Experimenting with application-based benchmarks on different Cloud providers via a multi-cloud execution and modeling framework

A. Evangelinou[1], N. Galante[2], G. Kousiouris[1], G. Giammatteo[2] , E. Kevani[1], C. Stampoltas[1], A. Menychtas[1], A. Kopaneli[1], K. Ramasamy Balraj[2], D. Kyriazis[1], T. Varvarigou[1], P. Stuer[3], L. Orue-Echevarria Arrieta[4] , G. Echevarria Velez[4], Alexander Bergmayr[5]

[1]Dept. Of Electrical and Computer Engineering, NTUA, 9 Heroon Polytechnioy Str, 15773, Athens, Greece
[2]Research and Development Laboratory, Engineering Ingegneria Informatica S.p.A., V. R. Morandi, 32 00148 Roma, Italy
[3]Spikes Research dept., Spikes, Mechelsesteenweg 64, B-2018 Antwerpen, Belgium
[4]TECNALIA, ICT-European Software Institute Division, Parque Tecnológico Ed #202, E-48170 Zamudio, Spain
[5]*Business Informatics Group, TU Vienna*, Favoritenstraße 9-11/188-3
1040 Vienna, Austria,

aevang@mail.ntua.gr, NunzioAndrea.Galante@eng.it, gkousiou@mail.ntua.gr, gabriele.giammatteo@eng.it, eltonkevani@hotmail.com, stampoltaschris@gmail.com, ameny@mail.ntua.gr, alikikop@mail.ntua.gr, Kanchanna.RamasamyBalraj@eng.it, dimos@mail.ntua.gr, dora@telecom.ntua.gr, peter.stuer@spikes.be, Leire.Orue-Echevarria@tecnalia.com, Gorka.Echevarria@tecnalia.com, bergmayr@big.tuwien.ac.at

Abstract. Cloud services are emerging today as an innovative IT provisioning model, offering benefits over the traditional approach of provisioning infrastructure. However, the occurrence of multi-tenancy, virtualization and resource sharing issues raise certain difficulties in providing performance estimation during application design or deployment time. In order to assess the performance of cloud services and compare cloud offerings, cloud benchmarks are required. The aim of this paper is to present a mechanism and a benchmarking process for measuring the performance of various cloud service delivery models, while describing this information in a machine understandable format. The suggested framework is responsible for organizing the execution and may support multiple cloud providers. In our work context, benchmarking measurement results are demonstrated from three large commercial cloud providers, Amazon EC2, Microsoft Azure and Flexiant in order to assist with provisioning decisions for cloud users. Furthermore, we present approaches for measuring service performance with the usage of specialized metrics for ranking the services according to a weighted combination of cost, performance and workload.

Keywords: Benchmarking · Cloud services · Multi-Cloud · Performance ·Benchmarking

1 INTRODUCTION

Performance of cloud environments has started to gain significant attention in the recent years and is now related to more extensive concepts such as availability, competency and reliability (Hauck, 2010). After promises for infinite resources and on-demand scalability, the issues of cloud environments instability with regard to performance issues of the allocated resources have begun to arise (Kousiouris et al., 2012). Thus, for a successful cloud migration process, the issue of provider performance is a key factor that should be taken into account, in order to save money but also guarantee a (as much as possible) stability in the migrated application with respect to service-level agreements. In (Milenkoski et al., 2013), both the wide range of applications deployed in cloud environments and the variability of cloud services are detected; thus, measurement practices are required. Cloud benchmarks play a significant role in the wide-spread adoption of cloud services, providing end-to-end performance evaluation across different application areas and pricing of cloud offerings (Folkerts et al., 2013). As identified in our previous work (ARTIST Consortium D7.2, 2013), a significant gap in existing research is the lack of such descriptions in current metamodels regarding Cloud infrastructures. However, different providers may have their own metrics and strategies for guaranteeing cloud QoS. Thus, there is the need of a modeling framework in order to incorporate the related information and provide a more abstracted and common way for identifying performance aspects of cloud environments.

In our approach, the benchmarking process is based on the identification of a set of representative application types that correspond to various common applications that can be met in real life. For these types suitable benchmarks are identified that provide representative application scenarios and capture services abilities in the respective field. The main aspect of interest for the selection of the benchmarks was the ability to have application level workloads characterization. The major aim of this process is to abstract performance of cloud services to a suitable degree that can be understood and used by the majority of cloud non performance-aware individuals.

Given the extent of cloud computing environments, many factors can affect the performance of cloud services and their resources. The main performance aspects of cloud computing as analysed by the ARTIST approach can be summarized as:

a) Heterogeneous and unknown hardware resources: the computing resources offered by the cloud providers are unknown to the external users. Available information may be limited to number of cores for example, memory sizes or disk quotes. According to a study on Amazon platform conducted by Aalto University (Zhonghong Ou et al., 2012), the variation between the fast instances and slow instances can reach 40%. In some applications, the variation can even approach up to 60%.

b) Different configurations: even in the existence of the same hardware however, the way this resource is configured plays a significant role in its performance. The same applies for software configurations (e.g. a DB instance over a virtual cluster) or variations in the software development.

c) Multi-tenancy and obscure, black box management by providers: cloud infrastructures deal with multiple different users that may start their virtual resources on the same physical host at any given time. However, the effect of concurrently running VMs for example significantly degrades the actual application performance.

d) VM interference effects. Studies (Koh et al., 2007) (Kousiouris et al., 2011) show that combined performance varies substantially with different combinations of applications. Applications that rarely interfere with each other achieve performance to the standalone performance. However, some combinations interfere with each other in an adverse way.

e) Virtualization is a technology used in all cloud data centers to ensure high utilization of hardware resources and better manageability of VMs. According to the aforementioned studies despite the advantages provided by virtualization, they do not provide effective performance isolation. While the hypervisor (a.k.a. the virtual machine monitor) slices resources and allocates shares to different VMs, the behaviour of one VM can still affect the performance of another adversely due to the shared use of resources in the system. Furthermore, the isolation provided by virtualization limits the visibility of an application in a VM into the cause of performance anomalies that occur in a virtualized environment.

All these aspects along with the fact that cloud providers are separate entities and no information is available on their internal structure and operation, make it necessary to macroscopically examine a provider's behavior with regard to the offered resources and on a series of metrics. This process should be performed through benchmarking, by using the suitable tools and tests. One of the key aspects is that due to this dynamicity in resources management, the benchmarking process must be iterated over time, so that we can ensure as much as possible that different hardware, different management decisions (e.g., update/reconfiguration/improvement of the infrastructure) are demonstrated in the refreshed metric values, but also observe key characteristics such as performance variation, standard deviation etc. Finally, the acquired information should be represented in a machine understandable way, in order to be used in decision making systems.

In software engineering, metamodeling concepts are increasingly being used for representing a certain kind of information in a more abstracted level. During the last years, several proposals for cloud modeling concepts emerged supporting different scenarios, such as MODAClouds (Ardagna et al., 2012), which proposes a model-based migration approach similar to ARTIST. Nevertheless, the first one focusses on the migration of cloud-based software between cloud providers and their interoperability, while ARTIST on the migration of software artefact to cloud-based software as a means of software modernization. With regard to our work, following the metamodel definition, concrete instances for specific cloud providers and services can be created in order to describe the target environments of the migrated applications. Thus, during the deployment phase, the provider that fits best to the application type being migrated will be selected.

The aim of this paper is to provide such mechanisms to address the aforementioned issues. A benchmarking framework designed in the context of the FP7 ARTIST project is presented in order to measure the ability of various cloud offerings to a wide range of applications, from graphics and databases to web serving and streaming. The framework consists of a software suite for benchmarking cloud platforms in order to extract performance-related data and to include it in the cloud models. What is more, we define a metric, namely Service Efficiency (SE), in order to rank different services based on a combination of performance, cost and workload factors. YCSB and DaCapo are the two benchmarks used in the performance testing. The first one is a framework that facilitates comparisons of different NoSQL databases, while DaCapo benchmarking suite measures JVM related aspects. Measurement results are demonstrated after the implementation of Service Efficiency on various known cloud environments.

The paper is structured as follows: In Chapter 2, an analysis of existing work is performed; in Chapter 3, the description of the ARTIST tools for mitigating these issues is presented; in Chapter 4, a case study on three cloud commercial providers Amazon EC2, Microsoft Azure and Flexiant is presented; conclusions and future work is contained in Chapter 5.

2 RELATED WORK

Related work around this paper ranges in the fields of performance frameworks, available benchmark services and description frameworks and is based in the according analysis performed in the context of the ARTIST project (ARTIST Consortium D7.2, 2013). With regard to the former, the most relevant to our work is (Garg, 2012). In this paper, a very interesting and multi-level cloud service comparison framework is presented, including aspects such as agility, availability, accountability, performance, security and cost. Also an analytical hierarchical process is described in order to achieve the optimal tradeoff between the parameters. While more advanced in the area of the combined metric investigation, this work does not seem to include also the mechanism to launch and perform the measurements. Skymark (Iosup, 2012) is a framework designed to analyze the performance of IaaS environments. The framework consists of 2 components – Grenchmark and C-Meter. Grenchmark is responsible for workload generation and submission while C-Meter consists of a job scheduler and submits the job to a cloud manager that manages various IaaS Clouds in a pluggable architecture. Skymark focuses on the low level performance parameters of cloud services like CPU, Memory etc. and not on elementary application types.

CloudCmp (Li, 2010) provides a methodology and has a goal very similar to our approach to estimate the performance and costs of a Cloud deployed legacy application. A potential cloud customer can use the results to compare different providers and decide whether it should migrate to the cloud and which cloud provider is best suited for their applications. CloudCmp identifies a set of performance metrics relevant to application performance and cost, develop a benchmarking task for each metric, run the tasks on different providers and compare. However CloudCmp does not seem to define a common framework for all the benchmark tasks.

With regard to benchmarking services, the most prominent are CloudHarmony.com and CloudSleuth.com. The former utilizes a vast number of benchmarks against various cloud services, offering their results through an API.

However, there are two aspects that can be improved with relation to this approach. Initially it is the fact that too many benchmarks are included in the list. We believe that a more limited scope should be pursued in order to increase the focus of the measurements. Furthermore, the measurement process is not repeated on a regular basis, in order to investigate aspects such as deviation. For CloudSleuth, the focus is solely on web-based applications and their response time/availability. Their approach is very worthwhile, by deploying an elementary web application across different providers and monitoring it constantly, however it is limited to that application type.

With regard to description frameworks, a number of interesting approaches exist. According to the REMICS project (REMICS Consortium Deliverable D4.1 v2.0, 2012) PIM4Cloud, which is focused in both private and public Clouds, has been defined to provide support to model the applications and also to describe the system deployment on the cloud environment. PIM4Cloud is implemented as a profile for UML and a metamodel which is capable to describe most of the features of a system that will be deployed in a Cloud environment. It is organized in four packages (Cloud Provider domain, Cloud Application domain, Physical Infrastructure domain and Resource domain).

FleRD (Schaffrath et al., 2012) is a flexible resource description language for inter-provider communication in virtual networks architectures. It appears enhanced with regard to realism and generality (ability to describe real world topologies), extensibility, grouping and aggregation. FleRD is mainly focused around networking elements, however its concepts of modeling more information for QoS of networks has influenced our approach.

EDML (Charlton, 2009) defines a XML syntax for declaring internal and external general parsed entities. VXDL(Koslovski et al, 2008) is an XML-based language that describes Virtual Private eXecution Infrastructure (ViPXi) which is a time-limited organized aggregation of heterogeneous computing and communication resources. VXDL can describe resources, networks' topology that are virtual but are also, to some extent, adapted to physical ones and finally to represent timeline.

The implementation of DADL (Mirkovic et al., 2010) is based on the prediction that future businesses will use allocated resources from different Clouds such as public or private to run a single application. DADL was developed as an extension of SmartFrog (framework for deploying and managing distributed software systems based on java) and it is used to specify application architecture and cloud resources that are necessary for an application to run. There are elements to describe QoS features such as CPU speed, number of cores etc.

The main issue with the aforementioned approaches, which most of them support description of QoS terms, is the fact that in many cases the standard ways (CPU cores, frequency etc.) of describing capabilities are not sufficient to demonstrate the actual performance of virtualized resources. Thus a new approach based on benchmark scores should be pursued that would indicate the direct capability of a resource service to solve a particular computational problem. The descriptions defined in this paper are revolved around this test-based approach.

3 BENCHMARKING APPROACH IN ARTIST

The benchmarking approach followed in the context of ARTIST has the following targets:

- Identify a set of common application types and the respective benchmarks to measure the performance of cloud services
- Create a framework that is able to automatically install, execute and retrieve the benchmark results, with the ability to support multiple providers
- Investigate aspects of cloud service performance with regard to variation and ability across a wide range of potential applications
- Define a machine understandable way of representing this information and improved metrics that will characterize more efficiently the services.

The use case diagram for the benchmarking suite appears in Figure 1. We envisage that the capabilities of the toolkit will be exploited by an entity ("Benchmarks Provider") that will be responsible for performing and obtaining the tests, similar to the role of CloudHarmony.com. This entity will utilize the various aspects of the toolkit in order to create provider models that have concrete results and metrics per service offering, that are stored on the ARTIST repository, so that an external entity ("Model User") may be able to retrieve and consult them. More details on each part of the process are presented in the following paragraphs.

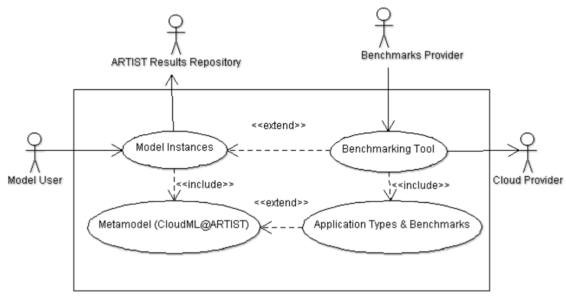

Figure 1. Use case diagram for the ARTIST Benchmarking process and tools

3.1 Application Benchmark types

The main target of the application benchmark types is to highlight a set of common and popular application tests that can be used in order to benchmark provider's offerings. Thus each offering may have a performance vector indicating its ability to solve specific problems or cater for a specific type of computation. The set of application benchmarks used in ARTIST appears in Table 1.

Table 1. Benchmark Tests used in the ARTIST platform

Benchmark Test	Application Type
YCSB	Databases
Dwarfs	Generic applications
Cloudsuite	Common web aps like streaming, web serving etc.
Filebench	File system and storage
DaCapo	JVM applications

3.2 Models of necessary information

In order to exploit the information from the benchmark execution, a suitable machine understandable format should be in place in order to store results and utilize them in other processes like service selection. For achieving this, suitable model templates have been designed. These templates include all the relevant information needed, such as measurement aspects (number of measurements, statistical information like standard deviation etc.), test configurations and workload profiles.

Initially these templates had been defined as an XML schema and in later stages they were incorporated into a suitable sub-metamodel developed in the context of the ARTIST project (CloudML@ARTIST metamodel). In this sub-metamodel, which is portrayed in (Figure 2), the workloads are static (and the default ones defined by each benchmark) in order to be able to compare the performance of different services on the same examined workload. For simplicity purposes we have defined one universal enumeration that includes the default workloads from all the aforementioned categories. Also for the Cloudsuite case, each category reports a large number of statistics, that are case specific. In order to simplify the descriptions, we have kept only the generic average score to be included in the model instances. However the remaining information will be kept during benchmark execution in a raw data DB, in order to be used in case an interested party needs the entire range of information.

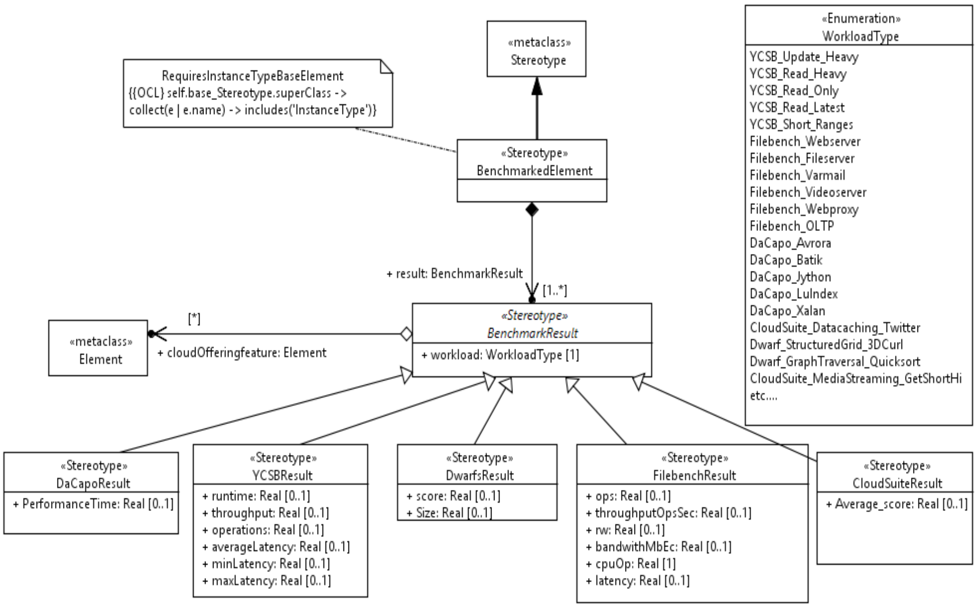

Figure 2. Performance metamodel for incorporation of application stereotypes information in CloudML@ARTIST

3.3 Benchmarking Suite Architecture

The Benchmarking Suite Architecture appears in Figure 3. The user through the GUI (Figure 4) may set the conditions of the test, selecting the relevant benchmark, work-load conditions, target provider and service offering. Furthermore, through extended interfaces, the results of the benchmarking process may be viewed and categorized based on different types of workload, VM and test types.

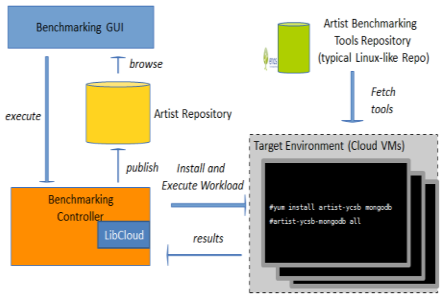

Figure 3. Overall System Architecture

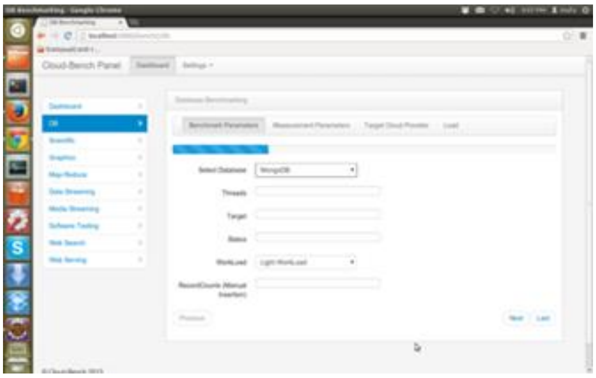

Figure 4. GUI for selecting and configuring tests on provider offerings

This process is performed in two stages. Initially the user enters the necessary information for data retrieval (Figure 5), including information on the test dates, SE metric configuration (for more info on the SE metric check 3.4), instance type, test type and workload. The results are displayed via a third interface (Figure 6) that contains the average score for the selected test, along with the SE metric result. Furthermore, the raw data lines from which the scores are extracted are returned.

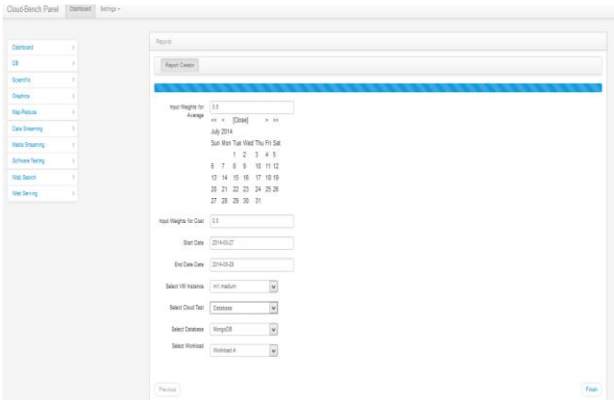

Figure 5. Filtering information for historical data

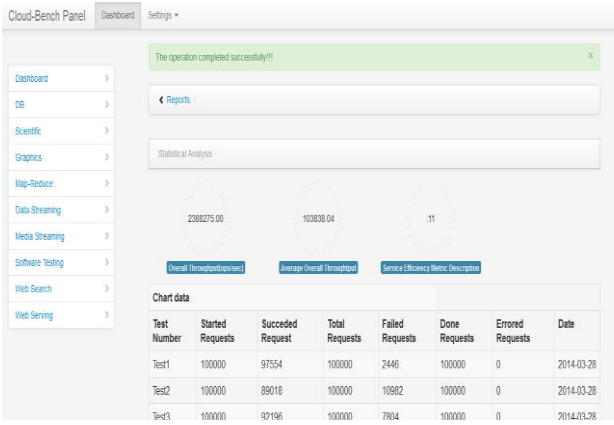

Figure 6. Result display and statistical information

This information is passed to the Benchmarking Controller which is responsible for raising the virtual resources on the target provider and executing the tests. The former is based on the incorporation of Apache LibCloud project, in order to support multiple provider frameworks. The latter needs to install first the tests, through the utilization of an external Linux-like repository that contains the scripts related to the different phases of installation and execution of each test. Once the tests are installed (through a standard repo-based installation), the workload setup scripts are transferred to the target machines and the execution begins.

The configuration of initial tests was modified in order to achieve some extra desirable figures like the number of iterations and extended run time. After the execution results are transferred back and processed in order to be included in the model descriptions, following the template definition highlighted in Section 3.2. In addition, a mysql raw database schema created and provided in case the results from the benchmark tests needed to be stored locally. The database structure is portrayed in Figure 7.

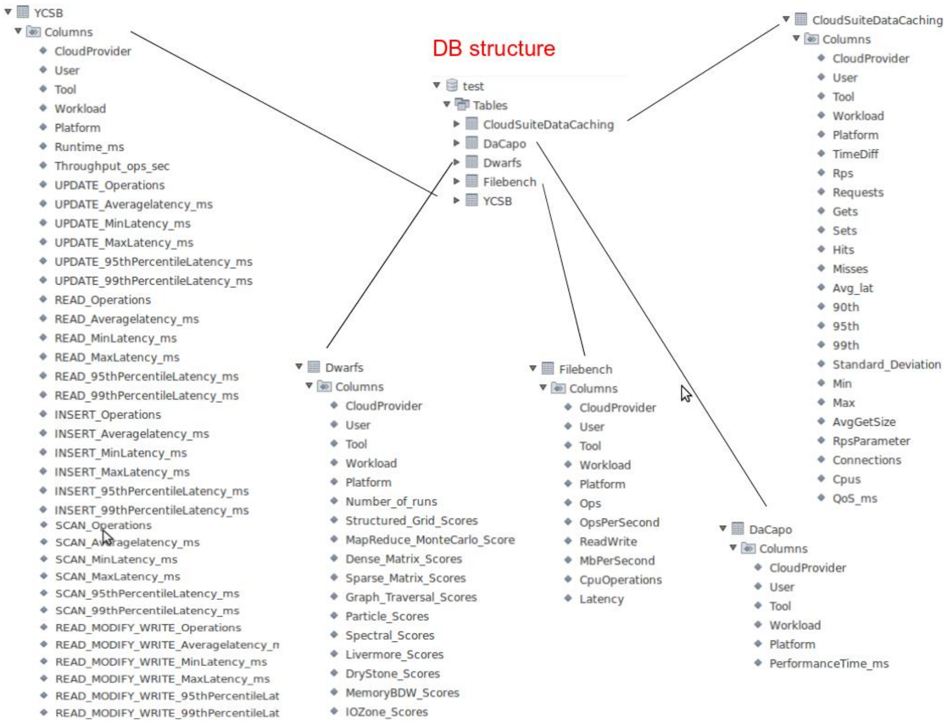

Figure 7. Raw Database structure

At the moment the two parts (GUI and Controller) are not integrated, however the benchmarking controller which is the main backend component can also be used standalone to perform measurements on the various application types. Installation instructions can be found in (ARTIST Consortium D7.2.1, 2013).

3.4 Service Efficiency Metric Description

In order to better express the performance ability of a service offering, we considered the usage of a metric that would fulfill the following requirements:

- Include workload aspects of a specific test
- Include cost aspects of the selected offering
- Include performance aspects for a given workload
- Give the ability to have varying rankings based on user interests
- Intuitively higher values would be better

Following these points, we considered that positive factors (e.g. workload aspects) should be used in the numerator and negative on the denominator (cost and Key Performance Indicators that follow a “higher is worse” approach). Furthermore, normalization should be used in order to have a more generic rating. Usage of sum would be preferable over product, since the former enables us to include weight factors. Thus such a metric can answer a potential question of the sort: “what is the best service to run my web streaming application, when I am interested more on a cheap service?”.

The resulting formula of Service Efficiency is the following (Equation 1):

$$SE = \frac{\sum_i s_i l_i}{\sum_j s_j w_j f_j} \quad (1)$$

Where s scaling factor
l: workload metric
f: KPI or cost metric
w: weight factor

The resulting metric can be compared between different offerings, potentially of different providers but on the same workload basis. However the incorporation of workload is necessary since it affects the performance and thus ranking of the services. Usage of different workloads may display a different optimal selection, based on the anticipated workload conditions for the application that needs to be deployed.

4 METRIC CASE STUDY ON THREE SELECTED CLOUD PROVIDERS: AMAZON EC2, MICROSOFT AZURE AND FLEXIANT

In order to experiment initially with the defined metrics and investigate differences in VM performance, we utilized workloads from both DaCapo benchmarking suite and YCSB benchmark framework. However the Benchmarking Controller apart from DaCapo and YCSB supports the managing execution of three more benchmarks included in Table 1, such as Dwarfs, CloudSuite and Filebench. Nevertheless, in this work, only the aforementioned ones have been tested.

DaCapo is designed to facilitate performance analysis of Java Virtual Machines, while YCSB measures databases performance. The selected workloads from each test were running on instances in three different cloud environments: Amazon EC2, Microsoft Azure and Flexiant. Regarding Amazon EC2, different types of VM instances were selected while for Microsoft Azure and Flexiant the tests were running on the same VM instances during the entire benchmarking process. Information regarding the selected benchmarking workloads and the VM instance characteristics are presented in

Table 2 and Table 3 respectively.

Table 2. DaCapo and YCSB application benchmark type

DaCapo	YCSB
xalan: transforms XML documents into HTML ones	**A:** Update heavy workload
tomcat: runs a set of queries against a tomcat server retrieving and verifying the resulting webpages	**B: Read mostly workload**
pmd: analyzes a set of Java classes for a range of source code problems	**C:** Read only
jython: interprets pybench Python benchmark	**D:** Read latest workload
h2: executes a JDBC benchmark using a number of transactions against a banking model application	**E:** Short ranges
fop: parses/formats XSL-FO file and generates a PDF file	**F:** Read-modify-write
eclipse: executes jdt performance tests for the Eclipse IDE	
avrora: simulates a number of programs running on a grid of AVR micro-controllers	

Table 3. VM instance characteristics

Cloud Provider	VM instance	Region
Amazon EC2	t1.micro	N.Virginia
	m1.medium	N.Virginia
	m1.large	N.Virginia
Microsoft Azure	small Standard	Ireland
Flexiant	4GB RAM- 3CPU	Ireland

The execution of the tests took place at specific hours (daily and at different time intervals) during a period of two weeks and the average values were extracted for each case. Moreover, the different time zones of the three respective regions were taken into consideration so that the peak hours were the same in each zone. Then the metric SE described in Section 3.4 was applied with the following form:

$$\mathrm{SE} = \frac{1}{w1*delay+w2*Cost}$$

Different weights were given to the performance and cost aspects (50-50, 90-10) and a normalization interval was considered (1-10). We avoided using a normalization interval including 0 since it may lead to infinite values for some cases. One should compare between same color bars, indicating similar workloads. From the graphs it is evident how the ranking of a potential VM rating can change based on the aspect that we are most interested in.

For example, both the weighted decision (50-50) and the performance-based selection (90-10) in the DaCapo case for ‘workload fop’ suggest Azure as the best choice (Figure 8). However the overall results may not be favorable for Flexiant due to the fact that we only measured large VM instances. Another interesting result derives from the fact that since the Service Efficiency metric includes the cost factor, we conclude that the smaller VMs give better results and the selection of small instances are more efficient. Moreover, it is worthy to mention that for the DaCapo case the Service Efficiency values for the large VM instances which were tested, are lower than the values for small or medium VM instances. Despite the fact that the performance of a large VM instance is approximately double than the performance of a small or medium instance, the cost is significantly higher, approximately 3.5 times (**Error! Reference source not found.**). Thus, taking into consideration that the Service Efficiency is inversely proportional to cost, this is expected.

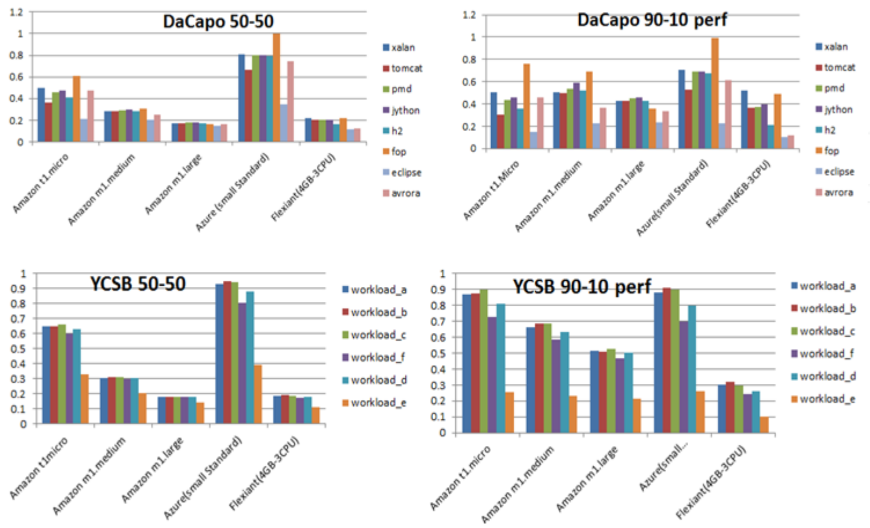

Figure 8. Implementation of the metric on application-based benchmarks across different cloud providers and different type of VM instances and variable workload (higher score is better). Comparison should be made between columns with similar color (identical workload conditions)

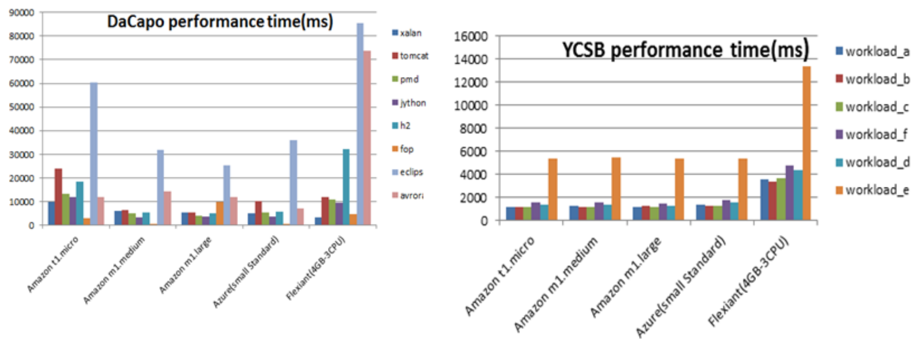

Figure 9 Performance time in ms for DaCapo and YCSB workloads

Moreover, the performance for the given workloads is similar across the Amazon and Azure instances. This is probably due to the fact that the maximum computational threshold of the VM was not reached. For Flexiant the performance is significantly lower and this behaviour seems to be related to a configuration of the VM in the Flexiant environment which was outside of our control.

In addition, for all the tested VM instances the performance for the "Short Ranges" workload, ‘workload_e’, is approximately three times lower than the other workloads. Thus, independently from the VM size (small, medium or large) the ‘workload_e’ seems to be three times slower than other workloads which were tested.

5 Conclusions

The emergence of cloud computing has led to a plethora of various offerings by multiple providers, covering different needs of the consumer. However significant questions emerge for the efficiency of these pay-per-use resources, mainly with regard to various application types, in this highly dynamic management environment. In this paper a multi-Cloud measurement framework has been presented, that has the aim of investigating these performance issues of the virtualized offerings. The framework utilizes a variety of application related benchmarks in order to cover a wide range of application types and it mainly focuses on investigating aspects such as service efficiency with relation to cost.

A combined metric (Service Efficiency) is also proposed in order to combine workload, performance and cost aspects in a single rating for comparing cloud offerings across different providers. A case study on 3 cloud providers has indicated the application of such a metric to characterize the offerings based on this combination.

For the future, we intend to complete the integration of the framework (currently missing the integration between GUI and Benchmark Suite Controller). However the Benchmarking Controller for the execution of the tests can be used also as standalone, following the instructions in (ARTIST Consortium D7.2.1, 2013). Another interesting aspect would also be the incorporation of other non-functional aspects such as availability in the main SE metric.

Finally, the implementation of the measurements in more instance types and for all the benchmarks defined in Section 3.1 is one of our major goals for the future.

Acknowledgements

The research leading to these results is partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 317859, in the context of the ARTIST Project.

References

1. ARTIST Consortium (2013), Deliverable D7.2 v1.0- PaaS/IaaS Metamodelling Requirements and SOTA, Available at: http://www.artist-project.eu/sites/default/files/D7.2%20PaaS%20IaaS%20metamodeling%20requirements%20and%20SOTA_M4_31012013.pdf
2. REMICS Consortium (2012), Deliverable D4.1 v2.0 - PIM4Cloud, Available at: http://www.remics.eu/system/files/REMICS_D4.1_V2.0_LowResolution.pdf
3. George Kousiouris, Dimosthenis Kyriazis, Andreas Menychtas and Theodora Varvarigou, "Legacy Applications on the Cloud: Challenges and enablers focusing on application performance analysis and providers characteristics", in Proceedings of the 2012 2nd IEEE International Conference on Cloud Computing and Intelligence Systems (IEEE CCIS 2012), Oct. 30th ~ Nov. 1st,Hangzhou, China
4. Hauck, M., Huber, M., Klems, M., Kounev, S., Muller-Quade, J., Pretschner, A., Reussner, R., and Tai, S. Challenges and opportunities of Cloud computing. Karlsruhe Reports in Informatics 19, Karlsruhe Institute of Technology - Faculty of Informatics, 2010
5. Zhonghong Ou, Hao Zhuang, Jukka K. Nurminen, Antti Ylä-Jääski, and Pan Hui. 2012. Exploiting hardware heterogeneity within the same instance type of Amazon EC2. In Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing (HotCloud'12). USENIX Association, Berkeley, CA, USA, 4-4.
6. George Kousiouris, Tommaso Cucinotta, Theodora Varvarigou, "The Effects of Scheduling, Workload Type and Consolidation Scenarios on Virtual Machine Performance and their Prediction through Optimized Artificial Neural Networks , The Journal of Systems and Software (2011),Volume 84, Issue 8, August 2011, pp. 1270-1291, Elsevier, doi:10.1016/j.jss.2011.04.013.
7. Y. Koh, R. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu. An analysis of performance interference effects in virtual environments. In IEEE In-

ternational Symposium on Performance Analysis of Systems and Software (ISPASS), pages 200–209, April 2007.

8. Schaffrath, Gregor and Schmid, Stefan and Vaishnavi, Ishan and Khan, Ashiq AND Feldmann, Anja, 2012. A Resource Description Language with Vagueness Support for Multi-Provider Cloud Networks. Munich, Germany: International Conference on Computer Communication Networks (ICCCN '12).
9. Charlton, S. (2009). Model driven design and operations for the Cloud. In OOPSLA09, 14th conference companion on Object Oriented Programming Systems Languages and Applications, pages 17–26
10. Koslovski, G., Primet, P.V.-B., Charao, A.S.: VXDL: Virtual Resources and Interconnection Networks Description Language. In: GridNets 2008 (October 2008).
11. Mirkovic, J., Faber, T., Hsieh, P., Malayandisamu, G., Malavia, R.: DADL: Distributed Application Description Language. USC/ISI Technical Report ISI-TR-664 (2010).
12. Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya, A framework for ranking of Cloud computing services, Future Generation Computer Systems, Volume 29, Issue 4, June 2013, Pages 1012-1023, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2012.06.006.
13. A. Iosup, R. Prodan, and D. Epema, "Iaas Cloud Benchmarking: Approaches, Challenges, and Experience," Proc. of the Int'l Conf. on High Performance Networking and Computing (SC), MTAGS 2012 . IEEE/ACM, pp. 1–8.
14. Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. 2010. CloudCmp: comparing public Cloud providers. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10). ACM, New York, NY, USA, 1-14. DOI=10.1145/1879141.1879143 http://doi.acm.org/10.1145/1879141.1879143
15. ARTIST Consortium (2013), Deliverable D7.2.1 v1.0- Cloud services modelling and performance analysis framework, Available at: http://www.artist-project.eu/sites/default/files/D7.2.1%20Cloud%20services%20modeling%20and%20performance%20analysis%20framework_M12_30092013.pdf
16. Folkerts, E., Alexandrov, A., Sachs, K., Iosup, A., Markl, V., & Tosun, C. (2013). Benchmarking in the cloud: What it should, can, and cannot be. In Selected Topics in Performance Evaluation and Benchmarking (pp. 173-188). Springer Berlin Heidelberg.
17. Aleksandar Milenkoski, Alexandru Iosup, Samuel Kounev, Kai Sachs, Piotr Rygielski, Jason Ding, Walfredo Cirne, and Florian Rosenberg. Cloud Usage Patterns: A Formalism for Description of Cloud Usage Scenarios. Technical Report SPEC-RG-2013-001 v.1.0.1, SPEC Research Group - Cloud Working Group, Standard Performance Evaluation Corporation (SPEC), April 2013
18. D. Ardagna, E. Di Nitto, G. Casale, D. Petcu, P. Mohagheghi, S. Mosser, P. Matthews, A. Gericke, C. Balligny, F. D'Andria, C.-S. Nechifor, and C.

Sheridan, “MODACLOUDS, A Model-Driven Approach for the Design and Execution of Applications on Multiple Clouds,” in ICSE MiSE: International Workshop on Modelling in Software Engineering. IEEE/ACM, 2012, pp. 50–56.