

# Multi-objective Optimization Framework for VMI Distribution in Federated Cloud Repositories

Dragi Kimovski<sup>\*1</sup>, Nishant Saurabh<sup>1</sup>, Sandi Gec<sup>2</sup>, Vlado Stankovski<sup>2</sup>, and Radu Prodan<sup>1</sup>

<sup>1</sup> Distributed and Parallel Systems, Institute of Informatics, University of Innsbruck, Austria.

\*[dragi@dps.uibk.ac.at](mailto:dragi@dps.uibk.ac.at)

<sup>2</sup> University of Ljubljana, Faculty of Civil and Geodetic Engineering and Faculty of Computer and Information Science, Ljubljana, Slovenia

**Abstract.** Cloud Federation facilitates the concept of aggregation of multiple services administered by different providers, thus opening the possibility for the customers to profit from lower cost and better performance, while allowing for the cloud providers to offer more sophisticated services. Unfortunately, current state-of-the-art does not provide any substantial means for streamlined adaptation of federated Cloud environments. One of the essential barriers that prevents Cloud federation is the inefficient management of distributed storage repositories for Virtual Machine Images (VMI). In such environments, the VMIs are currently stored by Cloud providers in proprietary centralised repositories without considering application characteristics and their runtime requirements, causing high deployment and instantiation overheads. In this paper, a novel multi-objective optimization framework for VMI placement across distributed repositories in federated Cloud environment has been proposed. Based on the communication performance requirements, VMI use patterns, and structure of images or input data, the framework provides efficient means for transparent optimization of the distribution and placement of VMIs across distributed repositories to significantly lower their provisioning time for complex resource requests and for executing the user applications.

**Keywords:** Federated Cloud Environment, Distributed Storage Repositories, Multi-objective Optimization

## 1 Introduction

The rapid growth and development of Cloud computing platforms has brought high level of operational efficiency, thus provoking the appearance of multitude public cloud providers. Therefore, the increased availability of wide range of different providers has prompted the idea of federating Clouds infrastructures

[2]. The core aspect of Cloud federation can be considered the possibility for aggregation of complementary resources, which can be bundled together to allow boundless availability. The particular incentive for forming Cloud federations can be of different nature, such as application driven or community driven. In this sense, cloud federation can be viewed from the perspective of the Cloud providers or from the user's point of view. It may allow for the customers to profit from lower cost and better performance, and at the same time it can open the opportunity for the cloud providers to offer more sophisticated services [1]. Besides, this symbiosis can empower the formation of smart communities with decentralized infrastructures at the edge of the global network.

Unfortunately, current state-of-the-art does not provide any substantial means for streamlined adaptation of federated Cloud environments [5]. One of the essential barriers that prevents Cloud federation is the inefficient management of distributed storage repositories for Virtual Machine Images (VMI). In such environments, the VMI are currently stored by Cloud providers in proprietary centralised repositories without considering application characteristics and their runtime requirements, causing high deployment and instantiation overheads. Moreover, users are expected to manually manage the VMI storage, which is tedious, error-prone and time-consuming process, especially if working with multiple Cloud providers. Formerly, limited research has been conducted on the optimization of file distribution in relatively tightly coupled systems. Regrettably, those strategies are not suitable for federated Cloud environment.

In this paper, a novel multi-objective optimization framework for VMI placement across distributed repositories in federated Cloud environment has been proposed. Based on the communication performance requirements, VMI use patterns, and structure of images or location of input data, the framework provides efficient means for transparent optimization of the distribution and placement of VMI across distributed repositories to significantly lower their provisioning time for complex resource requests and for executing the user applications.

The optimization framework can be applied on two distinctive levels within a federated environments: *(i)* initial VMI distribution and *(ii)* offline VMI redistribution. Diverse heuristic tracks have been pursued for the implementation of the distinctive application levels of the framework, such as NSGA-II and other population based algorithms. Above all, a consolidated service based application program interface has been provided for easy integration of the framework within heterogeneous environments. The proposed framework has been developed by leveraging the jMetal Multi-objective optimization library and its behaviour has been evaluated in multiple different scenarios [3].

## 2 Background

In this section a brief overview of all concepts pertaining to this research work will be presented. Significant attention has been directed towards the basic concepts of multi-objective optimization and to the NSGA-II algorithm implemented in the proposed optimization framework.

## 2.1 Multi-objective Optimization

Optimization is a process of denoting one or multiple solutions that relate to the extreme values of multiple specific objective functions within given constraints. When the optimization task encompasses a single objective function it typically results in a single solution, called an optimal solution. Furthermore, the optimization also considers several conflicting objectives simultaneously. In such circumstances, the process will result in a set of alternative trade-off solutions, so-called Pareto solutions, or simply non-dominated solutions. The task of finding the optimal set of non-dominated solutions is known as multi-objective optimization [4].

A multi-objective optimization problem usually involves a number of objective functions which have to be minimized or maximized. In the most generic form, the problem can be formulated as:

$$\min(f_1(x), f_2(x), \dots, f_v(x)) \quad (1)$$

subject to  $x \in X$  where  $v \geq 2$  is the number of conflicting objectives functions  $f_i$  that we want to minimize, while  $X$  is a nonempty feasible region enclosing the set of variable (decision) vectors  $x = (x_1, x_2, \dots, x_n)$ .

The generic formulation of the multi-objective optimization is free from any constraints. However, this is hardly the case when real life optimization problems are being solved, which are typically constrained by some bounds. Constraints divide the search space into two distinctive regions: feasible and infeasible.

The multi-objective optimization consist of three distinctive phases: problem modeling, optimization and lastly decision making. Each of these phases is of paramount importance for attaining the the optimal set of feasible solutions.

## 2.2 Elitist Non-dominated Sorting Genetic Algorithm - NSGA-II

The Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) is an evolutionary multi-objective optimization procedure which attempt to find multiple Pareto-optimal solutions in a multi-objective optimization problem [6]. NSGA-II is characterised by the following three features: (i) it uses the principle of elitism, which dictates that the best solutions in the population should always be preserved and never deleted, (ii) it implements an explicit mechanism for diversity preserving in the population, (iii) and it emphasizes the non-dominated solutions on each iteration. Like with every genetic algorithm, the offspring population  $Op$  is created by using the parent population  $Pp$  and applying the proper crossover and mutation operators. Afterwards, the two populations are combined together to form  $Rt$ , which has double the initial population size. Only then a non-dominated sorting algorithm is applied to classify the full  $Rt$  population. Even though, this process induces higher computational costs, it allows for global non-dominated check to be performed both on parent and children populations. After the non-domination sort has been performed, the new population is created by adding solutions from different non-dominated fronts. The filling starts

with the best non-dominated front and continues with addition of solutions from the other fronts. It is important to note that since the overall population of  $Rt$  is  $2N$ , not all fronts will be accommodated in the new population. Lastly, when the final allowed front is considered, if the number of solutions is bigger than the available population slots, a strategy called crowding distance sorting is applied to select solutions from the least crowded region in the Pareto front.

### 3 Multi-objective Optimization Framework for VMI distribution

In this section a detailed description of the multi-objective optimization framework for VMI distribution in Federated Cloud repositories will be presented. The optimization framework has been applied on two distinctive application levels: (i) initial VMI distribution and (ii) offline VM image redistribution.

#### 3.1 Framework description

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 two 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. Afterwards, the “Offline VMI Redistribution” sub-module encapsulates the optimization of the VM images 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 images and available repository sites.

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 module has been realized as an ontology-based knowledge base [8]. 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 is required for proper documentation of the data transfers. The monitoring tool itself can be realized in multiple different manners, and it is dependent on the specifics of the Cloud infrastructure.

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. The high level structure of the optimization framework is presented on Figure 1.

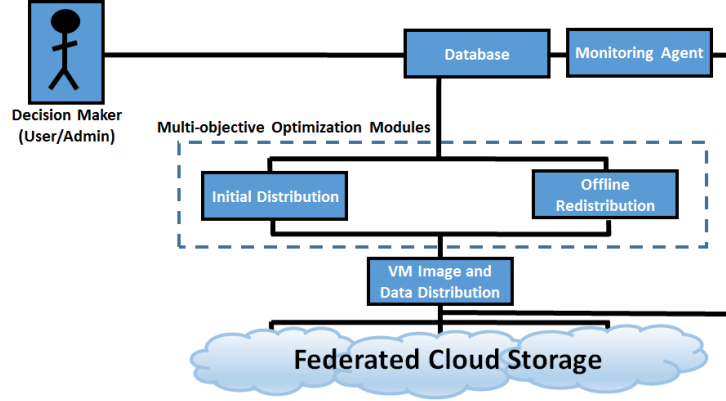


Fig. 1. Top level view of the Multi-objective Optimization Framework for VM Image distribution

### 3.2 Initial VM image upload

It is of paramount importance to properly store new VMIs and related data sets in federated Cloud repositories. In this section we introduce concepts from the field of Multiple-criteria decision making, to assist image providers and users to efficiently store new VMIs in accordance with their needs and repository characteristics. The described 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.

The problem of initial VM image upload consist of a finite number of combinatorial alternatives, which are explicitly known in the beginning of the solving process. In this case, each alternative solution represents one storage site in the federated repository, where the image or data-sets can be stored. Every solution is evaluated on the basis of two conflicting objectives. For the specific problem, the following objectives have been defined:

$$f(P) = B_r \quad (2) \quad f(C) = C_{st} + C_{tr} \quad (3)$$

where  $B_r$  represents the maximal theoretical performance of the interconnections of the repository, while  $C_{st}$  is the cost for storing data on the given repository and  $C_{tr}$  is the cost for transfer. Based on the given objectives, all possible storage sites in the repository, are then evaluated. It is important to be noted, that the evaluation is performed only on the feasible solutions, i.e only on the list of available repository sites. This means that prior to evaluation, all constraints for storing the VMI are taken into account. Afterwards, by introducing the concept of domination all evaluated solutions are sorted. The solutions which are non-dominated by any other solution are presented to the user in the form of Pareto front. In a sense, those solutions represent multiple optimal storage sites for

storing a single VM image within the federated repository. Next, the user, as a decision maker, can choose where to initially store its own images.

It also worth mentioning, that due to the static nature, this type of evaluation should only be performed when new storage sites have been added or removed from the federated repository. Afterwards, if there are no changes in the structure of the federated repository, the evaluation data can be used for selecting the appropriate storage site for every VM image that might be uploaded in future.

### 3.3 Offline VM image redistribution

Unlike the initial image upload, the problem of offline VMI redistribution consist of a finite, although 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. This process is performed by analyzing the repositories usage patterns, and results in optimized distribution of the VMIs and the associated data-sets across the federated environment. In what follows the exact sequence of steps of the offline VMI redistribution sub-module is presented.

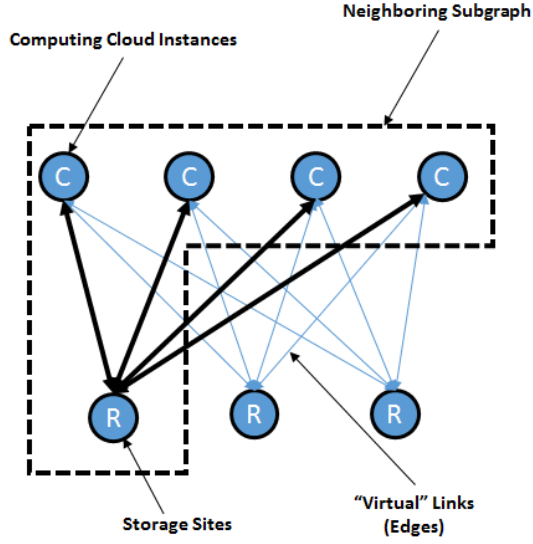
**Objective functions modeling** The cost model is described around the notion of the financial expenses which are needed to store a unit of data in a given repository site  $C_{st}$  and the economical burden for transferring the data from the initial to the optimal site  $C_{trnew}$ . The exact values of the financial expenses for data storage and transfers should be provisioned by all Cloud providers within the federation. For each VM image the cost objective can be calculated by using the formula below:

$$f(C) = C_{st} + C_{trnew} \quad (4)$$

The performance model includes much more complex reasoning behind it. It is based on the VM image usage patterns and it requires proper monitoring tool for efficient execution. The raw theoretical throughput of the interconnecting structure within a Cloud federation does not properly describe the factual communication performance, as it is difficult to predict the actual route the packets may take to reach the destination and the load on the intermediate communication channels. Opportunely, it is possible to leverage the data from the framework’s monitoring module to perform a coarse but sufficient estimation on the actual throughput between any pair of end points in the federation. In this way, if there is a sufficient information on the previous transfers among the repository sites and the Cloud computing instances, a direct “virtual” links between the above mentioned entities can be abstracted over the physical network and their bandwidth can be estimated.

Furthermore, it is possible to model an undirected weighted graph, where the vertices correspond to either a repository site or a computational Cloud instance and the edges of the graph are represented by the “virtual” links. The

weighted graph actually enclosed a union of multiple neighboring subgraphs, where each storage site vertex, as direct neighbor, is linked to all known computational cloud vertices. The weights of the edges in the graph are determined by leveraging the estimated average bandwidth  $B_{rc_i}$  on the corresponding “virtual” links. The weights are calculated dynamically, based the VMI distribution that is being considered. To properly model the weight of the edges, we introduce weight function, which considers the total number of downloads of the VMI to all neighbours  $G_{tv}$  and the number of downloads to particular Cloud neighbor  $G_i$ . The ratio of those two values is then multiplied with the estimated bandwidth of the particular “virtual” link to provide the final value of the edge’s weight. The structure of the neighbouring sub-graph has been represented on Figure 2.



**Fig. 2.** An example of a neighbouring sub-graph in a structure with 3 repository sites and 4 different cloud providers

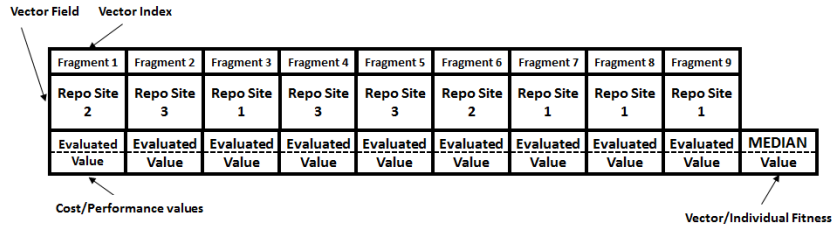
Subsequently, for modeling of the performance objective, the sum of the weights of the edges in the neighbouring subgraph is exploited, thus the performance can be described as:

$$f(P) = \sum_{i=1}^n B_{rc_i} \left( \frac{G_i}{G_{tv}} \right) \quad (5)$$

**Search Algorithm and Decision Making** The core of the offline VMI redistribution sub-module is constructed over the NSGA-II multi-objective optimization algorithm. As with any population based genetic heuristic the basic entity

is the individual. Within the given problem description the individual has been represented as vector with a size equal to the number of stored VMIs. The value kept in every element of the vector corresponds to a single storage repository where a particular VMI can be stored. For accomplishing the above statement, within the proposed framework, each VMI is assigned with a unique ID value, which correspond to the index of the vector element. Respectively, all storage sites in the federation are also assigned with unique IDs that are parallel to the appropriate values saved in the vector elements. In such way, each individual corresponds to a solution vector that represents unique global mapping of all VMIs to storage sites in the federated repository.

Afterwards, multiple solutions vectors are created and then randomly populated with values in the range from one to the number of available storage sites, thus creating the initial population. Every single individual represents one possible distribution solution that has to be evaluated. Then, the evaluation of each individual is performed by reading the values stored in the vector fields. Based on those values, starting from every element in the vector, a neighboring subgraph is constructed and the appropriate objective functions are applied. Those values are then grouped together and the median value is selected as the overall fitness of the given individual. An example of a single individual that correspond to a solution vector for mapping 9 VMIs to 3 storage repository sites in a given federation is presented on Figure 3. When all individuals in the initial



**Fig. 3.** An example individual represented as a solution vector

population have been successfully evaluated, the proper mutation and crossover operators are applied to create the children population. Then, the parents and children populations are grouped together and sorted according to dominance. Afterwards, only the best solution of the newly formed group are selected for the next iteration. This process is then repeated for a predefined number of iterations. The solutions which have been acquired after the last iteration are sorted based on the dominance. The non-dominated solutions are then presented to the administrative entity of the federation, which acts as a DM, and should select the most appropriate solution based on the pre-defined decision making policy.

Decision making on the alternatives discovered by the optimization algorithm requires an explicit model of the decision maker preferences. For the case of offline VMI redistribution the DM model will depend on the implementation of the



federated infrastructure. As the offline image redistribution envelops federation wide distribution of the VMIs we envision that the DM will be an administration entity, which will implement the federation storage policy based on the decision making model.

## 4 Experimental Evaluation

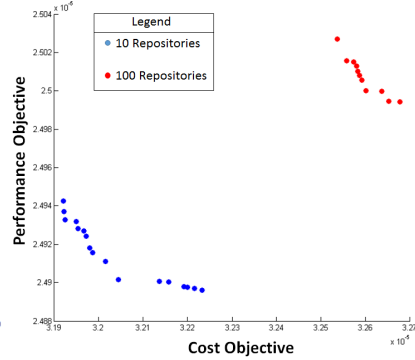
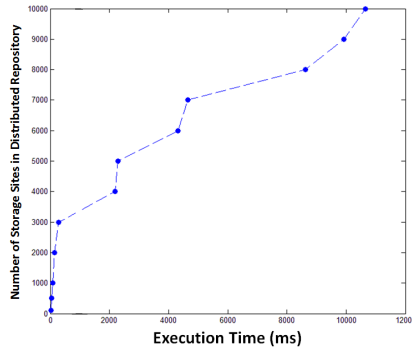
In this section, the proposed framework has been experimentally evaluated based on a synthetic set of benchmark data. As our research deals with the implementation of a combinatorial multi-objective problem in federated Cloud environment, we present an experimental results that demonstrate the ability of our approach to provide an adequate VMI distribution across federated repositories.

With respect to the different application levels of the multi-objective optimization framework, distinctive set of experiments were conducted. The initial VMI upload module has been evaluated on the basis of the degree of scalability, while the behaviour of the redistribution module has been examined from multiple aspects, such as accuracy, scalability and computational performance.

To begin with, the scalability and computational performance of the initial VMI upload module have been evaluated by varying the number of repository sites in the federation from 10 up to 10000 sites. Figure 4 shows the correlation between the average execution time and the number of storage sites in the federation. It is evident that the module can be lightly scaled up to large sizes. For relatively small federations the module can be invoked at each VMI upload, as it requires only few milliseconds to be executed.

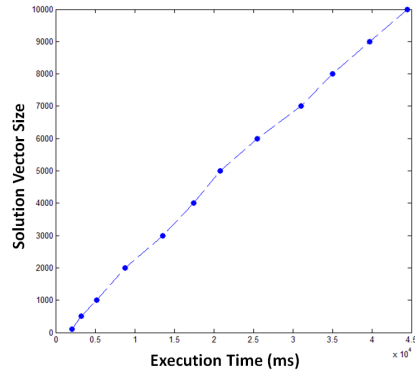
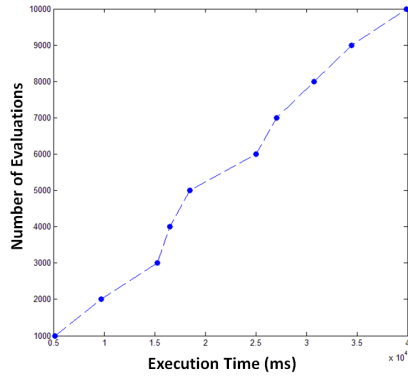
On the other hand, the VMI redistribution module encloses diverse operations that can affect its behavior to a various degree. Due to the nature of the algorithm it is not adequate to evaluate it's computational performance based on the number of repositories in the federation. Increasing the number of storage sites, influences on the number of possibilities where to store a single VMI image, which translates into reduced quality of the proposed solutions, but relatively constant execution time. For example, on Figure 5 a scenario in which the vector size (number of fragments) and number of evaluations have been kept constant, while the number of available repositories has been increased from 10(blue) to 100 (red), is presented. The Pareto fronts from both executions have been plotted together to show the difference in quality of the final solutions. The experimental scenario clearly shows that if we increase the number of storage sites, while maintaining constant number of evaluations, the quality of the solutions will decrease.

Furthermore, on Figure 6 and Figure 7, respectively, the influence that the number of evaluations and the size of the solution vector have on the computational performance is presented. In both cases, the number of associated cloud computing instances and storage sites were maintained constant; only the corresponding parameters were increased gradually. The presented results support the assumption of satisfactory scalability, both in a sense of increased number



**Fig. 4.** Execution time in comparison with the Number of storage sites in case of initial distribution  
**Fig. 5.** Comparison of two Pareto fronts during redistribution with varying storage sites

of stored VMIs and number of iterations needed to provide mapping solutions with good quality.



**Fig. 6.** Execution time in comparison with the number of evaluations during offline redistribution  
**Fig. 7.** Execution time in comparison with the size of solution vector during offline redistribution

Lastly, Tables 1 and 2 are providing a comprehensive review of the quality values for the trade-off mapping solutions calculated by the redistribution module. Moreover, a comparison has been presented with a set of mapping solutions determined by using "round robin" mapping model for storing VMIs in the federation. The statistical significance of the results has been analyzed by applying ANOVA test, which has shown significant difference between the proposed algorithm and the "round robin" mapping strategy, both in respect with the cost and performance objective. The cost objective has been calculated based

on the publicly provided price list for storing data in the Cloud by Amazon. The performance objective has been modelled based on the reported communication performance measures for 10Gbit and 1Gbit Ethernet [7]. For readability reasons, the bandwidth values, were converted to delivery time needed for 1Mbit of data to be transferred from the source to the destination.

**Table 1.** Comparison of the offline VMI redistribution module with "round robin" strategy for the performance objective (represented as required time to transfer 1Mbit of data).

Evaluations	Average Performance	STD (+/-)	Difference (%)	p-value
10000	0.00005356	0.00000272	18.48472759	>0.005
20000	0.00004732	0.00000287	34.10821694	>0.005
30000	0.00004109	0.00000316	54.42734792	>0.005
40000	0.00003793	0.00000263	67.29142067	>0.005
50000	0.00003526	0.00000314	79.98110620	>0.005
60000	0.00003281	0.00000259	93.39303541	>0.005

With respect to the parameters of the evolutionary algorithms, we have used a population of 1000 individuals, that iterates from 1 to 6 generations across populations. Every single individual(solution vector) is comprised of 1000 chromosomes, thus inducing mapping solutions for 1000 VMIs. Taking into account the results obtained in preliminary experiments, we have used simulated single point crossover with a crossover probability of 0.9, a mutation probability equal to  $1/n$  ( $n$  is the number of decision variables). The results indicate very high efficiency of the redistribution module, as it can provide better quality mapping solutions, especially in regards with the performance objective.

**Table 2.** Comparison of the offline VMI redistribution module with "round robin" strategy for the cost objective.

Evaluations	Average Cost	STD (+/-)	Difference (%)	p-value
10000	0.00003273	0.00000005	0.49133799	>0.005
20000	0.00003262	0.00000005	0.83933772	>0.005
30000	0.00003251	0.00000005	1.16811082	>0.005
40000	0.00003247	0.00000006	1.32053992	>0.005
50000	0.00003240	0.00000006	1.52613233	>0.005
60000	0.00003237	0.00000005	1.62638639	>0.005

## 5 Conclusion and future work

In this paper a novel approach for multi-objective optimization of the distribution of VMIs, as an essential storage resources, across distributed repositories in federated Cloud environment has been proposed. The research work has resulted

in development of a optimization framework that exploits multiple different factors, such as communication performance requirements, VMI use patterns, and structure of images, in order to optimize the distribution and placement of VMI across distributed repositories and to significantly lower their provisioning time for complex resource requests and for executing the user applications. The optimization framework has been evaluated based on synthetic simulation benchmark. As our research deals with the implementation of a combinatorial multi-objective problem, where the main incentive is to find the proper mapping of VMIs across storage sites, we present an experimental results that demonstrate the ability of our approach to provide an adequate VMI distribution across federated repositories.

There are multiple opportunities for future work in this research field. Novel heuristic algorithms can be implemented to further improve the performance and quality of the redistribution process. Furthermore, lightweight optimization algorithms can be utilized for performing time sensitive fine-grained optimization of the distribution of the VMIs and the associated data-sets during application execution.

## Acknowledgments

This work is being accomplished as a part of project *ENTICE: "decentralised repositories for transparent and efficient virtual machine operations"*, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644179.

## References

1. Goiri, I., Guitart, J., Torres, J. (2010, July). Characterizing cloud federation for enhancing providers' profit. In *Cloud Computing (CLOUD)*, 2010 IEEE 3rd International Conference on (pp. 123-130). IEEE.
2. Villegas, D., Bobroff, N., Rodero, I., Delgado, J., Liu, Y., Devarakonda, A., ... Parashar, M. (2012). Cloud federation in a layered service model. *Journal of Computer and System Sciences*, 78(5), 1330-1344.
3. Durillo, J. J., Nebro, A. J. (2011). jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10), 760-771.
4. Branke, Jürgen, et al., eds. *Multiobjective optimization: interactive and evolutionary approaches*. Vol. 5252. Springer, 2008.
5. Kurze, T., Klems, M., Bermbach, D., Lenk, A., Tai, S., Kunze, M. (2011). Cloud federation. *CLOUD COMPUTING*, 2011, 32-38.
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation*, IEEE Transactions on, 6(2), 182-197.
7. Feng, W. C., Balaji, P., Baron, C., Bhuyan, L. N., Panda, D. K. (2005, August). Performance characterization of a 10-Gigabit Ethernet TOE. In *High Performance Interconnects*, 2005. Proceedings. 13th Symposium on (pp. 58-63). IEEE.
8. Abburu, Sunitha. "A survey on ontology reasoners and comparison." *International Journal of Computer Applications* 57.17 (2012).