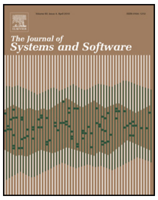
The Journal of Systems and Software 86 (2013) 2321–2326

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss



Solidifying the foundations of the cloud for the next generation Software Engineering

J.L. Vazquez-Poletti*, R. Moreno-Vozmediano, R.S. Montero, E. Huedo, I.M. Llorente

Departamento de Arquitectura de Computadores y Automatica, Facultad de Informatica, Universidad Complutense de Madrid, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 14 November 2012
Received in revised form 9 May 2013
Accepted 11 May 2013
Available online 30 May 2013

Keywords:
Cloud computing
IaaS

ABSTRACT

Infrastructure clouds are expected to play an important role in the next generation Software Engineering but currently there are some drawbacks. These clouds are too infrastructure oriented and they lack advanced service oriented capabilities such as service elasticity, quality of service or admission control to perform a holistic management of a whole application. The deployment of complex multi-tier applications on top of IaaS infrastructures requires to provide the IaaS platforms with an extra service layer that provides advanced service management functionality.

© 2013 Elsevier Inc. All rights reserved.

1. The problem so far

Cloud computing is a provision model that different IT areas, companies and business models can benefit from Foster et al. (2008). Software Engineering is not an exception, specially when considering the following persistent attributes of software (Uyyala et al., 2012):

- Adaptability and scalability: the cloud model supports the software ability to adapt to changing circumstances, through the automatic scaling of software stacks and infrastructures.
- Availability and fault-tolerance: current cloud offerings allow the guaranteed provision of software services in a sustainable way, by providing failover functionality at different levels (virtual machine level, physical server level, or cloud region level).
- Portability and reusability: the cloud market exhibits different de-jure and de-facto standards for cloud interfacing and interoperability that enable the portability and reusability of existing software solutions both partial or totally.
- Security: cloud technologies can implement different authentication, authorization, and auditing mechanisms that enable the protection of software components against danger, loss or unauthorized access.
- Cost and sustainability: the pay-per-use model used in the cloud market allows to reduce the costs of technologies needed for developing, testing and deploying software, applications and services. On the other hand, cloud systems can use different policies for optimal allocation, consolidation, and migration of virtual resources resulting on significant reduction of energy consumption.

However, the term cloud computing involves different provision models, namely Infrastructure, Platform and Software as a Service (IaaS, PaaS, and SaaS) that are wrapped up in the Everything as a Service (XaaS) cloud model. Currently, the most popular and mature technology are the IaaS clouds, boosted by the proliferation of many commercial providers (Amazon EC2,[1] Rightscale,[2] GoGrid,[3] Rackspace,[4] ElasticHosts,[5] etc.) and different technologies for private cloud management (OpenNebula[6] (Milojicic et al., 2011), Eucalyptus[7] (Nurmi et al., 2009), OpenStack,[8] VMware vCenter,[9] etc.). Typically, most IaaS cloud systems and providers manage virtual resources as independent elements, and SLAs are applied individually to each single resource, although recently, some commercial clouds (e.g. Amazon EC2, Rightscale) are providing new features for the definition of groups or arrays of resources to implement, for example, auto-scaling capabilities. However, these features are insufficient to provide elasticity and QoS for a broad range of multi-tier applications deployed in the cloud, so the efficient management of these applications still requires complex IT

[1] http://aws.amazon.com/ec2.
[2] http://www.rightscale.com.
[3] http://www.gogrid.com.
[4] http://www.rackspace.com.
[5] http://www.eslastichosts.com.
[6] opennebula.org.
[7] http://www.eucalyptus.com.
[8] http://www.openstack.org.
[9] http://www.vmware.com.

* Corresponding author. Tel.: +34 913947600.
E-mail address: jlvazquez@fdi.ucm.es (J.L. Vazquez-Poletti).

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.05.063

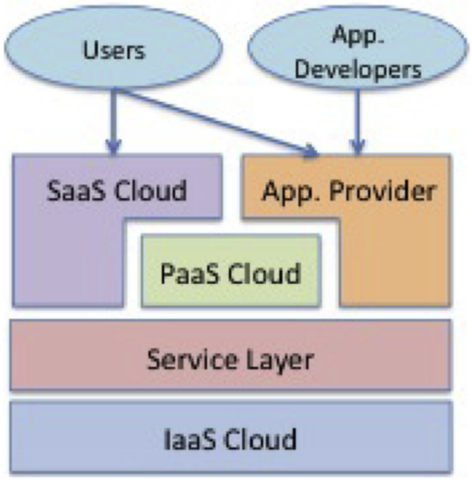


Fig. 1. Layered service-oriented cloud architecture.

landscapes and specialized IT staff. In other words, current IaaS clouds are too infrastructure oriented and they lack advanced service oriented capabilities (e.g. service elasticity, QoS, admission control, etc.) to perform a holistic management of the whole application.

In the present contribution we will introduce the benefits of a cloud based service-oriented architecture, which produces a set of research and scientific challenges. Then, current efforts to face these challenges will be described and finally, some conclusions on the work that still needs to be done at the IaaS level will be provided.

2. A service-oriented architecture

A service-oriented cloud architecture like the one depicted in Fig. 1 would provide the following benefits:

- *Automated service management.* The service-oriented cloud architecture can deal with services as basic entities by provisioning, deploying and scaling the different service components in an automated way. The service layer abstracts users and application providers from the details of the service management, such as, how many instances have been deployed to implement a given service component, which type of instances have been used, or which clouds have been used to deploy this service component.
- *Optimized service provision.* The service-oriented cloud architecture can automatically provision the different service components trying to optimize different quality of service (QoS) paramenters (e.g. service cost, performance, energy consumption, etc.) and satisfying different user constraint (e.g. maximum budget, minimum performance, etc.). In addition, in case of services experiencing fluctuating workloads, the service can be automatically scaled up/down to adapt the service capacity to the variable demand, in order to avoid service performance degradation (in case of increasing demand) or service over-sizing (in case of lowering demand).
- *Brokering mechanisms.* Brokering mechanisms included in the service-oriented architecture enable the optimal deployment of a service among different cloud providers. A cloud broker can provide a uniform interface independently of the particular cloud provider technology, can automatically collect information from providers (instance availability, prices, etc.), and can help cloud users to choose the right technologies when deploying their services across multiple clouds. In addition, in dynamic cloud scenarios (e.g. when new cloud providers appear, an instance type is retreated/added from the cloud market, or instance prices change along the time-line), the cloud broker can decide a better placement of the resources by reallocating the current service infrastructure (or a part of it) to some different clouds.

Example. Let's consider an application that runs on a computing cluster that is deployed on a cloud infrastructure. The application has a minimum performance requirements expressed in MFLOPS. The automated service management is responsible to translate the application specifications to something understood by the IaaS provider once the user decides to deploy the service. These specifications may be the desired performance or restrictions, like sticking to a specific cloud provider in order to avoid having the compute nodes spread geographically. The translation to an IaaS language such as that of Amazon EC2 would be deploying a big number of instances with a small number of compute units[10] or using fewer instances with a high number of compute units instead. The optimized service provision would be responsible of monitoring the application's performance and maintain it within the specified parameters by launching new instances or reorganizing the current ones. Finally, brokering mechanisms would dynamically migrate the current instances totally or partially from a IaaS provider to another if new offerings appear, making the overall cost stay within a given budget.

3. Research and scientific challenges

The capabilities of the previously described architecture can be implemented as an additional service layer built on top of current cloud infrastructures, so that users and application providers can use the functionality of this service layer to provision and deploy their services. In order to accomplish this aim, we have identified three main interrelated scientific challenges:

- *Dynamic service provision in cloud infrastructures.* A multi-tier service can comprise several component/tiers (e.g. a cluster web server, composed by load balancers, web servers, applications servers, and database servers) with some intrinsic dependencies among them. These services can be deployed as a group of interconnected virtual machines in the cloud with specific deployment dependencies and, optionally, some location, affinity, elasticity, and SLA requirements. The service layer should support the definition of complex services, including different interelated components, dependecies, and constrains. The service layer is also responsible for managing the entire life-cycle of a service (including deployment, runtime scaling, and termination), and offering some actions for service handling (deploy, shutdown, suspend, restore, etc.). The research on different admission control policies to accept or reject a service depending on the available resources is also a key topic in the provision of services.
- *Advanced service SLA and elasticity management.* Cloud computing technologies should support standardized service expressions for QoS and SLA negotiation mechanisms between consumers and providers to state the terms under which the service need to be provided. Most current IaaS systems provide simple SLAs based on the availability of the resources. However, more advanced SLA mechanisms should be incorporated at service layer level able to manage service-oriented QoS metrics, such as service performance, service availability, service response time, or service cost, among others. In addition, to guarantee the performance of the service in case of fluctuating workloads and offer an optimal service provisioning, the service layer should also provide some tools for service elasticity management and service scaling, by

[10] http://aws.amazon.com/ec2/faqs/.

allowing the definition of complex elasticity rules based on different types of metrics (including both infrastructure-level metrics and QoS metrics). The efficient management of service elasticity and SLA parameters relies on the monitoring system, that should be adapted to a service-oriented framework in order to monitor and gather information about the state of different infrastructure components (VMs, network devices, storage systems, etc.) and the services deployed.

- *Support for multi-cloud service deployment.* The possibility of deploying services across different IaaS providers and the support for the portability of deployed services are challenges to be addressed. This involves the research on architectures and brokering mechanisms for the aggregation of cloud providers, which can be implemented at service layer level. Most of current cloud brokers do not provide advanced service management capabilities to take automatic decisions, based on optimization algorithms, about how to select the optimal cloud to deploy a service, how to distribute optimally the different components of a service among different clouds, or even when to move a given service component from a cloud to another to satisfy some optimization criteria. So, an open research line in cloud brokering is the integration of different placement algorithms and policies in the broker for optimal deploying of virtual services among multiple clouds, based on different optimization criteria, as for example cost optimization, performance optimization, energy efficiency, proximity, high-availability, etc.

4. Research behind the challenges

Each challenge enumerated before has a wide research field related. The following subsections describe each one.

4.1. Service life-cycle management and admission control

There are some interesting initiatives that address the issue of provisioning services on top of IaaS cloud infrastructures. For example, the RESERVOIR[11] project (Rochwerger et al., 2009), a flagship European project in cloud computing with participation of our research group, was aimed to provide a platform for deployment and management of complex IT services across different administrative domains, IT platforms and geographies. The RESERVOIR cloud model enables the federation and interoperability of infrastructure providers, taken advantage of their aggregated capabilities to provide a seemingly infinite service computing utility. One of the most prominent technologies evolved from the RESERVOIR project is the Claudia[12] platform (Rodero-Merino et al., 2010). In this work, authors propose a new abstraction layer for cloud systems that offers a more friendly interface to service providers by enabling the control of the services lifecycle, and they introduce Claudia, an implementation proposal of such a layer. Another interesting work (Kirschnick et al., 2010) describes an integrated architecture that enables the automated provisioning and management of cloud services. It orchestrates the different steps involved, such as virtual infrastructure management, in addition to installing, configuring, monitoring, running, and stopping software components in these virtual machines. The goal of this project is to enrich and extend the functionality of these existing platforms for service management, by supporting new parameters and constrains in the definition of the service, and implementing new capabilities for managing service SLAs, service elasticity, and multi-cloud deployment.

Regarding the Admission Control (AC) problem, cloud service providers need to choose wisely which services they should accept for provisioning via AC mechanisms. The impact from faults due to unprovisioned resources, which could incur in a violation of the SLAs contracted for the service, has to be minimized and the net income of provisioning has to be maximized at the same time. These faults has been from the client side and fault tolerance mechanisms have been proposed (Ramakrishnan et al., 2009). These situations could be avoided or at least smoothed out from the provider side with the incorporation of admission control mechanisms. The utilization of AC mechanisms has been analyzed in other environments. This is the case of media services provision for on demand devices, where acceptance is subject to a positive revenue (Bichler and Setzer, 2007). Also, jobs with response time guarantees and deadline constraints have been considered in other related work (Islam et al., 2004). Regarding the utilization of AC mechanisms in cloud environments, a policy-driven probabilistic admission control mechanism has been proposed (Llorente et al., 2010) in the context of the RESERVOIR project. This mechanism is based on the concept of acceptable risk level (ARL) to control oversubscribing of capacity. ARL is defined as the probability of having insufficient capacity to satisfy some capacity allocation requests on demand. The ARL value can be derived from a business policy of the IaaS provider that is, more aggressive versus more conservative over-subscription. Another interesting proposal (Dhok et al., 2010) is a learning based opportunistic algorithm for cloud computing environments. With this algorithm, a service is admitted only if it is unlikely to cross the overload threshold set by the service provider. The results of this work show that admission control is useful in minimizing losses due to overloading of resources, and by choosing services that maximize revenue of the service provider. In a recent work (Vazquez-Poletti et al., 2012), our research group present a novel definition of services and SLAs for cloud service providers which will make the tailoring of admission control policies easier and we introduce a double set of admission control policies which are studied from both an economic and failure rate point of view.

4.2. Service SLA and elasticity management

Service Level Agreement is a concept that does not pertain to cloud computing exclusively. Much research has been conducted on SLAs for different other computing environments. Cluster computing is an example, as SLAs have been defined by terms of QoS metrics which comprise utilization, packet loss rate and availability (Xiong and Perros, 2008). Also, algorithms to handle penalties in order to enhance the utility of the cluster based on SLAs have been already proposed (Yeo and Buyya, 2005). Moving to cloud computing, some work in SLAs on a prediction system for minimizing the resource consumption by requests has already being done (Reig et al., 2010). Basically, this prediction system enables the scheduling policies to discard the service of a request if the available resource capability would not reach an established deadline. The RESERVOIR project defines SLAs with conditional rules (Rochwerger et al., 2009). These rules allow dealing with situations that are likely to happen, serving as basic building blocks for autonomic computing. Contemporary cloud SLA mechanisms are normally limited to cost-performance tradeoffs (Comuzzi et al., 2009), but those offered by the OPTIMIS project aim to evaluate levels of trust and risk, even negotiating the use of license-protected software (Ferrer et al., 2012). This way, policies governing the deployment in this cloud infrastructure include aspects such as the level of risk according to cost thresholds, the degree of trust expected from the service provider or the performance levels.

To manage the elasticity of a service, the critical service components can be scaled horizontally (scale in/out) or vertically (scale up/down) (Vaquero et al., 2011). While horizontal scaling

[11] http://www.reservoir-fp7.eu.

[12] claudia.morfeo-project.org.

can be straightforwardly implemented in any cloud platform, by adding instances of the service component to be scaled out, vertical scaling, which involves increasing the capacity of a running instance in terms of CPU, memory and/or storage, is only partially supported by a few cloud providers. Some current IaaS cloud providers, e.g. Amazon EC2 or RightScale, also offer some simple mechanisms for auto-scaling groups of instances, based on alert mechanisms that trigger the deployment of new instances when some condition is met. These mechanisms are mainly based on infrastructure metrics (e.g. CPU load, bandwidth consumption, etc.) and they are usually reactive (i.e. infrastructure reacts to changes in metrics). However, these simple auto-scaling techniques exhibit many limitations: they do not allow a mixture of metrics as input for the auto-scaling algorithm; the auto-scaling algorithms are reactive (they do not make any future estimation, and just react to current infrastructure metrics); they only consider infrastructure level metrics without considering any quality of service (QoS) metric; they do not consider the non-linear effects that appear due to the fixed number of launched and terminated instances; they do not consider the hourly based payment method imposed by many cloud providers. There are some interesting research works that address some of these problems. For example, in Mao et al. (2010) they measure the boot time of instances as well as the shutting down time to make a proper use of the full hour billing model scheduling the instance startup and shutdown time. They also offer a solution for the non-linear auto-scaling effect. In Caron et al. (2010) they identify patterns in the past loads to predict future loads and make pro-active scaling decisions ahead of time. They remark the importance of forecasting based on the fact that new resources are not obtained instantaneously. They also make a study of prediction methods: auto-regression, linear regression, etc. In Vaquero et al. (2011) they address another problem that is often neglected from the auto-scaling consideration: the bandwidth, since the auto-scaling can potentially collapse the network if the network is not properly considered in a previous analysis. In a recent work of our research group (San-Aniceto et al., 2011), we present a method for optimal service provisioning to cover variable computation demands with mixed reserved and on-demand instances with the minimum cost. To avoid the performance degradation of the system, this novel method estimates the number of instances that will be required in the next time period, and provisions the instances in advance to hide start-up times.

4.3. *Cloud brokering*

The current cloud market is composed of several public cloud providers, such as Amazon EC2, GoGrid, or Rackspace, private clouds, which are on-premise infrastructures located managed by some cloud middleware, such as OpenNebula (Milojicic et al., 2011), Eucalyptus (Nurmi et al., 2009), OpenStack, or VMWare vCenter, and hybrid clouds (Montero et al., 2011). These cloud providers and platforms exhibit many differences regarding the functionality and usability of exposed cloud interfaces, the methods for packaging and managing images, the types of instances offered, the level of customization allowed for these instances, the price and charging time periods for different instance types, the pricing models offered (e.g. on-demand, reserved, or spot prices), etc. To help the user to cope with such a variety of interfaces, instance types, and pricing models, cloud brokers have emerged as a powerful tool to serve as intermediary between end users and cloud providers (Buyya et al., 2009). As explained before, a cloud broker is the best option for an uniform interface independently of the cloud provider technology, as it can collect automatically information from different providers while helping user to choose between platforms when deploying services across multiple clouds.

In the current cloud market we can find various commercial brokering solutions, such as RightScale or SpotCloud[13] among others, and also some open-source brokering middleware, such as Aeolus.[14] RightScale offers a private cloud middleware that provides a cloud management platform for control, administration, and life-cycle support of cloud deployments across multiple clouds. It includes a multi-cloud engine that interacts with cloud infrastructure APIs and manages the unique requirements of each cloud. Customers can select, migrate and monitor clouds of their choosing from a single management environment. SpotCloud is another commercial broker that provides a structured cloud capacity marketplace where service providers sell the extra capacity they have and the buyers can take advantage of cheap rates selecting the best service provider at each moment. Aeolus is an open source brokering middleware that allows users to choose between private, public or hybrid clouds, using DeltaCloud[15] cross-cloud abstraction library. Aeolus enable users to automatically bring up a set of different instances on a single cloud or spanning multiple clouds, configure them, and tell them about each other.

Besides these cloud commercial and open-source cloud broker implementations, there are several research works focused on the development of different scheduling algorithms in multi-cloud scenarios, based on different optimization criteria, such as cost or performance. Chaisiri et al. (2009) propose an optimal virtual machine placement algorithm to minimize the total cost due to buying reserved and on-demand resources from multiple clouds. They focus their research in exploring an optimal strategy to avoid the virtual resources over/under-provisioning problem to cope with uncertainty future demand. They achieve this goal adjusting the tradeoff between reserve resources and pay for the on-demand resources needed for load peaks. They compare the benefits of their algorithm against non-reservation, maximum-reservation, and statistical expected reservation cases. Andreolini et al. (2010) present a management algorithm to reallocate the placement of virtual machines for better performance and resource utilization by considering the load profile of hosts and the load trend behavior of the guest, instead of thresholds. Focusing on deployment cost algorithms, Elmroth et al. (2009) propose an accounting and billing architecture to be used in RESERVOIR project. Authors investigate new approaches to simultaneously manage postpaid and prepaid payment schemes that vary over time in response to user needs. And finally, focusing on the variability of several conditions, there are several works regarding optimal deployments of virtual machines. For instance, in a recent work of our research group (Tordsson et al., 2012), we present a cloud brokering approach that optimizes the placement of virtual infrastructures in a multi-cloud scenario. This work is focused only on static scenarios where user and providers conditions do not change along time and the placement decision is take once. In this method, users can request a virtual infrastructure and restrict its deployment to some placement constraints (i.e. favorite clouds to deploy the VMs, or favorite instance types to lease for the virtual infrastructure). We also make a useful comparison between optimal deployments for maximizing the capacity of an infrastructure, but regarding an incremental budget. In a continuation of this work (Lucas-Simarro et al., 2011, 2012), we explore deeply the cloud brokering issue applied to dynamic scenarios, specially where pricing conditions change along time. In this work, the placement action is done periodically reallocating the virtual infrastructure to the best clouds. Hence, we introduce the concept of performance degradation as a placement constraint due to the periodic reallocation action. Moreover, we make a

[13] http://www.spotcloud.com.

[14] http://www.aeolusproject.org.

[15] incubator.apache.org/deltacloud.

comparison between static and dynamic deployments showing the cost improvement potential of using brokering mechanisms.

5. Conclusions: there is still work to do at the basement level

Cloud computing is playing an important role in the provision of flexible, elastic and on-demand IT environments in which areas such as Software Engineering can flourish. Several software attributes such as adaptability, scalability, availability or portability can directly benefit from this provision paradigm. However, before moving to the upper levels (PaaS, SaaS, . . ., XaaS) the foundations of the responsible cloud fabric (IaaS) must be solid.

On the other hand, the abstraction between layers needs to be strengthened. As of IaaS, the current providers need to provide an interface for translating requirements from PaaS and SaaS. Considering a non-monolithic model, a user from the upper layers shouldn't need to define the deployment of an application by terms of machines and networks, but by performance, budget and other operative parameters instead.

Also, there is a need of developing standard methods for advanced service SLA and elasticity management. This, in conjunction with a multi-cloud service placement at IaaS broker level, will allow an efficient scalable deployment of any application.

In this contribution the benefits of an ideal service-oriented cloud architecture have been proposed, keeping in mind how this paradigm benefits the Software Engineering area. Its implementation produces some research and scientific challenges that have been identified along with the current efforts done, which in certain occasions come from other areas. These challenges and efforts are interesting research opportunities that will bring great improvements not only to cloud computing, but to a wide range of areas that benefit from this paradigm.

Acknowledgements

The authors would like to thank the following projects: MEDIANET (Comunidad de Madrid S2009/TIC-1468), ServiceCloud (MINECO TIN2012-31518) and 4CaaSt (European Union's 7th Framework Programme under contract number 258862).

References

Foster, I., Zhao, Y., Raicu, I., Lu, S., 2008. Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, 2008. GCE'08, pp. 1–10.

Uyyala, R., Battini, R., Mishra, K.K., 2012. Persistent software attributes and architecture for distributed application. International Journal of Scientific & Engineering Research 3 (6), 361–370.

Milojicic, D., Llorente, I.M., Montero, R.S., 2011. Opennebula A cloud management tool. Internet Computing, IEEE 15 (2), 11–14.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D., 2009. The Eucalyptus open-source cloud-computing system. In: Proceedings of the 9th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2009), IEEE Computer Society, pp. 124–131.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I.M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., Galan, F., 2009. The reservoir model and architecture for open federated cloud computing. IBM Journal of Research and Development 53 (4), 4:1-4:11.

Rodero-Merino, L., Vaquero, L.M., Gil, V., Galan, F., Fontan, J., Montero, R.S., Llorente, I.M., 2010. From infrastructure delivery to service management in clouds. Future Generation Computer Systems 26 (8), 1226–1240.

Kirschnick, J., Alcaraz Calero, J., Edwards, W.L.N., 2010. Toward an architecture for the automated provisioning of cloud services. Communications Magazine, IEEE 48 (12), 124–131.

Ramakrishnan, L., Koelbel, C., Kee, Y.-S., Wolski, R., Nurmi, D., Gannon, D., Obertelli, G., YarKhan, A., Mandal, A., Huang, T.M., Thyagaraja, K., Zagorodnov, D.,2009. VGrADS: enabling e-Science workflows on grids and clouds with fault tolerance. In: Proceedings of Conference on High Performance Computing Networking, Storage and Analysis (SC 09). IEEE Computer Society, pp. 47–59.

Bichler, M., Setzer, T., 2007. Admission control for media on demand services. Service Oriented Computing and Applications 1, 65–73.

Islam, M., Balaji, P., Sadayappan, P., Panda, D., 2004. Towards provision of quality of service guarantees in job scheduling. IEEE International Conference on Cluster Computing 0, 245–254.

Llorente, I.M., Montero, R.S., Sotomayor, B., Breitgand, D., Maraschin, A., Levy, E., Rochwerger, B., 2010. Cloud Computing: Principles and Paradigms. Wiley, pp. 157–191, Ch. On the Management of Virtual Machines for Cloud Infrastructures.

Dhok, J., Maheshwari, N., Varma, V.,2010. Learning based opportunistic admission control algorithm for MapReduce as a service. In: Proceedings of the 3rd India Software Engineering Conference, ISEC'10. ACM, New York, NY, USA, pp. 153–160.

Vazquez-Poletti, J., Moreno-Vozmediano, R., Llorente, I.,2012. Comparison of admission control policies for service provision in public clouds. In: International Conference on Parallel Computing (ParCo2011), Ghent (Belgium), August 2011, Vol. 22. IOS Press, pp. 19–28.

Xiong, K., Perros, H., 2008. SLA-based resource allocation in cluster computing systems. In: IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008, pp. 1–12.

Yeo, C.S., Buyya, R., 2005. Service Level Agreement based Allocation of Cluster Resources: Handling Penalty to Enhance Utility. IEEE International Cluster Computing 2005, 1–10.

Reig, G., Alonso, J., Guitart, J., 2010. Prediction of job resource requirements for deadline schedulers to manage high-level SLAs on the cloud. IEEE International Symposium on Network Computing and Applications 0, 162–167.

Comuzzi, M., Kotsokalis, C., Spanoudakis, G., Yahyapour, R., Establishing, 2009. Monitoring SLAs in Complex Service Based Systems. In: IEEE International Conference on Web Services, 2009. ICWS 2009, pp. 783–790.

Ferrer, A., Hernandez, F., Tordsson, J., Elmroth, E., Zsigri, C., Sirvent, R., Guitart, J., Badia, R., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forgo, N., Sharif, T., Sheridan, C., 2012. OPTIMIS: a holistic approach to cloud service provisioning. Future Generation Computer Systems 28 (1), 66–77.

Vaquero, L.M., Rodero-Merino, L., Buyya, R., 2011. Dynamically scaling applications in the cloud. SIGCOMM Computer Communication Review 41 (1), 45–52.

Mao, M., Li, J., Humphrey, M., 2010. Cloud auto-scaling with deadline and budget constraints. In: 11th IEEE/ACM International Conference on Grid Computing (GRID), 2010, IEEE, pp. 41–48.

Caron, E., Desprez, F., Muresan, A., 2010. Forecasting for grid and cloud computing on-demand resources based on pattern matching. In: IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), 2010, pp. 456–463.

San-Aniceto, I., Moreno-Vozmediano, R., Montero, R., Llorente, I., 2011. Cloud capacity reservation for optimal service deployment. In: Proceedings of the The Second International Conference on Cloud Computing, GRIDs, and Virtualization, Rome, Italy, IARIA Conference, pp. 52–59.

Montero, R., Moreno-Vozmediano, R., Llorente, I., 2011. An elasticity model for high throughput computing clusters. Journal of Parallel and Distributed Computing 71 (6), 750–757.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., 2009. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25 (6), 599–616.

Chaisiri, S., Lee, B.-S., Niyato, D., 2009. Optimal virtual machine placement across multiple cloud providers. In: Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific, pp. 103–110.

Andreolini, M., Casolari, S., Colajanni, M., Messori, M., 2010. Dynamic load management of virtual machines in cloud architectures. In: Avresky, D., Diaz, M., Bode, A., Ciciani, B., Dekel, E. (Eds.), Cloud Computing, Vol. 34 of Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. Springer, Berlin, Heidelberg, pp. 201–214.

Elmroth, E., Marquez, F., Henriksson, D., Ferrera, D., 2009. Accounting and billing for federated cloud infrastructures. In: Eighth International Conference on Grid and Cooperative Computing, 2009 (GCC'09), pp. 268–275.

Tordsson, J., Montero, R., Moreno-Vozmediano, R., Llorente, I., 2012. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems 28 (2), 358–367.

Lucas-Simarro, J., Moreno-Vozmediano, R., Montero, R., Llorente, I., 2011. Dynamic placement of virtual machines for cost optimization in multi-cloud environments. In: Proceedings of the 2011 International Conference on High Performance Computing & Simulation (HPCS 2011), IEEE, pp. 1–7.

Lucas-Simarro, J., Moreno-Vozmediano, R., Montero, R., Llorente, I., 2012. Scheduling strategies for optimal service deployment across multiple clouds. Future Generation Computer Systems 29 (6), 1431–1441.

Jose Luis Vazquez-Poletti is an Assistant Professor in the Department of Computer Architecture and Systems Engineering at Complutense University of Madrid (UCM). His research interests include distributed computing, focusing on high-performance in cloud infrastructures. Vazquez-Poletti received a PhD in computer architecture from UCM.

Rafael Moreno-Vozmediano is an Associate Professor in the Department of Computer Architecture and Systems Engineering at Complutense University of Madrid (UCM). His research interests include high-performance and distributed computing, virtualization, and cloud computing. Moreno-Vozmediano has a PhD in computer architecture from UCM.

Ruben S. Montero is the Chief Architect of the OpenNebula Project, a co-founder of C12G Labs, and an Associate Professor at UCM. Montero received a PhD in computer

architecture from UCM. His research interests include resource provisioning models for distributed systems and cloud computing.

Eduardo Huedo is an Associate Professor at UCM. Huedo received a PhD in computer architecture from UCM. His research interests include high-performance, distributed, grid and cloud computing.

Ignacio M. Llorente is the Director of the OpenNebula Project, a co-founder of C12G Labs, and a Full Professor at UCM. Llorente received a PhD in computer architecture from UCM and an executive MBA from the Instituto de Empresa. His research interests include high-performance computing, virtualization, cloud computing, and grid technology. He is a senior member of IEEE.