized preparation and deployment of applications based on adequate user and resource-provided information, while taking full advantage of the federated Cloud.

III. EOD Pilot Use Case

A. Introduction

The treatment of massive and large-sized data obtained from EO satellite recordings still represents a critical processing and distribution of big space data challenge. Remote sensing industries implement on-site conventional infrastructures to acquire, store, process and distribute the generated geo-information. However, these solutions do not cover sudden changes in the demand of services and the access to the information presents large latencies. For this reason, DEIMOS research focuses in the development of future Internet technologies to improve EO services and to highly reduce the costs associated with on premises deployment.

The EOD processing and distribution aims to implement the DEIMOS geo-data processing, storage and distribution platform of Deimos-2 satellite using Cloud technologies [2]. The main functionality of the system are:

a) Acquisition of raw data: When the imagery data is 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 processing.

b) Processing of the raw data: Once the data is ingested, it is processed in the product processors at several levels.

c) Archiving and cataloging geo-images: The different products obtained from the processing of raw data are archived and cataloged for delivery to the end-users or for providing high-added value services.

d) Offering user services: This is the front-end of the system that allows end-users to select the product to visualize or download, and provides high-added values in the imagery.

The EOD pilot case ports the already commercial `gs4EO` suit in a Cloud infrastructure. The suit is currently operational within the Deimos-2 satellite mission, and it is deployed in the Satellite Systems Centre in Puertollano, Spain. In the ENTICE project, Deimos intends to implement the system in a Cloud computing infrastructure to provide Deimos-2 satellite imagery to end users and carry out a pilot demonstration. The ENTICE environment is used to optimize the implementation and performance of the EOD pilot. It is expected that the optimization of the VMs highly contribute to provide autoscaling and flexibility to the ingestion of satellite imagery, its processing and distribution to end users with variable demands.

B. Requirements

Although the proven importance of EO in society, the access to the information obtained from satellites follows traditional and expensive paths to cover on-demand services for different potential customers such as conventional data centers and service distributors. This presents three drawbacks: (1) the cost of acquiring recent images of the Earth is very high, which limits the access of the general public, (2) clients do not have directly and fast access to the information because this has to be processed and distributed in an ad-hoc fashion, and (3) the service is not flexible to be adapted to sudden changes in the demand. Cloud computing represents a possible solution to improve common services and create new market opportunities because it is elastic and scalable on-demand through virtualization of resources for three reasons.

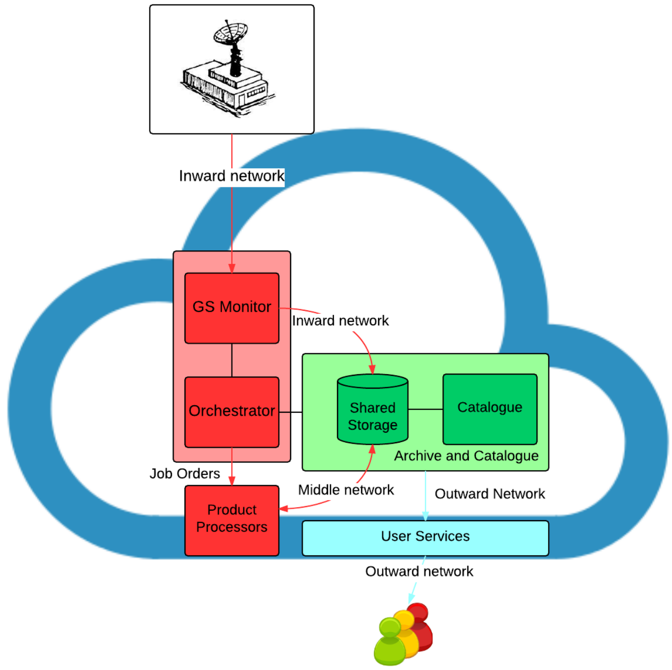

Fig. 1. The EOD architecture.

a) The global nature of EO data: with ground stations and users geographically located all over the world mandates the deployment of a worldwide infrastructure connecting all the stakeholders. Ground stations, ground control centers and data processing centers can take advantage of a rapid, agile, resilient and secure interconnected computer system for sharing the bulk of EO data. Users can take advantage of having data access points as close as possible to minimize the delay.

b) The massive size of EO data: generated by today's sensors, in the order of daily Terabytes, suggests the use of a cost-effective procurement of the computer infrastructure for archiving and processing by paying for resources on-demand, according to the occasional needs The EO data is received in batches for each receiving ground station on each "contact", at which point a satellite downloads all perceived data since the previous communication. Afterwards, the processing, archive and dissemination process of that received data are triggered and executed on the provided infrastructure.

c) The amount of resources: needed to process and distribute EO data to the global user community is large. Data processing can be greatly enhanced by using a large number of parallel computing resources for and for processing as many jobs as possible in a given time.

C. Architecture

The EOD architecture shown in Figure 1 is composed of the following components:

a) GSMonitor: ingests the available new raw data from the ground stations to the Cloud system and notifies the action to the Orchestrator. When it finishes the task, it notifies the end of the task to Orchestrator by using a message.

b) Orchestrator: 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 the outputs to be generated by the processors;
- To generate job requests using an XML description containing complete input and output format information;
- To poll for raw data in the ground stations and ingest it in a Cloud storage for distribution to the processing chain;
- To control the processing chain by communicating with the product processors;
- To manage the archive and catalog.

c) Product processors: are modules in charge of processing the raw data and products of the previous levels to produce image products. The four most important operations that the products processors perform on the input data are:

- *Calibration* that converts the pixel elements from instrument digital counts into radiance units;
- *Geometric correction* that eliminates distortions due to misalignment of the sensors in the focal plane geometry;
- *Geolocation* that compute the geodetic coordinates of the input pixels;
- *Ortho-rectification* that produce ortho-photos with vertical projection, free of distortions.

d) Archive and catalog: store and catalog the processed images for their distribution. It offers a catalog service for the Web (CSW) interface with the cataloged products.

e) User services: give access to products upon request or obtain high added values services by using Web services.

D. Benefits

The optimization of the VM deployment in ENTICE reduces the time of autoscaling the processing and distribution of satellite imagery. First, VM images and their storage are optimized for the needed QoS requirements. For the optimization, the necessary details including functional descriptors of EOD software components and quantitative information about their computational, storage and communication needs, are described with the ENTICE ontology and included in the knowledge base, allowing for an automatic generation of possible VM configurations at runtime. An optimal configuration may be found, for example in cases when some VMs are not permanently running, as the process of creating the VMs is triggered by the satellite revisit plan occurring once every three days over each region or receiving station.

Currently, VMIs include not only the product processors developed by DEIMOS, but also the OS and local storage associated to the VM. In case of autoscaling processing instances, the VMI is replicated as many times as required with all the redundant information associated to the OS and local storage. An optimization of the processing chain highly simplifies the VM and its replication in autoscaling tasks, mainly if there is variability in the ingestion of raw data.

The distribution of final products to the end-users intrinsically has variable demand. Although the number of users in a near future can generally be predicted, is unpredictable under exceptional circumstances for specific applications, for example in emergency situation scenarios. Autoscaling the service with non-optimized VMs storing redundant information provides higher latencies in the deployment of VMs and it implies a higher cost for deployment and storage.

Moreover, ENTICE can be used to dynamically optimize the runtime environment. Each time the satellite downloads its data on a receiving station, all processing chain tasks need to be recreated and start the data processing, cataloging and writing on the archive, requiring fast deployment of appropriate VMs close to the receiving station. Meanwhile, the user services, archive and catalogue tasks need to continuously serve read queries from their users, and quickly scale-out to cope with likely bursts in user demand. The expected lifetime of product processors instances depends on the amount of received data and the overall systems processing speed.

Finally, ENTICE finds the most cost-effective configuration that can cope with the hard and soft real-time constraints on data processing, while ensuring a good performance, cost and storage optimization trade-off. The overall process necessitates new SLA agreements, automatically prepared by ENTICE.

E. EOD problems

The problems of the EOD pilot related with five ENTICE research problems outlined in Section II are as follows:

a) Manual creation of VM templates: Whenever a new subsystem version gets released, engineers need to manually create VM templates with the required software packages taking hours to complete. The first ENTICE objective to automatically create of lightweight VM images through functional descriptions significantly improves the process, making them highly-specific and highly-tuned for minimal size and management overhead.

b) Monolithic VM images: EOD VM templates are all created on top of a baseline Red Hat Enterprise Linux distribution, including all software packages from the desktop suite, not tailored in any way to the target subsystems. This of course penalizes the overall VM performance and size. The second ENTICE objective to create and distribute of lightweight VM images decomposed into smaller reusable parts reduces the storage space and associated costs.

c) Proprietary un-optimized VM repositories: EOD systems uses a proprietary VM repository including all current and previous versions of the VM templates, which is not optimized in size. The third ENTICE objective is to develop autonomous multi-objective repository optimization processes which ensures that commonly used VM image parts are replicated and stored more widely, and thus, they are discovered and delivered from local repositories quicker, while user-provided parts come from a different external location.

d) Inelastic resource provisioning: EOD does not support dynamic resource provisioning on DEIMOS premises. There is no automatic scalability procedure for responding to changes in the supply and demand of data, or to the latency requirements of DEIMOS' products and clients. Manually shutting down VMs, assigning additional resources, and powering up a new and better VM could take as long as 20 minutes, causing a service interruption. The fourth ENTICE objective is to ensure elastic resource provisioning to improve the on-demand scaling of EOD in the Cloud in response to changes in the computation and storage requirements.

e) Automated preparation and deployment: All VM preparation and deployment tasks are performed manually, taking an unacceptable amount of time, longer than a complete processing pipeline. The fifth ENTICE objective is to create an information infrastructure for strategic and dynamic reasoning, to support automatic VM packaging of EOD based on criteria such as QoS functional properties, execution time, costs and storage and dynamic benchmark information about the underlying federated Cloud infrastructure.

IV. Image Repo Pilot Use Case

A. Introduction

Image creation and storage is still a massive challenge for Cloud providers such as FLEX. As new multi-Cloud technologies are arising, images need to be pulled and pushed from one location to another with a greater focus on transfer speed, image size, and storage requirements. To combat this need, the ENTICE environment develops a ubiquitous repository-based technology which provides optimized VM image creation, assembly, migration and storage. For this purpose, FLEX implements the Image Repo pilot case to test, validate and demonstrate that ENTICE can support and improve the performance image transfer and storage with the Cloud. The Image Repo pilot case mainly consists of the implementation in the Cloud of FLEX Cloud Orchestrator (FCO), currently operational within a FLEX testbed hosted in Edinburgh, Scotland. Within the ENTICE project, FLEX intends to implement the system in a Cloud infrastructure to provide smaller images and faster deployment times.

B. Requirements

There are three main challenges a service provider building a Cloud infrastructure must overcome:

- to offer continuous availability even when faults occur to maintain QoS and customer satisfaction;
- to ensure that the infrastructure is a help, not a hindrance to growing their Cloud business over the course of time;
- to achieve this while working under tight cost constraints.

With ENTICE, we expect to optimize the creation, storage and deployment of VMs with smaller lightweight images. This allows a reduction in the required storage to host the different VMIs. By using the ENTICE environment, the expected results of the implementation of this use case are:

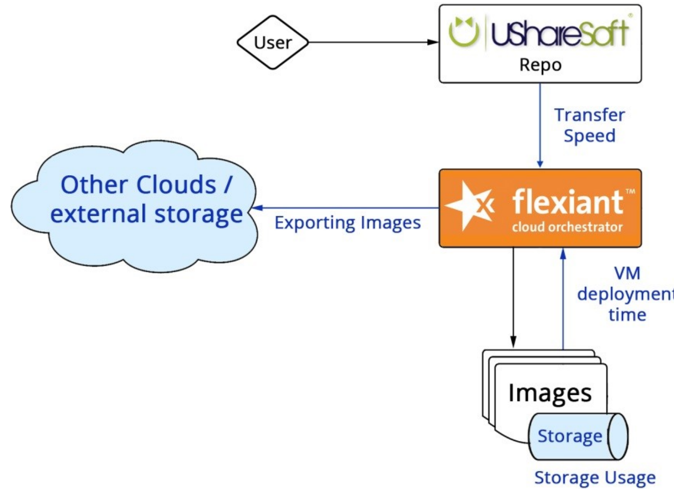

Fig. 2. The Image Repo architecture.

- to reduce delivery and deployment time of VM images;
- to decrease the total storage consumption;
- to achieve image portability and avoid vendor lock-in.

C. Architecture

Figure 2 shows the architecture of the Image Repo pilot case consisting from the following components:

1) UShareSoft Image Repo: is used by customers of FLEX to access a pre-configured range of VMIs for use within their own environment. These images can be downloaded locally or pulled into the target Cloud platform. UShareSoft enables fast, easy, self-service on boarding of software by Xloud users, a Cloud server template that is independent of any hypervisor or IaaS platform. UForge then automatically generates and publishes a final VMI to the hypervisor or Cloud of choice, in this case, FCO, ready for provisioning as a live VM. UForge also fully automates the full software delivery process, from build and generation to publish and maintenance. UShareSoft provides a number of benefits such as:

- ability for users to create simple and complex software solutions with self-service provisioning to the Cloud;
- migration services which extract information from VMs running in another environment, ready for instant repackaging to the providers Cloud;
- a Cloud software marketplace of ready-to-run applications, where developers and end-users can create, share, clone, buy and customize Cloud server templates;
- injection of managed services on-demand when a user purchases a Cloud server template.

2) Flexiant Cloud Orchestrator (FCO): [4] is a world-leading Cloud orchestration software solution, which enables service providers and enterprises to design, create and manage their own virtual public, private or hybrid Cloud solutions, with a front-end user interface and access. FCO provides an API to perform actions such as creating and managing VMs and migrating images into the platform. FCO can manage the entire Cloud solution, from hardware, network and storage management through to metering, billing and end-user self-service. It can also be used to augment and orchestrate existing platforms such as VMware vSphere.

3) Image storage: within the FCO, backend storage is used to store the images used within the platform. Storage is well-known to be one of the major, if not the main infrastructure challenge and, for many years, service providers could choose only between two options: either local or centralized network storage. FLEX provides excellent distributed CEPH storage capabilities to ENTICE which solves many recent local and centralized Cloud storage-related issues by using adaptive load balancing, whereby frequently accessed objects are replicated over more nodes.

D. Benefits

ENTICE provides to FLEX environment, functional and non-functional requirements to implement industry driven requirements for image use with the Cloud. ENTICE also provides information to improve the transfer speed of images and improve the deployment time of VMs and to reduce the storage required for images and, as a result, a cost saving for the Cloud operator. Through the Image Repo pilot, ENTICE gets tested within in a commercial Cloud offering and the benefits detailed for all Cloud providers to use.

The Image Repo pilot provides ENTICE a use case to deploy smaller images within its Cloud platform showcasing faster transfer speeds and a reduced storage requirement, as well as investigating the performance gains on VM deployment times. In case of Image Repo pilot case, the main aspects that ENTICE optimizes are:

- the size that each image needs to be stored;
- the transfer time of images;
- the deployment time that VMs need to be fully operative;
- the possibility to avoid vendor lock-in with images that are easily transferable.

ENTICE has a clear relevance to the IaaS providers and could potentially be part of an interoperable ecosystem of Clouds, where images are easily optimized, traded and deployed. Indirectly, SaaS providers benefit from the results, so this group of providers as well appear as potential relevance market objective. The relevant market details are identified and addressed as the project develops, based on results and market research from the different use cases that are implemented.

E. Image Repo Problems

Nowadays, the implementation of the Image Repo pilot case in the actual Cloud platforms suffers from several problems.

a) VM deployment time: is a key issue addressed in the project. FLEX looks into lowering the time taken to deploy a VMI using the ENTICE framework by 33%. This allows its customers to deploy faster, resolve issues, and meet demand quicker. Since the ability to export images from FCO is limited, this possibility of using the ENTICE framework to allow the exporting of images from the FLEX platform is a key investigation of the project. This ability is essential to avoid vendor lock-in and allow a multi-Cloud approach.

b) Image storage: is becoming a critical issue ever more images are used in a Cloud platform. To combat this, FLEX uses new image compression methods to reduce this overhead which allows its customers to reduce storage requirements and costs, estimated to be 10% of the total storage used.

c) New and improved image types: are required by FLEX within its Cloud platform. With newly created image types within the ENTICE project, FLEX looks to supporting these versions within future versions of FCO.

V. CSP Pilot Case

A. Introduction

The CSP pilot use case aims to create a viable solution for optimizing the operational and infrastructure resources required for provisioning proprietary and third party SaaS applications. The pilot consists of different services that WT offers to its costumers to have a clear perspective on how any CSP could benefit from the use of the ENTICE environment. The main services to be tested within this use case are:

a) Unified Communications (UC): [5] is an open source framework for enterprises aiming to establish a full collaboration inside companies. The main challenge is to design an architecture that safe, elastic, scalable and high available.

b) Alfresco: [6] is a free fully managed enterprise content management solution for critical documents. The main challenge of this component is to optimize the VMI configuration and dependencies with Cloud providers to enable on-demand scaling on different Clouds.

c) Redmine: [7] is a free open-source Web application for project and issue management. The main challenge is to optimize the VMI configuration procedures to provide the necessary elasticity to support the dynamic system load at all times.

B. Requirements

The requirements of this use case are to:

- optimize VM image creation and storage;
- optimize VM image configuration procedures;
- optimize VM deployment;
- eliminate the dependencies with other providers;
- improve elasticity for on-demand scaling.

C. Architecture

The CSP use case architecture has been designed to simultaneously support multiple different services, thus inducing high system complexity and convoluted resource management. The provided services and the systems itself are divided and distributed into multiple modules performing specific workloads. Figure 3 shows the architecture for the CSP use case consisting of the following components:

1) Session initiation protocol (SIP) proxy-core: capable of providing secured TLS communication, encryption and network address translation. Furthermore, it supports instant messages communication and call admission control.

2) SIP proxy-edge: encapsulating the SPI server, the registering and location server and the load balancing service.

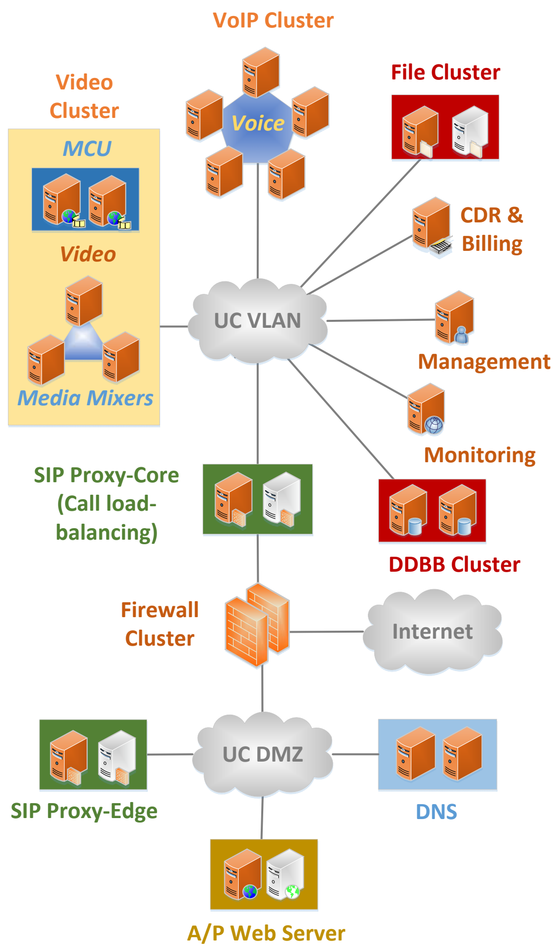

Fig. 3. The UC architecture.

3) VoIP cluster: composed of highly optimized media servers for voice of Internet services.

4) Video cluster: wrapping multipoint control unit and media mixer servers.

5) Database cluster: for user authentication and storage configuration.

6) Public web portal: secured by specialized high availability firewall.

D. Benefits

The ENTICE environment provides a feasible solution to optimize the whole ecosystem of a CSP. By using ENTICE environment, WT not only highly decreases the deployment and customization of its VMIs but also increases the QoS offered to its customers by offering a faster and elastic service. Thanks to the ENTICE environment, CSP is able to offer more competitive services, reduce operational costs and avoid vendor-locking.

E. Problems

In their daily work, Cloud service providers deal with several major critical problems that require dedicated resources for the provisioning of services to their costumers.

a) Manual VM image creation: For an average technician at WT, the creation time of a new VMI takes approximately one hour. The ENTICE environment can solve this problem providing previously configured and developed images, so that their creation and deployment time can be highly reduced to around ten minutes, which is still too long considering the highly dynamic demand for new images.

b) Monolithic (non-decomposable) VMIs: WT offers a platform for custom SaaS solutions for external parties, which usually build on custom technologies and create monolithic VMIs. To save space, customers only store basic images that require additional manual work (e.g. reconfiguration) for every deployment. As a result, custom SaaS offerings incur either significant VM preparation time, or resource penalties due to insufficient manual optimization.

c) Proprietary unoptimised VM repositories: Every Cloud provider, including WT, is highly interested in attracting new customers from other providers. Unfortunately, current users must be familiar with WTs repository interfaces for using them, which is an unsurpassable barrier in deploying new images and exploiting the resources of WT. In other situations when WTs resources cannot fulfill certain resource demands, it is desired not to lose the customers, but to transfer and deploy part of their VMs on a different Cloud infrastructure, an operation hindered by similar problems. While VMI format converters do exist, they are ad-hoc solutions that do not preserve the performance.

d) Inelastic resource provisioning: At WT, applications need to be manually scaled up or down based on the user load. Shutting down, manually assigning additional resources, and then powering up a VM again can take between 1015 minutes causing a high service interruption during this time.

e) Lack of information to support automated preparation and optimized deployment: WT needs to improve on the engineering process and user acceptance of its Cloud platform through automated and optimized preparation and deployment of applications based on adequate user and resource information, while taking advantage of the federated Cloud.

VI. Metrics

Monitoring of the ENTICE environment is an essential activity that provides crucial information about the status of the Cloud federation. Furthermore, it enables proper functioning of the multiple involved processes, such as repository optimization and VMI fragmentation. Conclusively, it provides key insides for maintaining the Pareto SLA in accordance with the requirements pilot applications.

Based on the ENTICE's main research objectives, we have identified the following categories of success metrics for validating the functionality of the researched environment: (1) VMI-related metrics, (2) elastic resource provisioning metrics, including VM-related metrics, service-related metrics and network-related metrics, and (3) federation-wide metrics.

The VMI-related metrics gather all online measurements for the creation time, size reduction and delivery time, both for optimized and unoptimized VMIs. These metrics are focused

TABLE I
VMI-RELATED METRICS AND SAMPLE MEASUREMENTS FOR TWO VMIS IN THE ENTICE ENVIRONMENT.

	Creation time		Size (content)		Delivery	
VMI	*Original*	*Improved*	*Original*	*Improved*	*Original*	*Improved*
Wordpress	60 min	17 min	2.3 GB	0.8 GB	210 s	73 s
Redmine	60 min	17 min	1.55 GB	713 MB	190 s	68 s

TABLE II
ELASTICITY RESOURCE PROVISIONING METRICS FOR THE ENTICE ENVIRONMENT.

Metric type	*Metric*	*Unit*
Service metric	Response time	s
	Throughput	Responses per request
VM metric	RAM usage	MB
	CPU usage	%
	Disk usage	MB
Network metric	Packet delay	ms
	Packet loss	%
	Jitter	Absolute Jitter

towards validation of the ENTICE objectives in relation to the functionality for lightweight VMI creation, VMI fragmentation, and size optimization. Table I presents the measurement units for the VMI-related metrics and provides a monitoring example for two VMIs: one from the CSP pilot (Redmine) and one from the Repo pilot use case (Wordpress).

Table II provides the elastic resource provisioning metrics with the corresponding units. These measurements encompass the QoS monitoring of the functional and non-functional requirements, thus conveying capability to evaluate the ENTICE environment in accordance with the auto-scale functionality.

Finally, Table III presents the federated repository related metrics, which encircle the monitoring process of the ENTICE environment and its efficiency, both from the financial and performance point of view. These metrics will allow us to properly assess the functionality for optimized VMI distribution across the ENTICE repositories.

VII. CONCLUSIONS

The ENTICE project aims to research and create a novel VM repository and operational environment for federated Cloud infrastructures aiming to:

- simplify the creation of lightweight and highly optimized VMIs tuned for functional descriptions of applications;
- automatically decompose and distribute VM images based on multi-objective optimization (performance, economic costs, storage size, and QoS needs) and a knowledge base and reasoning infrastructure to meet application run-time requirements; and
- elastic auto-scale applications on Cloud resources based on their fluctuating load with optimized VM interoperability across Cloud infrastructures and without provider lock-in, in order to finally full the promises that virtualization technology has failed to deliver so far.

We presented in this paper a carefully-chosen set of use cases from three companies aiming towards testing, validation, and demonstration of the ENTICE environment, as follows:

TABLE III
ENTICE FEDERATION-WIDE METRICS

Metric	*Unit*
Storage space	MB
Storage cost	€
VMI transfer time	s

- DEIMOS as an SaaS provider with different services aiming to demonstrate the efficiency and usability of the results of the project;
- FLEX as IaaS provider aiming at a better portability and efficiency of the resources of their Cloud offerings that aid service providers to reach better customer support with versatile and efficient means to upload, utilize and transfer VMIs;
- WT which is both a SaaS and IaaS provider, who plans to demonstrate portability, efficiency, and elasticity of the ENTICE environment.

The envisaged set of pilot applications comprise a mix of proprietary (EOD, FCO) and external-party (Unified Communications, Redmine, Alfresco) SaaS applications selected from the most demanded services by SaaS providers. These applications allow us to demonstrate the optimization of the VMIs for the needed QoS and deployment requirements in terms of number of users and storage.

ACKNOWLEDGEMENT

This work is funded by the European Union's Horizon 2020 research and innovation programme under the grant agreement 644179 ENTICE: dEcentralized repositories for traNsparent and efficienT vIrtual maChine opErations.

REFERENCES

[1] S. Ostermann, R. Prodan, T. Fahringer, G. Kecskemeti, S. Taherizadah, and V. Stankovski, "Decentralised repositories for transparent and efficient virtual machine operations: Architecture of the ENTICE project," in *Developing Interoperable and Federated Cloud Architectures*, G. Kecskemeti, A. Kertész, and Z. Nemeth, Eds. IGI Global, April 2016, ch. 7, pp. 170–219. [Online]. Available: http://www.igi-global.com/chapter/decentralised-repositories-for-transparent-and-efficient-virtual-machine-operations/149696

[2] J. J. Ramos and J. Becedas, "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*, P. Soille and P. G. Marchetti, Eds. European Spance Agency, March 2016, pp. 107–110.

[3] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of Cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 931–945, June 2011. [Online]. Available: http://ieeexplore.ieee.org/iel5/71/4359390/05719609.pdf

[4] (2016, May) Flexiant cloud orchestrator. Flexiant Ltd. Edinburgh, UK. [Online]. Available: http://docs.flexiant.com/

[5] W. A. Flanagan, *VoIP and Unified Communications: Internet Telephony and the Future Voice Network*. Emereo Publishing, April 2012. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118019210.html

[6] M. Bergljung, *Alfresco CMIS*. Packt Publishing, March 2014. [Online]. Available: https://www.packtpub.com/web-development/alfresco-cmis

[7] A. Lesyuk, *Mastering Redmine*. Packt Publishing, January 2013. [Online]. Available: https://www.packtpub.com/application-development/mastering-redmine