Proceedings of IEEE CCIS2012

Legacy Applications on the Cloud: Challenges and enablers focusing on application performance analysis and providers characteristics

George Kousiouris, Dimosthenis Kyriazis, Andreas Menychtas and Theodora Varvarigou

National Technical University of
Athens
9 Heroon Polytechniou str,
157 73 Zografou, Athens
{gkousiou,dimos,ameny}@mail.ntua.gr, dora@telecom.ntua.gr

Abstract: The advent of Cloud computing led to new ways for developing, engineering, providing and consuming services. As a paradigm building on a set of combined technologies, clouds enable on-demand service provisioning with guaranteed levels of quality on virtualized resources across disparate administrative domains. The latter has been one of the main factors for the wide adoption of clouds. Nevertheless, one class of applications has not yet taken advantage of the added-value of clouds: legacy applications. In this paper we highlight the challenges that arise when porting legacy applications on cloud environments and propose enabling technologies for overcoming these challenges and the corresponding limitations. We emphasize on application analysis and aspects oriented to quality of service, multi-tenancy contexts that affect the application behaviour at runtime, as well as characterization of providers in order to support decision making regarding the optimum ones to host the aforementioned legacy applications following their analysis.

Keywords: Cloud Computing, Legacy Applications, Performance, Multi-tenancy, Application Profiling

1 Introduction

Cloud computing is rapidly evolving and becoming the most widely adopted computing paradigm in Information and Communication Technology (ICT) for the rapid provision of resources and services. Its usefulness for users and enterprises in general is clearly recognized [1], with end users, startups and (in general) companies offering services that overcome the need to invest upfront in dedicated infrastructure resources - Animoto being a quite representative successful example [2].

While initially discussed within the telecommunications infrastructures domain for dynamic traffic switching to balance utilization [3], since 2001 it refers to an infrastructure of computers (as discussed in a presentation about the .NET framework [5]. Nevertheless, what really boosted cloud computing and enabled its wide adoption was Amazon's publication of the Elastic Compute Cloud EC2 [6]. Nowadays, cloud computing refers to a computing paradigm whose foundation is the delivery of services and ICT assets [7], often denoted as XaaS (Everything as a Service). As the cloud-related technologies mature and become ubiquitous, standardization efforts led to the definition of the Software-Platform-Infrastructure (SPI) service model [8], identifying different areas for service provision: Software (SaaS), Platform (PaaS) and Infrastructure (IaaS) as a service.

Various characteristics, such as virtualization of hardware, rapid service provisioning, scalability, elasticity, accounting granularity and cost allocation models enable clouds to efficiently adapt resource provisioning to the dynamic demands of Internet users. With respect to applications, service-based ones have been amongst the first being deployed in cloud environments. Nevertheless, a specific class of applications, namely legacy systems, has not yet found its way to the cloud. Legacy systems refer to software products that have been developed following the software development lifecycle, from requirements to engineering, development and integration. Usually, legacy systems and applications are developed according to identified and predefined specifications and with a priori knowledge of different application characteristics, which range from the infrastructure on which the applications will be executed and the corresponding network topology to the number of users, the application containers, etc. There are various technical (e.g. reliability) and non-technical (e.g. frequent updates) reasons for not migrating legacy applications on clouds.

Moreover and given that in many cases, solutions to legacy systems may even lead to re-development of them [9], what is needed refers to mechanisms and approaches that will enable the migration of legacy applications to cloud environments. The latter highlights a number of research challenges that need to addressed. In this paper we present these challenges as well as enabling technologies with respect to: (a) the description and performance prediction of newly created services after the legacy application transformation as well as the selection of the fittest IaaS provider for the specific application, (b) the usage of services in a multi-tenancy context in order to optimize legacy application execution, (c) the decoupling of the legacy application

transformation to the SaaS paradigm from specific PaaS / IaaS vendors, and (d) the characterization of a service provider based on functional and non-functional attributes.
The remainder of the paper is structured as follows. Section 2 presents related work in the field of porting legacy applications on the cloud with respect to performance analysis of these applications, while the following sections introduce the challenges (Section 3) and enablers (Section 4) for efficient execution of legacy applications on cloud environments. Finally, Section 5 concludes with a discussion on future research and potentials for the current study.

2 Related Work

The authors in [24] discuss the issue of hybrid deployment of a legacy application, partly in the Cloud and partly on premises, based on the needs for security and performance. A detailed analysis is conducted based on the performance benefits and overheads that are inserted by the hybrid deployment in various scenarios and use cases. While very interesting, this work does not consider the issue of choosing the fittest provider among different candidates based on their internal configuration of the infrastructures.

In [23], the authors discuss the benefits of application multi-tenancy in the field of Cloud Computing along with an architecture for efficiently creating SOA-based platforms that incorporate the multiple tenants feature. CloudMig [25] has a very interesting concept for migrating legacy applications including target infrastructure rankings and matchmaking. Application profiling and categorization however seems to not be part of the overall system.

REMICS [26] has focused on a tool-supported model-driven methodology for migrating legacy applications to interoperable service cloud platforms. The first step isto understand the legacy system in terms of its architecture and functions, and then transform its design to a new SOA application that provides the same or better functionality, and verifying and implementing the new application in the cloud. This work covers the transformation part of a monolithic application to a component based one, however performance in the newly formed application is not investigated. Similar to REMICS, OSCAR [27] discusses the transformation from an old fashion single component application to a web based, component like architecture.

An interesting analysis on how applications may be mapped on cloud platforms and whether all the parts should be migrated is contained in [28].

3 Challenges

This section describes in detail the challenges that may be encountered during the migration of an application on Cloud platforms. While this analysis is mainly performed in the context of legacy applications, it may be extended in a more generic case of any application that has not been designed from scratch to be used in a Cloud environment.

3.1 Unknown internal structure

For the majority of applications, the internal structure may be in practice unknown. This may be due to the complexity of the software or to the usage of external libraries for which no knowledge is available.

3.2 Lack of knowledge for infrastructure environment

The application owner in principle has no knowledge of the management processes inside the Cloud provider or the hardware types, both in number, interconnection and infrastructure capabilities. These may include scheduling, server consolidation, energy management strategies that can significantly influence application behavior [17].

3.3 Multi-tenancy influence

The multi-tenancy concept, thus the concurrent use of the application by a variety of users, may create a number of issues that span from security to performance and availability.

3.4 Variable configuration based on user preferences

When a legacy application is offered as a service, each instance that may be created for each service owner may be configured with different parameters, based on the latter's needs and wishes.

3.5 Various application types with different characteristics and usage of the resources

Different application types with different types of workloads may have varying demands in a specific resource area (e.g. networking, storage etc.).

4 Enablers

The enablers refer to a set of technologies and steps that should be performed during the porting of legacy applications on the Cloud. Through these, the issues reported previously can be alleviated in order to achieve a predictable performance and actually gain from the porting. Investigation in these issues has been performed in the context of the ARTIST FP7 project.

4.1 Legacy Applications Performance Modelling

4.1.1 Correlation of user demands with performance characteristics and workload forecasting

The fact that application performance can vary significantly, based on the user configuration and demand, creates the need for extensive modelling of an application's setup and usage[12]. This may be performed on a component based model creation. However this implies that for each new component (part of the legacy application) a new performance model should be created.

Furthermore, the different patterns of usage should be taken under consideration in order to regulate the resource assignment of cloud services during runtime operation [29].

4.1.2 Application Profiling and Classification based on Stereotypes

Even if the application's needs with regard to the user configuration is known, this does not alleviate the performance issues that may be encountered with relation to application usage of the underlying hardware. In order to overcome this, a behavioural trace/profile of each component should be created, including the patterns of access of the latter on the hardware resources. These patterns can be compared against predefined categories (or stereotypes) for which the performance requirements are known. Furthermore, unlike previous approaches, like [13] , high level probabilistic profiling may not be sufficient. Except for the concrete percentage of I/O requests for example over a period of time, what is also important is how the different actions of the components on the hardware iterate over time e.g. in a sequential or fragmented way. The modelling of these transitions can lead to improved QoS offered by the components, but can also aid in the selection of providers (following strategies for guaranteeing QoS like in [11] and [32]), as is described in the following section.

Furthermore, through suitable description models (based on XML schemas for example), this information can also be transmitted to the different Cloud providers in order to aid them in choosing the fittest resources over which the applications will eventually execute but also aiding them in choosing the way these are scheduled on the physical resources, as dictated by the results of [17].

4.2 Cloud Provider profiling and matching

In a more PaaS-centricd framework, the PaaS provider may also characterize the different IaaS providers based on the benchmarked performance of their infrastructures on a set of predefined stereotypes (the same categories based on which application profiling is conducted). A benchmarking attempt in this frame comes from [14],for a number of different providers and types of applications. For example, due to different internal setup or investment on the infrastructures, an IaaS provider may have improved storage or network ([33]) capabilities in comparison to a competitor. Thus, the PaaS provider should choose this IaaS candidate for applications with extensive needs in storage accesses or I/O. In order to do so however, the PaaS layer should have the aforementioned profiling framework with which it can characterize the type of workload produced by the main components of an application.

Another interesting approach would be to investigate the bottleneck that may be caused in the SaaS due to the Cloud provider's API performance. For example, a specific provider's monitoring API may take too long to respond to calls regarding a service's information. This in turn may have negative effects in the performance of the SaaS in case elasticity rules are used for adapting the resources to the current application workload.

In Figure 1, the enablers described previously are portrayed. In a nutshell, the concrete steps that have been highlighted are the following:

- The definition of suitable description and profile models and stereotypes that capture efficiently the different patterns of resource usage for a limited number of elementary applications (e.g. benchmarks like SPEC 2006, Berkeley Dwarfs etc.).
- The comparison between the traces produced by the SaaS components and the according ones from the elementary benchmarks and the classification of the former to one or more categories.

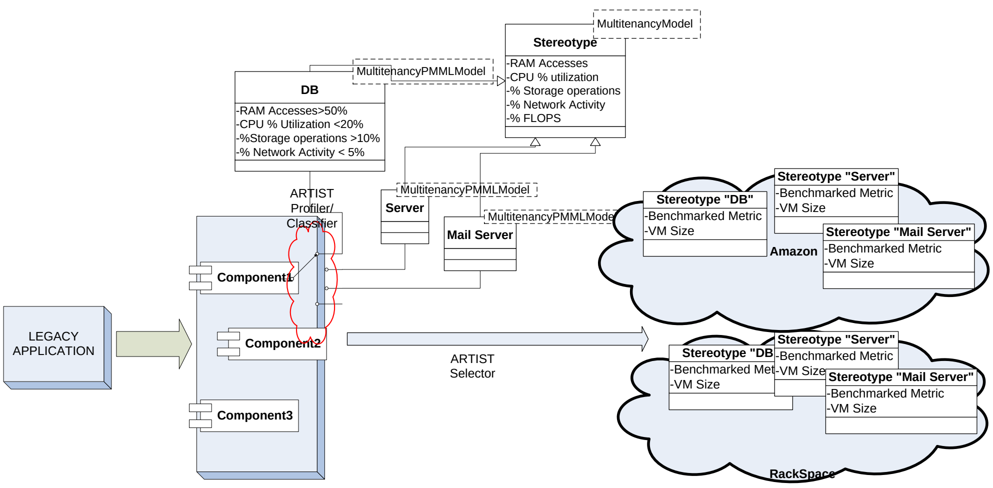

Figure 1: Classification of application components to elementary typical stereotypes and selection of Cloud provider based on the best metrics for this stereotype

- The identification of the fittest provider for deployment through their ratings in the according benchmarks.

4.3 Abstraction Layers based on Functionality Models

Aiming to overcome interoperability issues in Clouds, the UML models and stereotypes described in the previous sections can be enriched with functional characteristics in order to aid in the abstraction layer between different clouds. The major problem with different Cloud vendors (and their differences in their APIs) is twofold:

- The deployment process involved in order to get the virtualized resources running on the IaaS provider and obtain necessary information that must be used in the configuration of the service (e.g. IP addresses of the raised VMs)
- Specialized functionality that may be offered by the IaaS/PaaS provider through its API and that is used in the SaaS source code in order to optimize the performance of the application (e.g. usage of elastic rules based on a combination of billing and network traffic)

For the first issue, there are solutions currently that aim to solve this like [18]. The second issue is significantly more difficult, since it is included inside the source code and refers to specific functionalities that are offered by the IaaS/PaaS and consumed by the SaaS. These functionalities can be included as functional stereotypes in the UML models and incorporate them in the overall model layout, along with specific interfaces to the application code. These abstract and generic interfaces will not change towards the SaaS, but suitable drivers may be used in order to map them to the existing or future implementations of Cloud providers APIs. By using existing approaches for cloud interoperability like [21] and [22] we can focus on the UML modelling layer, coupled with semantics for categorization of the different functionalities offered by the various providers. These drivers may be then used as plug-ins in the SaaS workflow, in an interchangeable way, similar to the LibCloud case for deployment. The process is ideal to be achieved in the early stages of the legacy application transformation to the SaaS model, before any specific IaaS provider has been chosen.

In Figure 2, this approach is presented. In a nutshell, the concrete steps that have been highlighted in the previous sections are portrayed:

- The modelling of SaaS usage of generic IaaS/PaaS APIs in a functionality-based, abstract way with predefined interfaces in order to achieve decoupling from the IaaS specific API implementations
- The creation of suitable drivers that transform these generic interfaces to the specific API of the chosen Cloud provider
- The automatic incorporation of these drivers as intermediates (bridges) prior to the SaaS deployment on every Cloud

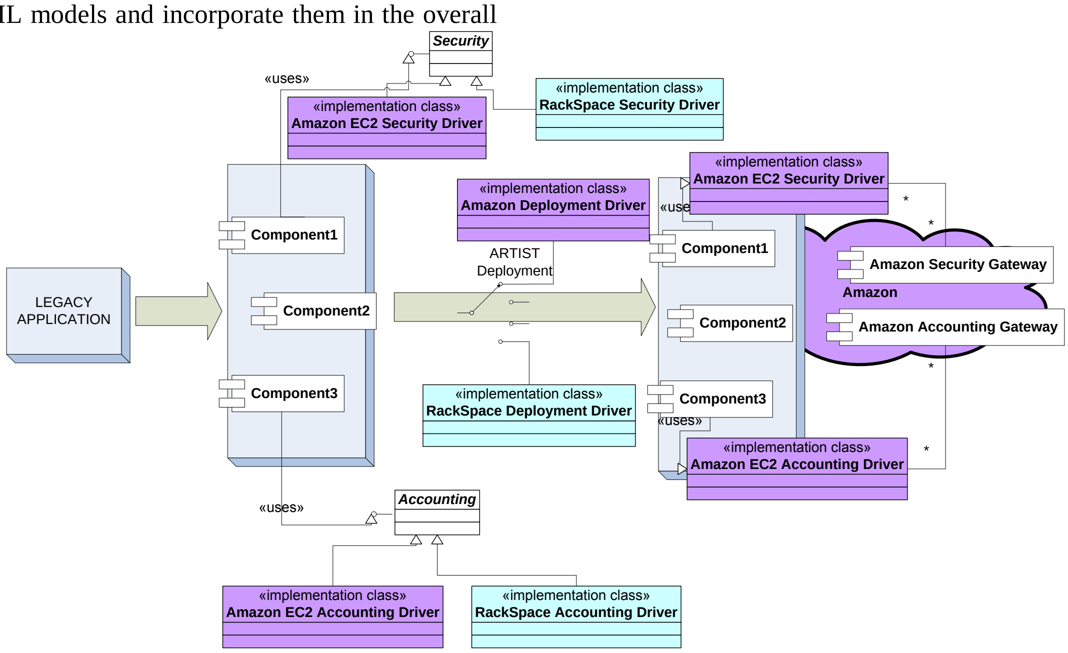

Figure 2: Functionality-based generic modeling of SaaS link to external APIs and usage of this information during deployment for automatic adaptation to different Cloud providers

4.4 Multi-tenancy Trade-off Analysis

Regarding the multi-tenancy issue, there are a number of factors that may influence a decision on the setup of the service. Legal requirements for example [19], functional and performance issues are among them. Approaches exist for automatically inserting necessary components as intermediates for simplifying the development of a multi-tenant application (e.g. [15]).

On the other hand, performance issues need extensive modelling in order to investigate whether the trade-off between a running instance per tenant and a consolidation of tenants in a single instance is worthwhile. Consolidating the tenants in a single instance may lead to increased resources for this instance, in comparison to the simple one instance per tenant approach. However this may not lead to performance improvement due to the bottlenecks inside the application from the increased workload caused by the multiple sources (tenants). For example, choosing carefully the multi-tenants so that their anticipated workloads are complementary with regard to their timeframe of appearance can lead to performance benefits.

This configuration may also be performed during runtime, in order to support a dynamic and adaptive operation of the SaaS on the varying conditions of execution or workload. After the creation of the models, their transformation into a UML friendly format (like PMML [16]) is necessary for their automated use in the context of SaaS management.

4.5 Automated Service Self Configuration and Adaptation

One of the greatest advantages of Cloud computing is the elastic offering of resources and the on-demand adaptation. However, this means that the elastic part of the application should be clearly separated from the remaining sections so that the benefits of scaling would indeed affect the performance. In this direction, research in automatic parallelisation of monolithic applications ([4], [20]) and transformation to a component based system is of extreme importance.

What is more, the ability of services to include parametric design considerations ([30]) so that they can adapt to runtime conditions in order to optimize self-management is becoming increasingly important. This aspect can also be combined with multi-tenancy, in order to achieve a highly automated service management that consumes as few resources as possible without influencing the application's key performance indicators. Another combination can also include control-based approaches ([31]) so that robust service behaviour is maintained without human intervention.

5 Conclusions

In this paper, we have discussed and analyzed a number of open issues around application porting in the Cloud, especially for the case of legacy applications. These refer mainly to the performance aspects of this category and can be seen as a step towards reducing the level of uncertainty that governs service oriented computing.

A number of existing challenges have been identified that originate mainly from the nature of services computing (lack of single entity across the layers, lack of information regarding critical parameters etc.). In this context, a set of enabling technologies and approaches have been highlighted, that can address mainly the topic of application identification and categorization, provider selection and improved performance behavior.

For the future, our goal is to design and implement an operational system that will encompass the features highlighted in this paper in order to mediate during the application adaptation and runtime operation on Cloud platforms.

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] R. Buyya, C. S. Yeo, S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities", 10th IEEE International Conference on High Performance Computing and Communications, 2008

[2] Steve Clifton (Animoto Productions CTO), "Why Cloud Computing is Good for Startups", Workshop on Grids, Clouds & Service Infrastructures, December 2009, Sophia Antipolis, France, available online at: http://docbox.etsi.org/workshop/2009/200912_GRID/Clifton_ANIMOTO.pdf

[3] A. Malis, "Routing over Large Clouds (ROLC) Charter", Meeting minutes of the 32nd IETF meeting, available online at: http://www.ietf.org/proceedings/32/charters/rolc-charter.html

[4] E. Tejedor, J. Ejarque, F. Lordan, R. Rafanell, J. Alvarez, D. Lezzi, R. Sirvent, R.M. Badia, "A Cloud-unaware Programming Model for Easy Development of Composite Services," IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), 2011

[5] New York Times, "Internet Critic Takes on Microsoft", available online at: http://www.nytimes.com/2001/04/09/technology/09HAIL.html?ex=1217563200&en=7c46bdefb6a8450a&ei=5070

[6] J. Barr, "Amazon EC2 Beta", available online: http://aws.typepad.com/aws/2006/08/amazon_ec2_beta.html

[7] A. Lenk, M. Klems, J. Nimis, S. Tai, T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape", ICSE Workshop on Software Engineering Challenges of Cloud Computing, Washington, DC, USA, 23-31, 2009

[8] P. Mell, T. Grance, "NIST Definition of Cloud Computing", Sp. Publication 800-145, available

online at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[9] J. Bisbal, D. Lawless, B. Wu, J. Grimson, “Legacy Information Systems: Issues and Directions”, IEEE Software, 1999

[10] D. Kyriazis, R. Einhorn, L. Furst, M. Braitmaier, D. Lamp, K. Konstanteli, G. Kousiouris, A. Menychtas, E. Oliveros, N. Loughran, B. Nasser, " A Methodology for engineering real-time interactive multimedia applications on Service Oriented Infrastructures", IADIS International Conference Applied Computing, 2010

[11] D. Kyriazis, A. Menychtas, G. Kousiouris, K. Oberle, T. Voith, M. Boniface, E. Oliveros, T. Cucinotta, S. Berger "A Real-time Service Oriented Infrastructure", Annual International Conference on Real-time and Embedded Systems (RTES),Singapore, 2010

[12] G. Kousiouris, D. Kyriazis, S. Gogouvitis, A. Menychtas, K. Konstanteli, T. Varvarigou, "Translation of Application-level Terms to Resource-level attributes across the Cloud Stack Layers", IEEE Symposium on Computers and Communications (ISCC), 2011

[13] K. Younggyun, R. Knauerhase, P. Brett, M. Bowman, W. Zhihua, C. Pu, "An Analysis of Performance Interference Effects in Virtual Environments" Performance Analysis of Systems & Software, 2007, ISPASS 2007

[14] http://cloudharmony.com/benchmarks

[15] R. Mietzner, T. Unger, R. Titze, F. Leymann, "Combining Different Multi-tenancy Patterns in Service-Oriented Applications" IEEE International Conference on Enterprise Distributed Object Computing Conference (EDOC), 2009

[16] http://www.dmg.org/pmml-v3-0.html

[17] G. Kousiouris, T. Cucinotta, T. Varvarigou, "The Effects of Scheduling, Workload Type and Consolidation Scenarios on Virtual Machine Performance and their Prediction through Optimized Artificial Neural Networks”, Elsevier The Journal of Systems and Software, 2011

[18] http://claudia.morfeo-project.org/wiki/index.php/OVF

[19] B. Barnitzke, W. Ziegler, G. Vafiadis, S. Nair, G. Kousiouris, M. Corrales, O. Wäldrich, N. Forgó, T. Varvarigou, “Legal Restraints and Security Requirements on Personal Data and Their Technical Implementation in Clouds”, eChallenges 2011

[20] A. Ferrer, F. Hern´andez, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forg´o, T. Sharif, and C. Sheridan, “OPTIMIS: a Holistic Approach to Cloud Service Provisioning,” Elsevier Future Generation Computer Systems, 2011

[21] http://libcloud.apache.org/index.html

[22] http://www.openstack.org/

[23] A. Azeez, S. Perera, D. Gamage, R. Linton, P. Siriwardana, D. Leelaratne, S. Weerawarana, P. Fremantle, "Multi-tenant SOA Middleware for Cloud Computing," IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010

[24] M. Hajjat, X. Sun, Y.W.E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, M. Tawarmalani. “Cloudward bound: planning for beneficial migration of enterprise applications to the cloud”, SIGCOMM Computing Communications, 2010

[25] S. Frey, W. Hasselbring, “Model-Based Migration of Legacy Software Systems to Scalable and Resource-Efficient Cloud-Based Applications: The CloudMIG Approach”, CLOUD COMPUTING, 2010

[26] P. Mohagheghi, A. J. Berre, A. Henry, F. Barbier, A. Sadovykh, “REMICS- REuse and Migration of Legacy Applications to Interoperable Cloud Services”, Lecture Notes in Computer Science, 2010

[27] W. Zhang, A.J. Berre, D. Roman, H.A. Huru, “Migrating legacy applications to the service cloud”, 14th conference companion on Object Oriented Programming Systems Languages and Applications, 2009

[28] D. Chantry, "Mapping applications to the Cloud” Microsoft Corporation—Platform Architecture Team, January (2009) Online at: http://msdn.microsoft.com/enus/library/dd430340.aspx

[29] G. Kousiouris, A. Menychtas, D. Kyriazis, S. Gogouvitis, T. Varvarigou, " Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in Cloud platforms" , Elsevier Future Generation Computer Systems, 2012

[30] G. Kousiouris, A. Menychtas, D. Kyriazis, K. Konstanteli, S. Gogouvitis, G. Katsaros, T. Varvarigou, "Parametric Design and Performance Analysis of a Decoupled Service-Oriented Prediction Framework based on Embedded Numerical Software," IEEE Transactions on Services Computing, 2012

[31] H. C. Lim, S. Babu, J. S. Chase, S. S. Parekh, “Automated control in cloud computing: challenges and opportunities” 1st workshop on Automated control for datacenters and clouds (ACDC), 2009

[32] X. Huang, M. Ma, "QoS-based Scheduling for Task Management in Grid Computing", 4th International Conference on Information Technology and Applications, 2007

[33] X. Huang, M. Ma, “A performance model for differentiated service over single-hop passive star coupled WDM optical networks”, Journal of Network and Computer Applications, 2011