

Towards an Environment for Efficient and Transparent Virtual Machine Operations: The ENTICE Approach

Dragi Kimovski*, Nishant Saurabh*, Sandi Gec†, Polona Štefanič†, Gabor Kecskemeti‡, Vlado Stankovski†, Radu Prodan* and Thomas Fahringer*

* University of Innsbruck,

† University of Ljubljana,

‡ Hungarian Academy of Sciences

Email: dragi@dps.uibk.ac.at

Abstract—Cloud computing is based on Virtual Machines (VM) or containers, which provide their own software execution environment that can be deployed by facilitating technologies on top of various physical hardware. The use of VMs or containers represents an efficient way to automatize the overall software engineering and operation life-cycle. Some of the benefits include elasticity and high scalability, which increases the utilization efficiency and decreases the operational costs. VMs or containers as software artifacts are created using provider-specific templates and are stored in proprietary or public repositories for further use. However, technology specific choices may reduce their portability, lead to a vendor lock-in, particularly when applications need to run in federated Clouds.

In this paper we present the current state of development of the novel concept of a VM repository and operational environment for federated Clouds named ENTICE. The ENTICE environment has been designed to receive unmodified and functionally complete VM images from its users, and transparently tailor and optimise them for specific Cloud infrastructures with respect to their size, configuration, and geographical distribution, such that they are loaded, delivered, and executed faster and with improved QoS compared to their current behaviour. Furthermore, in this work a specific use case scenario for the ENTICE environment has been provided and the underlying novel technologies have been presented.

I. INTRODUCTION

Cloud computing is based on Virtual Machines (VM) or containers (CI) which provide their own software execution environment that can be deployed by facilitating technologies on top of various physical hardware [1]. The use of VMs or containers represents an efficient way to automatize the overall software engineering and operation life-cycle. Some of the benefits include elasticity and high scalability, which increases the utilization efficiency and decreases the operational costs [3]. VMs or containers, as software artifacts, are created using provider-specific templates and are stored in proprietary or public repositories for further use. However, technology specific choices may reduce their portability, lead to a vendor lock-in,

particularly when applications need to run in federated Clouds.

Unfortunately, current state-of-the-art does not provide any substantial means for streamlined adaptation of distributed applications in federated Cloud environments [2]. Overall, multiple critical barriers that largely effect the usability of federated Cloud infrastructures can be identified: (i) manual, error-prone and extensive time consuming VM image creation with proprietary templates, (ii) monolithic VM images with large deployment and migration overheads, (iii) proprietary unoptimised VM repositories, and (iv) lack of information to support effective VM image optimization [4].

In this paper we present the current state of development of the novel concept of a VM repository and operational environment for federated Clouds named ENTICE.¹ The ENTICE environment has been designed to receive unmodified and functionally complete VM images from its users, and transparently tailor and optimise them for specific Cloud infrastructures with respect to their size, configuration, and geographical distribution, such that they are loaded, delivered, and executed faster and with improved QoS compared to their current behaviour. The proposed environment has been focused towards overcoming the barriers that prevent the wide usage of federated Cloud infrastructures and it aims to: (i) simplify the creation of light weight and highly optimised VM images tuned for functional descriptions of applications; (ii) automatically distribute VM images based on multi-objective optimisation and a knowledge base and reasoning infrastructure to meet application runtime requirements; and (iii) provide interface which enables users to manage their VM images in federated Cloud infrastructures without provider lock-in. Furthermore, in this work a specific use case scenario for the ENTICE environment has been provided and the underlying novel technologies have been presented.

¹<http://www.entice-project.eu/>

II. USE-CASE SCENARIO AND REQUIREMENTS

As discussed in Section I, in order to overcome the barriers that limit the possibilities for Cloud federation, we present a specific use-case scenario, which is highly relevant for proper definition of the ENTICE design requirements. Furthermore, based on these requirements, we list down the imminent functionalities specific to each architectural component of ENTICE.

The use-case of ENTICE environment at the outset, typically initiates with the Application Provider uploading VM Image encompassed with appropriate application functionalities. Henceforth, an easy interface to the user which integrates to the functionalities including image optimization, storage and the knowledge oriented image portal with enhanced reasoning based module for easy access of services (example: upload, download, update, optimize etc.) and VM image management is required. In general, user specific images are large sized comprising of redundant functionalities incurring high storage cost resulting to image distribution and instantiation overheads. Henceforth, the splitting of an image into multiple fragments serving common functionality and storing them only once is necessitated to reduce the redundancy. Furthermore, the repository optimization with regards to the storage location of images evaluating performance metrics of each attached repository medium is imminent for faster VMI distribution across multiple clouds. Based on these factors, we identify the generalized requirements for the ENTICE environment. Hence, we divide the ENTICE design requirements in the following broad categories: (i) Application data-related requirements, (ii) Security-related requirements, (iii) VMI and fragment management-related, (iv) Information and metadata management-related requirements, and (v) QoS metrics-related.

The Application-data related requirements encompass the delivery of various content alongside the VM image, for the efficient execution of the corresponding application. To this extent, the ENTICE environment provides a functional descriptions tool that contains references to the additional VM image content and evaluate its presence in terms of files, folders, sizes, distribution and etc.

Furthermore, the ENTICE environment needs to address various security measures, to prevent unauthorized access of VMIs. For this purpose, the repository must support content delivery through encrypted channels. Moreover, the analysis methods for VMIs of the ENTICE environment should not compromise the privacy and security of the Cloud application owners and should not expose any user data, which should be securely stored in the knowledge base.

One of the requirements that ENTICE repository holds is to provide fast VM image delivery, hence identify the VMIs and fragment management-related requirements. The management module requires efficient integration of specific modules for VM image and meta-data delivery,

with the a Multi-objective optimization framework for online and offline distribution of the data. In addition, the ENTICE environment needs emphasis to support the delivery of data in the form of files or folders containing user specifications, thus requiring relevant informations and metadata management. In such cases, ENTICE must provide appropriate storage and data delivery mechanism together with the VMIs. Further, the data needs to be indexed, so that the appropriate distribution techniques allow efficient delivery to relevant geographical locations. Hence, it is required to maintain index hashing of VMIs and corresponding fragments. The semantic model represented in the Knowledge Base requires to store information about the VMIs and their functionality, the geographic location, the URI, and other details for the search facility.

Lastly, monitoring of QoS metrics is necessary to be able to observe the time needed to move VMIs and/or fragments among repository locations and to deploy a VMI needed by a particular Cloud application. A further requirement is to monitor the infrastructure on which the ENTICE repository is deployed to ensure that the scaling and speed of operation is adequate. Hence, based on the discussed requirements, we list down important use-case functionalities for each ENTICE architectural components relevant to the requirements explained above. The components are explained in detail in Section III.

1) *VM Image synthesis, analysis and storage:*

- *Identification of optimization potential of VM Images by using functional descriptions provided by the users;*
- *VM Image size optimization by removing parts of the OS which are not required and have not been defined in the functional description*
- *Division of the VM images in multiple fragments with respect to the required functionalities, thus removing duplicated software packets across multiple images*
- *Assembly of Fragments into complete VM Image during the deployment stage*

2) *Federated Image Repository with multi-objective optimization:*

- *Applying Multi-objective Optimization for distribution of the VM Images and fragments across distributed storage sites*
- *Optimized extraction of fragments, based on functionality, for assembly of the VM Images during the deployment stage*
- *Movement of VM image from one storage location to another*
- *Location Tracking of each VM Image to enhance storage and distribution*
- *VMI conversion from one provider specific template to another to achieve interoperability over multiple cloud*

3) *Knowledge based VMI portal modelling and reasoning:*

- *An easy interface for the users to access available VM Image and to search fragment*

- VM Image upload and update functionality for the streamlined provision of user's images
- VM Image management and graphical analysis tool for the Pareto-SLA module, which allow for the users to choose SLA's from multiple optimal metrics provided by the Multi-objective optimization framework
- Funtionality to sort VM Images with respect to user defined metrics
- Easy interface to User profile management

III. ARCHITECTURE

The ENTICE environment has been designed to receive unmodified and functionally complete VM images from users, and transparently tailor and optimise them for specific Cloud infrastructures with respect to their size, configuration, and geographical distribution, such that they are loaded, delivered, and executed faster and with improved QoS compared to their current behaviour. ENTICE gradually stores information about the VMI and fragments in a knowledge base that is used for interoperability, integration, reasoning and optimisation purposes.

VM images management is supported by ENTICE at an abstract level, independent of the middleware technology supported by the underlying Cloud computing infrastructure. To further shield the users from the complexity of underlying Cloud technologies and simplify the development and the execution of complex use cases, ENTICE focuses on providing the flexibility for tailoring the VM images to specific Cloud infrastructures.

The ENTICE repository includes, among other features, techniques to optimise the size of VM images while maintaining their functionality, automatically share images (or parts of the images) among repositories (even in multiple administrative domains or cloud infrastructures), and optimally deploy them in response to application and data centre requirements.

The ENTICE environment, depicted in Figure 1, aims to prove a universal backbone for IaaS VM management operations, which accommodate the needs for different use cases with dynamic resource and other QoS requirements. The ENTICE technology is completely decoupled from the applications and their specific runtime environments, but continuously supports them through optimised VM image creation, assembly, migration and storage.

In what follows, a detailed description of the essential components of the ENTICE environment and the current development status will be presented.

A. Interoperable and decentralised VM image synthesis, analysis and storage

The ENTICE VMI synthesis services (as seen in Figure 2) allow the creation of new virtual machine as well as container images with several approaches. First, accomplished through the duties of the Recipe based image creation service, it allows the use of generic user provided images (the so called original images) or (software) recipes

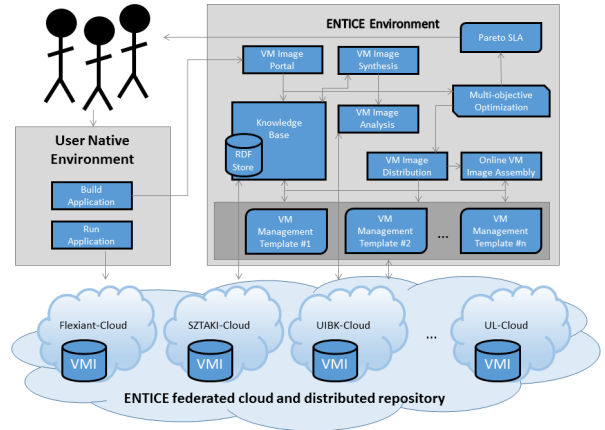


Fig. 1. Top level view of the ENTICE environment

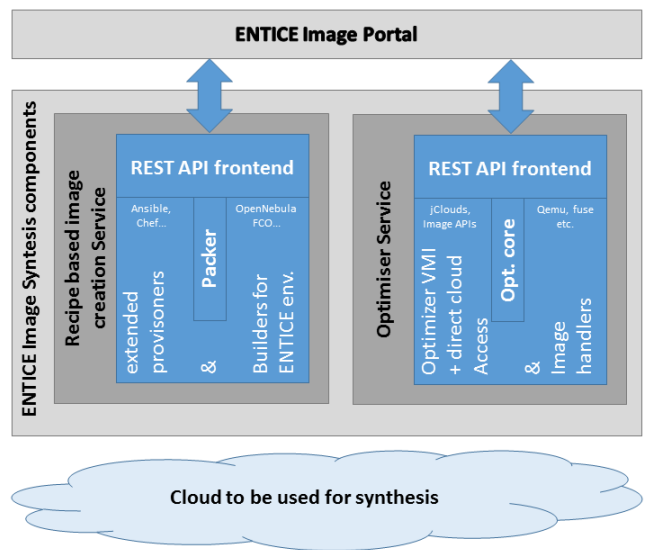


Fig. 2. Overview of the ENTICE Image synthesis components

to act as the foundation of specialisation. Next, the Optimiser service will collaborate with the image portal to identify the functional requirements a newly created image must meet. Then project's optimisation technique will alter the user provided images either directly or indirectly (through altering recipes). These alterations will target partial content removal from the original images allowing them offering only their single purpose, namely the functions identified in the image portal. Once the initial optimised image is ready, the synthesis service offers image maintenance operations (like managing software updates on the image).

Next to synthesis, ENTICE will introduce VMI analysis functionality to allow discovery of identical portions in apparently unrelated VM images coming from even different stakeholders and communities, regardless of the Cloud Provider where they are physically stored [7]. This information will be automatically stored in the ENTICE

knowledge base for later use. The environment will also allow splitting of VM images into fragments for storing the frequently shared image components only once (e.g. a particular flavour of Linux used by two different images), this operation allows the VM image distribution component to optimize the overall storage space throughout the distributed repository. As VM images are now fragmented, the ENTICE will distribute VMMT to the various repositories in the ENTICE environment. The VMMTs will allow VMs be assembled at runtime from the previously identified fragments. The templates will be stand-alone VM images containing the necessary components to access the distributed repository. After a VM is instantiated from a VMMT, it will customise the contents of the instantiated VM with the VM fragments required to match the user’s functional requirements (even allowing new files/directories be placed in particular VMs to meet the demands of the various stakeholders).

B. Federated multi-objective VM image repository optimization

In this subsection we provide a description of the multi-objective optimization (MOO) framework for VMI distribution in Federated Cloud repositories. The optimization framework can be applied on multiple distinctive application levels:

- 1) Initial VMI distribution;
- 2) Offline VM image redistribution;
- 3) Online VM image redistribution;
- 4) VM image discovery and assembly.

The framework is encompassed around unified multi-objective optimization module, which can be utilized for multiple different optimization purposes. Internally, the optimization module is branched in four distinctive sub-modules. Each of the sub-modules has been tailored specifically for a given task. The “Initial Distribution” sub-module covers the multi-criteria evaluation of the possible repository sites where the VMIs or associated data sets can be initially stored. This module, provides a tool which mitigates the process of initial VMI upload, when the available storage sites possibilities are so large that can overwhelm the user during the decision process. Afterwards, the “Offline VMI Redistribution” sub-module encapsulates the optimization of the VM images and fragments distribution within the federated repository sites. By taking into account the VMIs usage patterns, the algorithm is capable of providing multiple trade-off solutions, where each solution represents a possible mapping between the stored fragments/images and available repository sites. Unlike the initial image distribution, the problem of “Offline VMI redistribution” consist of a finite, but very large, number of combinatorial alternatives, which are not known in the beginning of the solving process. The optimization process is conducted by utilizing two conflicting objectives: cost for storing and transferring of the data, which we simply call Cost

objective and Performance objective. Lastly, the “Online VM image Redistribution” sub-module aims on dynamical redistribution of particular VM images/fragments during particular application execution. This sub-module is only applied for user’s specific VM images and in accordance with the technicalities of the application that is being executed. The framework is dependent on the repository’s usage patterns to properly optimize the distribution of the VM images. To this aim a specific module is required to store information on the previous transfers within the federation and to provide the collected data in a proper format. The ontology based Knowledge Base has been utilized for the purpose of the multi-objective optimization framework. The framework has been designed to acquire input data from the knowledge base, and also to return the output results there. Moreover, a specific monitoring agent has been provided for proper documentation of the data transfers. The monitoring tool itself can be realized in multiple different manners, and its exact structure will be elaborated with the maturing of the ENTICE environment. Furthermore, the framework provides a service based API, through which the Decision Maker (DM) can access the list of optimal Pareto solutions in a guided manner, thus reducing the complexity of the VMI storage management process [6]. The high level structure of the optimization framework has been presented on Figure 3. The proposed framework has been developed by leveraging the jMetal Multi-objective optimization library [5].

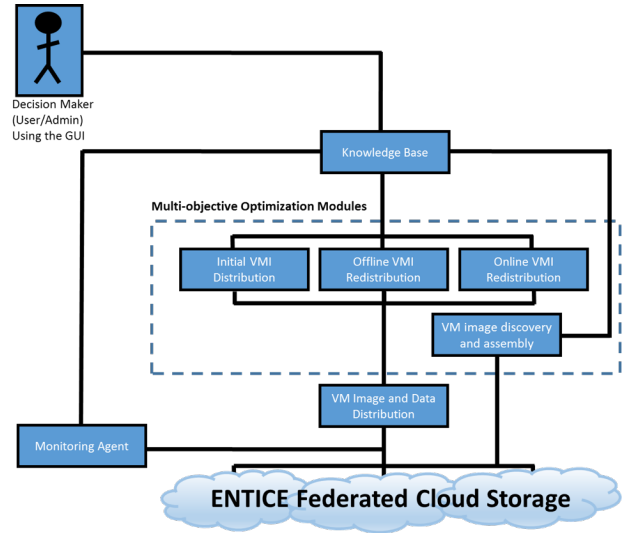


Fig. 3. Top level view of the Multi-objective Optimization Framework for VM Image distribution

C. Knowledge-based modelling and reasoning

An knowledge base has also been designed to support the ENTICE runtime by providing latest statistics and logistics information to the VMI distribution. RDF Schema (RDFS) provides a data-modelling vocabulary for RDF data. Actually, RDFS is an extension of the basic

RDF vocabulary. On top of RDF, the simple schema language RDFS has been defined to offer a distinguished vocabulary to model class and property hierarchies and other basic schema primitives that can be referred to from RDF models. Both RDF and RDFS constitute a newly emerging standard for metadata that is about to turn all kinds of information (structured, semi-structured and even unstructured data) [8] into a machine-understandable knowledge base. RDFS is an XML application that allows for the denotation of facts and schemata in a machine-readable format, building on an elaborate object model for describing concepts and relations. In other words, RDFS exploits the means of XML to allow for disjoint namespaces, linking and referring between namespaces and, hence, is a general methodology for sharing machine-processable knowledge in a distributed setting. To phrase the role of RDFS in knowledge engineering terminology, it defines a simple ontology for checking particular RDF documents against to determine consistency. The ENTICE ontology is depicted in Figure 4 and consist of entities and their relationships identified so far. Moreover, the Web Ontology Language (WOL) targets applications that need to process the content of information instead of just presenting information to humans. WOL facilitates greater machine interpretability of content than that supported via XML by providing additional vocabulary along with a formal semantics.

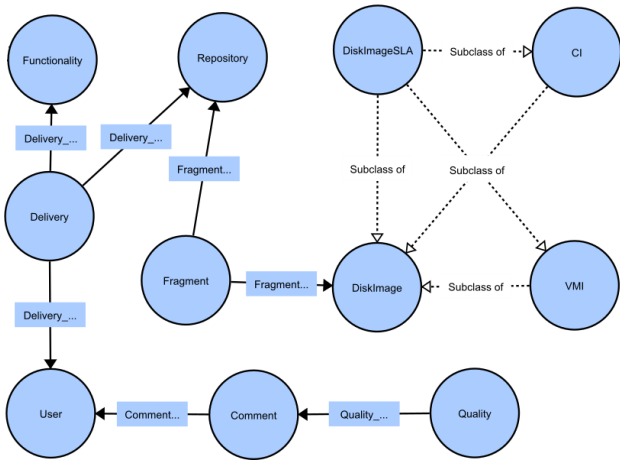


Fig. 4. ENTICE ontology showing main entities and their relationships

Particular care will be put on the capabilities of reasoning mechanisms. Different independent reasoning mechanisms will be applied (e.g. Hermit, Pellet, etc.) over the entire ontology, thus to ensure ontology consistency. The ontology will be gradually improved by identifying redundancy of information in the knowledge base and finding possible conflicting data by using reasoning mechanisms such as RDFS and SWRL rule engines, forward/backward chaining and others [9].

D. VMI portal and GUI

Since the user interface enables the users to interact with the computing systems efficiently, GUI represents an important part of the ENTICE software development.

The ENTICE VM image portal, through dynamic HTML5 based GUI can currently provide the following functionalities:

Landing (home) page. On the initial (home) page, there is basic information about the ENTICE project and environment, such as what is and what does ENTICE offer (e.g. optimization and deployment of VMIs/ CIs in distributed repositories). In the footer at the bottom of the page can be found further links to external information of the ENTICE, such as links to the terms and conditions, social media networks, FAQ, related news etc. On the top of the page is navigation menu, where users can route to other subpage with useful functionalities of the VMI portal.

Explore (images) page. Here a list of all public VM images with basic and advanced search options can be found. Cloud providers, Application Developers (users) will be able to explore the collection and filter results according to different criteria (e.g. image name asc. / desc., image owner asc. / desc., newest / oldest image first etc.). With click on each image users will be rerouted to detailed image information, where also other important metadata of each image, including download and optimization process will be available.

Create (upload) new image page. This subpage will be available only to the registered and logged in users. Users, who will not be logged in will be redirected to the sign up (or sign in) page. Uploading or creating new image means introducing a new previously non-existing image into the ENTICE repository. VMI portal will offer to the users two ways for creating or uploading new VMIs or CIs. Either users will be able to upload already completed and configured images from a local storage, together will additional metadata, required for creating and representing detailed image information of the image to the ENTICE repository. The other way for creating new images will be to upload the generic image description with recipes, according to which ENTICE system will later on create a fully functional VMI or CI. Additional metadata consists of specifying image name, image type, initial Pareto SLA (P-SLA) requirements (e.g. cost, performance level, primary repository), where users will be able to choose manually repository for their image or the system will choose the most appropriate repository for them according to algorithm for Multi-objective optimization. Furthermore, the interface will provide means, for the management entities of the federation, to optimize the overall distribution of the VM Images across all repository sites.

IV. SYSTEM INTEGRATION

The integration process of any type of software product is a particularly critical point in the development stage. Improper integration can lead to reduced functionality of the product, and can induce costly delays and reworks. During the initial stages of software development one of the typical integration difficulties are the low maturity of the components, insufficiently defined interfaces, and inappropriately prepared environment. To properly implement the integration process and to overcome those constraints, efficient integration strategies should be implemented. Those strategies are usually performed in an iterative manner, thus allowing for the system to evolve in a control manner.

The above mentioned processes should always be synchronized to support the overall project strategy. Within the ENTICE environment the following essential components have been integrated in an efficient manner to guarantee efficient operation of the system:

- 1) VM Image distribution module
- 2) VM Image Synthesis and Analysis module
- 3) VM Image multi-objective optimization framework
- 4) Knowledge base and reasoning
- 5) Image portal

Due to the distributed nature of the ENTICE environment, service based APIs have been developed in order to guarantee efficient integration between the components. The interaction is performed by utilizing representational state transfer (REST) architectural style over secure HTTPS protocol. The VM image and fragment transfer is also performed over HTTPS protocol.

V. CONCLUSION

The most important part of the ENTICE environment are the developed new techniques for efficient deployment and management of VM images across federated Cloud infrastructures. The project itself encapsulates a interdisciplinary team of computer scientists and application providers working on a Federated Cloud technologies, thus provoking innovations in multiple different research fields. The ENTICE is a newly integrated environment capable of:

- 1) reducing the size of the VM images by proper analysis of its content,
- 2) optimizing the distribution of VM image across federated Cloud repositories,
- 3) dynamic redistribution of VM images, fragments and meta-data in accordance with the application needs,
- 4) utilizing an efficient knowledge and information management system for proper VM classification,
- 5) providing VM image portal which allows effortless interaction of the users with the federated infrastructure.

All this makes ENTICE an advanced environment for efficient deployment and management of VM images across federated Cloud infrastructures. Additionally, to

further streamline the integration of federated Cloud environments, to shield the users from the complexity of underlying Cloud technologies and to simplify the development and execution of complex use cases, ENTICE provides means for flexible tailoring and instantiating various VM images to multiple Cloud infrastructures.

This research area still poses multitude new challenges that urge additional efforts for the accomplishment of the ultimate goal, effortless management of VM images and associated meta-data in federated Cloud environments. In near future, we plan to introduce more complex reasoning and decision making mechanisms within the knowledge base model and to further improve the multi-objective optimization framework with support for automated creation of VM images in multiple different formats by utilizing virtual machine management templates.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 644179 (ENTICE project: dEcentralised repositories for traNsparent and efficienT vRtual maChine opErations).

REFERENCES

- [1] K. Razavi, T. Kielmann, "Scalable virtual machine deployment using vm image caches". In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC 2013, ACM, pp. 65:1–65:12.
- [2] Goiri, I., Guitart, J., Torres, J., "Characterizing cloud federation for enhancing providers' profit". 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 123-130.
- [3] P. C. Brebner, "Is your cloud elastic enough?: Performance modelling the elasticity of infrastructure as a service (iaas) cloud applications". In Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, ICPE 2012, ACM, pp. 263–266.
- [4] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation". In IEEE CLOUD 2010, pp. 337–345.
- [5] Durillo, J. J., Nebro, A. J., "jMetal: A Java framework for multi-objective optimization". Advances in Engineering Software, 2011, 42(10), pp. 760-771.
- [6] Branke, Jürgen, et al., "Multiobjective optimization: interactive and evolutionary approaches". Springer, 2008, Vol. 5252.
- [7] A. Kertesz, G. Kecskemeti, I. Brandic, "An interoperable and self-adaptive approach for sla-based service virtualization in heterogeneous cloud environments". Future Generation Computer Systems, 2014, pp. 54–68.
- [8] M. S. Kettouch, C. Luca, M. Hobbs and A. Fatima, "Data integration approach for semi-structured and structured data (Linked Data)". 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, 2015, pp. 820-825.
- [9] S. Abburi: *A Survey on Ontology Reasoners and Comparison*. International Journal of Computer Applications (0975 – 8887), Volume 57– No.17, November 2012.