A Methodology for Aiding Users to Design and Model Cloud Computing Architectures

Alberto Núñez, Pablo C. Cañizares and Manuel Núñez
Universidad Complutense de Madrid, Facultad de Informática
Madrid, Spain
alberto.nunez@pdi.ucm.es, pablo.cerro.canizares@gmail.com, mn@sip.ucm.es

***Abstract*—During the last years, mobile technology has totally changed our life style by acquiring a very important role in a lot of aspects of the human life. This technology requires a robust back-end system able to both allocating services that support applications and managing very large amounts of data. This system is known as cloud computing. Thus, developers are not only concerned about programming applications, but designing and dimensioning cloud systems able to execute services that support them.**

The design of cloud systems is challenging and requires a certain degree of expertise in the field of distributed systems. Moreover, the complexity of this task increases when we consider aspects such as the vast range of possible configurations, pricing and energy consumption. In this paper we present a methodology for helping users to design cloud computing systems targeted to fulfil features such as performance, cost and energy-saving.

***Keywords*-Cloud Aided Design; Cloud computing; Modelling and Simulation;**

I. Introduction

In the last decade, the (r)evolution of Information Technology has totally changed our life style by acquiring a very important role in a lot of aspects of the human life, like communicating with other people, practising sports or watching TV [1]. ITU statistics suggest that almost 6 billion people have mobile phone subscriptions [2], it being almost the 90% of the world plugged in via cellular phone.

This era is also marked by the fact that mobile applications have become in a very lucrative and competitive market. As an example, at Apple's Worldwide Developer's Conference in San Francisco, the company announced in June 2^{nd}, 2014, that the iOS App Store has reached 1.2 million applications. At the time, the company also noted it had paid developers $10 billion. Similarly, its main competitor Google announced that the Android marked contains 1.3 million applications.

This mobile technology, strengthened by millions of applications, requires a robust back-end system able to both allocate services that support such applications and manage very large amounts of data. The result was an imminent integration with cloud computing systems for supporting these applications [3].

Although the concept of cloud computing were formulated in 1997 [4], cloud computing emerged in 2007, showing a noticeable adoption by several top enterprises in communications such as Google [5], IBM [6], Microsoft [7], and Amazon [8]. Although there are currently several definitions for explaining what is cloud computing, the one provided by U.S. NIST (National Institute of Standards and Technology) seems to provide a formal definition including the key elements used in the cloud computing community [9]: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

According to the cloud deployment model [9], each cloud system can be categorised in: public, where the services can be outsourced by paying each virtualised resource per unit of time. The cloud infrastructure is provisioned for exclusive use by a single organisation comprising multiple consumers; private, when the system is a single private data centre used, managed, and hosted by an organisation; hybrid, with a composition of both described before; and finally, science cloud, when the cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organisations that have shared concerns.

Since the dramatic evolution of mobile applications and smart phones, developers of these applications are not only concerned about programming the application to be executed on the mobile device, but designing and dimensioning a cloud system able to execute services that support these applications. However, the best cloud architecture for one application is not necessarily the best solution for other applications. Cloud system design is challenging and it requires a certain degree of expertise in hardware and architecture of distributed systems. Moreover, this task is more complex if we have into account the huge range of possible configurations, pricing and energy consumption, which drastically increases its complexity.

This paper presents a methodology focused on aiding users to design cloud architectures targeted to improve the user's requirements. This methodology is supported by a complete simulation platform to model and simulate complete cloud systems, called iCanCloud [10], [11].

Basically, our proposed methodology takes as input a set of parameters, like the total number of machines to be allocated in the cloud, features to be improved by the cloud (e.g. performance, cost or energy-saving) and the service to be executed in the cloud. As a result, this methodology will guide the user to design and build a cloud system by using only those components that satisfy the parameters provided by the user. This selection of devices drastically reduces the wide spectrum of possible configurations, allowing the user focusing on the relevant and important characteristics of the system.

Thus, we propose a concept which is not only about designing cloud systems, but mainly to suggest the best components to the user for configuring a cloud system given a specific target. In order to achieve this goal, different algorithms balance the target system and calculate those devices that better fit with the final objective.

The results obtained from the application of our methodology are reported on this paper, supported by the iCanCloud simulation platform to model and simulate cloud computing architectures. The main contributions of this work can be summarised as:

- In contrast with previous work that need the user to provide a complete design of the cloud, our approach uses relevant information of the final system and automatically discard those components that would produce a system out of the user's scope. Thus, a reduced list of components, like disks, CPUs and networks are suggested to the user in order to ease the design of the required cloud system.
- In order to analyse the quality of the designed cloud systems, those are modelled and simulated using the iCanCloud simulation platform. Thus, customised metrics can be gathered during the simulation to compare different cloud systems.
- A complete cloud system has been designed and tested by using our proposed methodology. This experiment has been conducted by executing the generated models in the iCanCloud simulation platform.

The rest of the paper is structured as follows. Section II describes the motivation for this work. Section III presents related work. Section IV shows an overview of the iCanCloud simulation platform. Section V describes our proposed methodology for aiding users to design and model cloud systems. Section VI shows an experiment where different cloud systems, aimed to improve different features, are designed by using our proposed methodology. Finally, Section VII presents our conclusions and some directions for future work.

II. Motivation

Since the rapid evolution of Internet and communications networks, IT companies and organisations are continuously looking for solutions that would help to reduce their cost structures and improve profitability. Currently, *shifting to the cloud* seems to be the current focus in order to reach these goals.

However, although public clouds probably are the easiest and cheapest option, where companies pay only for the requested resources (VMs and storage) and then they upload their data for launching applications via the Internet, there are some scenarios where public cloud is not always the best solution.

For example, public cloud environments hide some details to the user, like network topology and hardware architecture, using virtualisation to share resources and therefore, increasing the concurrency of users at the cost of decreasing the overall system performance. Hence, some applications are not suitable to be executed on cloud environments, specifically those ones that require exclusive access and very precise measurements of the hardware [12], [13]. Moreover, confidentiality and integrity of the stored data is deemed as one of the major challenges to be addressed in public cloud architectures [14], [15]. In a private cloud, users have complete control over their own data and resources, while in a public cloud they do not have the warranty that their data could not be accessed by other users.

Generally, public cloud systems are a good solution for those applications that do not require to fully exploiting the computing resources neither deploying extreme security mechanisms to protect data. For instance, those scenarios that require customised cloud architectures to fully exploiting the owned resources in order to provide the best QoS (Quality of Service) [16], like e-Health and online gaming, make very complicated to take the decision of migrating to a public cloud.

Provision of health services using digital technology has been termed as e-Health [17]. In the last few years, E-health monitoring systems has been a major trend that have attracted great attention of IT companies and researchers, developing a wide-spectrum of applications [18], [19]. Generally, the amount of data to be stored in the cloud for each patient dramatically increases in every minute, while the time required to access these data must be performed efficiently in real-time [20]. Moreover, traditional network security mechanisms are also not sufficient for the data outsourced for storage [21]. These challenges are present when developing an e-health system in a public cloud, requiring extreme and strict security systems and a high performance in the overall virtualised environment.

On the other hand, online gaming produces an important percentage of worldwide network traffic, where a very large number of players interact with one another within a virtual game world. Over 55% of Internet users are now also online gamers [22]. Specifically, the genre of MMORPGs (Massively Multiplayer Online Role-Playing Game) is known as a killer application of Internet, with a global subscriber number increased to 17 millions in 2010.

Usually, these games need a dedicated data centre to comply with the variable computational and low-latency demands of the players. One of the most important challenges of this kind of applications is the architecture of the data centre that supports the game. Unlike web applications, there are no standard or even de facto architectures for online game systems. Thus, gaming companies invest dedicated hardware for each game.

These features hamper the migration of this service to a public cloud because of performance, scalability and latency issues. The authors of [23] state that using virtualised resources can negatively affect the quality of gameplay when the system is heavily loaded. Hence, QoS is the critical part of the cloud deployment to ensure a good quality of the game, even when there exists a vast number of users.

In conclusion, there are several factors, like QoS, performance and security, that play an important role when deploying a service in a cloud system. Usually, the service provider must design the underlying cloud architecture that fits better with the requirements of the service to be executed in the cloud. However, due to the high number of devices that have a direct impact on the overall performance (e.g. CPUs, disks, communication networks, etc), building a system that improves a specific feature (e.g. performance, low energy consumption and low latencies) for a given service is a very difficult and complicated task.

Due to the high demand of customised clouds that support services, in this paper we present a methodology that helps users to design and build cloud system architectures. The iCanCloud simulation platform supports this methodology by providing a GUI by allowing users to select and design the cloud system. Once the system is modelled, the iCanCloud simulator is able to execute the required service into the simulated environment.

Let us emphasise the fact that since all the possibilities are simulated, the user of our methodology does not need to make an investment until different solutions are proposed. In addition, simulation has many advantages:

- Simulation experiments are cheap. On the one hand, both simulation platform and the operating system (Linux) are totally free. On the other hand, simulations can be executed in small commodity clusters or even in single regular computers.
- Simulation is more flexible than performing experiments in a real cloud system. In a simulation experiment the entire model can be easily modified by changing configuration parameters. On the contrary, real systems require to make changes in the hardware, which is usually more expensive and a time-consuming task.
- Simulation does not require specific hardware to launch experiments.
- The total time for executing a simulation can be reduced by performing parallel simulations. In this case, a cluster is needed to execute such simulations.
- Scaling the architecture of a real system is more expensive and time-consuming that performing the same changes in a simulated environment.
- Simulators can be shared easily with other researchers, while it is more difficult to do it with hardware.

III. Related Work

In recent years, simulation has become a widely adopted approach for estimating performance and energy consumption in cloud systems. The developer builds a simulation model that imitates the behaviour of the target system and then different measures, like performance and power consumption, are gathered by running simulations.

During last years, simulation has become an adopted approach for testing cloud systems. Basically, simulators built a model of the system to be simulated, such that it imitates the behaviour of the target system and then different measures, like performance and power consumption, are gathered by observing how the model works. To the author's knowledge, the simulation tools that better fit the purpose of modelling and simulating cloud computing environments are CloudSim [24], GreenCloud [25], SimGrid [26] and Virtual-GEMS [27].

Although a wide-spectrum of simulators can be found in literature, to the best of our knowledge there are few works focused on aiding users to design cloud architectures.

CDR (Cluster Design Rules) is an engineering tool that uses resource constraints and application performance models to identify the few best designs among all possible combinations of designs that could be constructed using parts from a given database. It is mainly focused on high performance clusters. Furthermore, CDR evaluates the designs it creates using a specific metric function of the HPL benchmark [28] and SWEEP3D benchmark [29]. However, it is not possible to simulate new applications written by users using the built system.

The authors of [30] use cloud computing for designing distributed embedded systems, where the cloud is used as a simulation platform. This platform allows the design and development of distributed embedded systems.

The approach presented by Shih-Hao Hung et al. [31] describes the evaluation of the performance of an M2M system by running the M2M software over the virtual machines which collectively simulate the M2M system. The virtual machines are connected with our virtual network devices (VND) that model the network in the M2M system. This approach allows the developer to deploy unmodified software onto the simulation environment and use our tools to analyse the execution time, energy consumption, and network transactions on each virtual machine.

These previous works are based on simulating specific distributed and cloud systems. However, these approaches do not provide assistance in order to completely design a cloud system.

IV. Overview of iCanCloud

iCanCloud is a simulation platform aimed to model and simulate cloud computing systems. The main objective of iCanCloud is to predict the trade-offs between performance, cost and energy consumption of a given set of applications executed in a specific hardware, and then provide to users useful information about these metrics.

The simulated cloud computing scenarios are modelled using a set of existent components provided by iCanCloud; they represent the behaviour of real components that belong to real architectures like disks, networks, memories, file systems, etc. Those components are hierarchically organised within the repository of iCanCloud, which compose the core simulation engine. Figure 1 shows the basic architecture of iCanCloud.

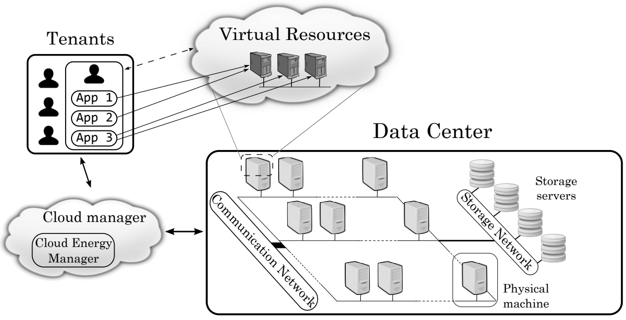


Figure 1. Architecture of the iCanCloud simulation platform

The tenants, grouped in the upper-left box, request resources to the cloud manager, that attends petitions by following a previously set-up scheduling policy. This policy can be fully customisable by the user. Each tenant can be also fully modelled in the cloud system to represent a mobile device, a Web application or a service.

Basically, a VM (Virtual Machine) is configured in iCanCloud by setting 4 systems: Network, HHDD (Hard Disk), CPU and Memory. Although this schema shows a high level of abstraction, the simulation platform must be configured by using a high number of parameters, which also provides a high level of detail for simulating this kind of systems. The iCanCloud platform allows modelling and simulating different types of VMs in the same cloud environment.

Initially, users may create a cloud model by using the GUI module jointly with the cloud repository. A cloud model consists of the configuration of those components that defines a cloud system. This configuration includes, among other components, the network topology, aggregation of physical machines in racks, definition of physical machines, virtual machines and software pieces like managers and schedulers.

Once this model is done, the simulator takes it as input to start the simulation. In this case, the iCanCloud simulator is used to simulate cloud systems. This simulator uses text files, written using the NED language, to configure cloud models. In some cases, these files may consist of thousands of text lines. Wherefore, editing these files manually may become a tedious and time-consuming task for users. In order to alleviate this task, the GUI module generates these configuration files automatically. In fact, this module can be seen as a translator that translates data entered by users using a graphical interface, to text files written in NED language readable by the simulator.

V. Description of the proposed methodology

This section describes in detail our proposed methodology for aiding users to design and model cloud systems. The iCanCloud simulation platform contains a repository of components that represent the analogous devices of a real cloud system. This repository is scalable a flexible, in the meaning of new models can be added to repository and current ones can be modified or removed. This feature essentially increases the functionality of the simulation platform because as the repository increase in size, the number of possible cloud configuration increases as well. This repository can be fully managed from the GUI (see Figure 2).

Figure 3 shows the hierarchical architecture of the main parts needed to design a cloud system using the proposed methodology. Basically, a cloud system consists of a set of racks, a communication network, a cloud manager and a set of VM configurations. The physical machines are allocated in the racks, where each rack contains a set of switches and a set of nodes. Depending of the main objective of the rack, it may contain computing nodes or storage nodes. While computing nodes contain powerful CPUs and high amounts of RAM, storage nodes focuses on high and fast storage devices. Finally, each node is configured by combining 5 basic systems: CPU (computing), disk (storage), memory, PSU (Power Supply Unit) and a network interface.

The high complexity of combining all these elements hampers the analysis of every combination that defines a cloud system. Moreover, if new features like pricing, performance and energy consumption have to be taken into account, this complexity grows exponentially, making very hard the task of designing a cloud system. Thus, the main objective of the proposed methodology is to ease this task by reducing this number of combinations.

The first step is to define the main target to be improved by the designed cloud system. The user must select two of the following three features: cost, performance and energy consumption. The proposed algorithm takes as input these features in order to process the components of the repository, where only those that provide an improvement towards the previously provided features, are suggested to the user (see Algorithm 1). Thus, the number of possible combinations is significantly shortened, and therefore its underlying complexity has been also decreased. Our proposed methodology is described in detail in Figure 4.

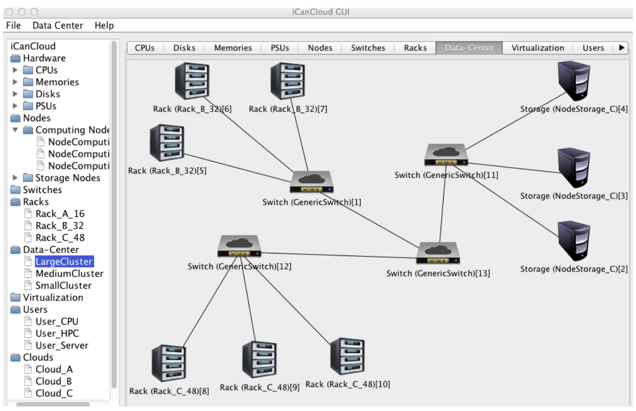


Figure 2. Screenshot of the iCanCloud's GUI

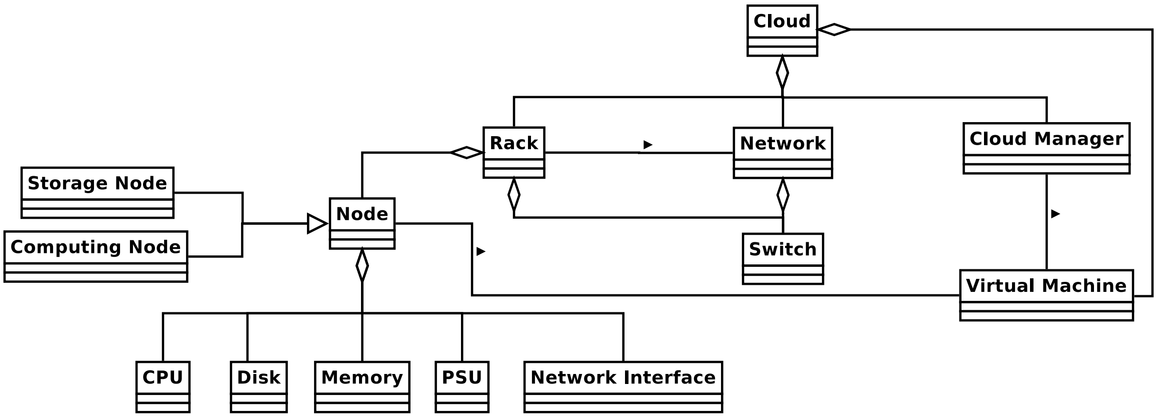


Figure 3. Global schema of a cloud design built using the iCanCloud simulation platform

1) The main target feature f_m must be provided. The user must select f_m from $list_f = \{cos, per, ene\}$. Where cos means that the final cloud is aimed to reduce the total cost, per means that the final cloud is aimed to increase the overall system performance and ene means that the final cloud is aimed to reduce the total energy consumption.
2) The secondary target feature f_s must be provided from $list = list_f / \{f_m\}$.
3) The total number of physical machines that will be allocated in the final cloud model must be provided.
4) The policy for allocating VMs in the cloud must be provided by the user from the list of suggested policies.
5) A communication network must be provided by the user from the list of suggested networks.
6) Computing nodes and storage nodes must be designed by configuring the next basic systems:
   a) A CPU model must be provided to configure the computing system. The list of CPU devices suggested to the user are allocated in $list_{cpu}$, such that $list_{cpu} = suggestedComponents(cpu, f_m, f_s)$.
   b) A Disk model must be selected to configure the storage systems. The list of Disk devices suggested to the user are allocated in $list_{disk}$, such that $list_{disk} = suggestedComponents(disk, f_m, f_s)$
   c) A Memory model must be selected to configure the RAM memory. The list of Memory devices suggested to the user are allocated in $list_{mem}$, such that $list_{mem} = suggestedComponents(memory, f_m, f_s)$
   d) A PSU model must be selected to configure the power supply unit. The list of PSU devices suggested to the user are allocated in $list_{psu}$, such that $list_{psu} = suggestedComponents(psu, f_m, f_s)$
   e) A Network Interface model must be selected to configure the network systems. The list of Network Interfaces suggested to the user are allocated in $list_{net}$, such that $list_{net} = suggestedComponents(net, f_m, f_s)$

Figure 4. Cloud design methodology

VI. EXPERIMENTS

In this section we present some experiments in order to show the validity of our proposed methodology. Basically, these experiments consist in designing and simulating two different cloud systems, *Cloud 1* and *Cloud 2*. The main objective is to compare the results obtained from two different clouds targeted to improve different features, like performance and energy consumption. The input parameters for these experiments, provided by the user, are shown in Table I.

Table I
INPUT PARAMETERS FOR DESIGNING 2 CLOUD SYSTEMS

Input	Cloud 1	Cloud 2
Computing nodes	64	64
Storage servers	16	16
Cloud Manager	First Fit	First Fit
Main feature	Performance	Energy consumption
Secondary feature	Cost	Performance

Both cloud systems have been configured to allocate 80 nodes, it being 64 computing nodes and 16 storage nodes. The algorithm provided to allocate VMs in the cloud is First-Fit. Basically this algorithm selects the first available physical machine able to host the required VM. The main and secondary features to be improved in *Cloud 1* are performance and cost, respectively, while *Cloud 2* has been designed to improve energy consumption and performance.

Table II shows the design of *Cloud 1* and *Cloud 2* generated by using the methodology proposed in this work. The total cost of *Cloud 1* is 84.560€, while the cost of *Cloud 2* is 37.655€.

In this experiment, the same workload has been executed in both cloud systems. Basically, this workload consists in simulating 1000 users requesting data to the servers executed in the cloud. In order to evaluate the quality of each cloud, two different metrics have been gathered from this experiment, the total amount of energy consumed by each cloud system and the average response time for each user.

Figure 5 shows the total amount of energy required to execute the workload in each cloud. However, although the shape of these charts are similar, we can appreciate that *Cloud 1* (see Figure 5(a)) requires more energy than *Cloud 2* (see see Figure 5(b)). The main reason of this difference in the energy consumption, 3e+06 J of *Cloud 1* versus 2e+06 J of *Cloud 2*, lies in the components used to design each cloud. While the devices of *Cloud 1* are powerful and fast because they are focused on increasing performance, and therefore they require more energy, the devices used to design *Cloud 2* focus on saving energy, requiring less amount of energy.

Algorithm 1 suggestedComponents(type, f_m, f_s)

MAX_SUGGESTED_DEVICES ← 3;
TRH_W ← 2; // Threshold for secondary feature
// Initialization...
weightBalancedList ← ∅;
suggestedList ← ∅;
listDevices ← getListOf(type);
// Calculates averages for each feature
$\bar{d}_{per} \leftarrow$ getAveragePerformance (listDevices);
$\bar{d}_{cos} \leftarrow$ getAverageCost (listDevices);
$\bar{d}_{ene} \leftarrow$ getAverageEnergy (listDevices);
// Calculates weights given f_m and f_s
for (each d ∈ listDevices) **do**
  if ((f_m = 'cos') AND (f_s = 'per')) **then**
    $d_w \leftarrow (d_{cos} - \bar{d}_{cos}) + ((d_{per} - \bar{d}_{per}) \backslash TRH\_W)$
  end if
  if ((f_m = 'cos') AND (f_s = 'ene')) **then**
    $d_w \leftarrow (d_{cos} - \bar{d}_{cos}) + ((d_{ene} - \bar{d}_{ene}) \backslash TRH\_W)$
  end if
  if ((f_m = 'per') AND (f_s = 'cos')) **then**
    $d_w \leftarrow (d_{per} - \bar{d}_{per}) + ((d_{cos} - \bar{d}_{cos}) \backslash TRH\_W)$
  end if
  if ((f_m = 'per') AND (f_s = 'ene')) **then**
    $d_w \leftarrow (d_{per} - \bar{d}_{per}) + ((d_{ene} - \bar{d}_{ene}) \backslash TRH\_W)$
  end if
  if ((f_m = 'ene') AND (f_s = 'cos')) **then**
    $d_w \leftarrow (d_{ene} - \bar{d}_{ene}) + ((d_{cos} - \bar{d}_{cos}) \backslash TRH\_W)$
  end if
  if ((f_m = 'ene') AND (f_s = 'per')) **then**
    $d_w \leftarrow (d_{ene} - \bar{d}_{ene}) + ((d_{per} - \bar{d}_{per}) \backslash TRH\_W)$
  end if
  // Insert current device in the weight-balanced list
  insertSorted (d, weightBalancedList);
end for
// Insert devices in the suggested list
for (i=0 to MAX_SUGGESTED_DEVICES-1) **do**
  insert (weightBalancedList[i], suggestedList)
end for
return suggestedList

Table II
DESIGN OF *Cloud 1* AND *Cloud 2*

Cloud 1	*Cloud 2*
Computing Nodes	
CPU intel i7 3970	CPU intel G630
16GB Kinston memory	8GB Kinston memory
500GB HHDD Maxtor	500GB HHDD Maxtor
750W PSU	750W PSU
Storage Nodes	
CPU intel i7 3770	CPU intel G630
8GB Kinston memory	4GB Kinston memory
3x1TB HHDD Maxtor	3TB HHDD Maxtor
750W PSU	750W PSU
Communication network	
3xSwitch Ethernet 10gbps	3xSwitch Ethernet Gigabit

Figure 6 shows the average response time for each user in the designed clouds. These charts clearly show that *Cloud 1* use more powerful components than *Cloud 2*. The response time is almost two times faster in *Cloud 1* (see Figure 6(a)) than in *Cloud 2* (see Figure 6(b)). Both charts show that the first group of users obtain quick responses. This situation mainly happens because when the requests of these users arrive to the cloud system, the resources of the cloud are not saturated, and therefore the requests are quickly processed. However, once the resources of the cloud are working on attending requests, the average response time is stabilised.

In conclusion, we can state that each designed cloud improves well the features provided by the user. *Cloud 1* is focused on performance, which is reflected in its elevated cost and the good results obtained in the average response time (see Figure 6). However, this cloud requires high amounts of energy to support these devices. On the contrary, *Cloud 2* is focused on saving energy, which can be reflected in its low energy consumption (see Figure 5).

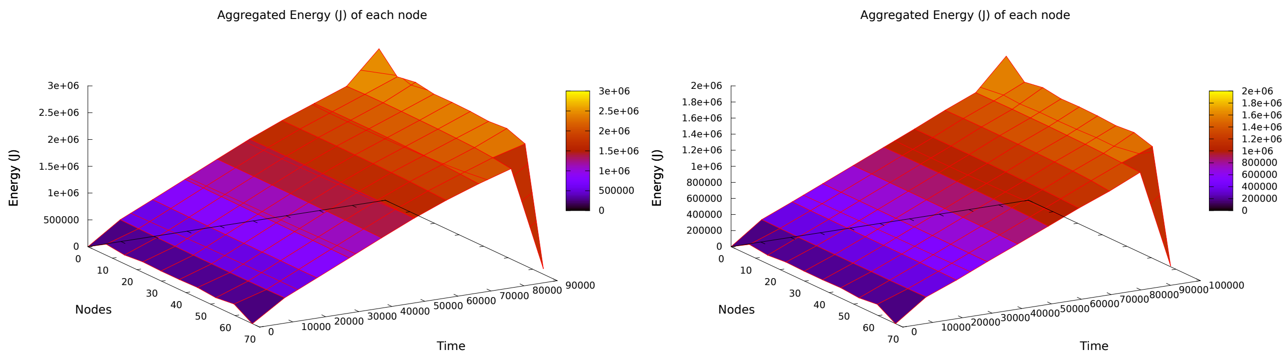
VII. CONCLUSIONS AND FUTURE WORK

This paper presents a methodology for aiding users to design cloud systems targeted to improve specific features. Generally, design cloud systems is challenging due to the vast number of possible combinations to build the system, which in most cases become a hard and tedious task. The main objective of our proposed methodology is to ease this task by reducing its complexity and saving both effort and time for the cloud designer.

This methodology is supported by the iCanCloud simulation platform, which is used to model and simulate the designed cloud systems. This allows representing the behaviour of the designed clouds in a simulated environment, allowing to gather relevant information of this behaviour in order to estimate the quality of such design.

In order to show the applicability of the proposed methodology, two different cloud systems have been designed by using this methodology, each one targeted to improve different features. Also, the resulting cloud designs have been simulated using the iCanCloud simulation platform. In these simulations, different metrics have been gathered in order to analyse the quality of each design.

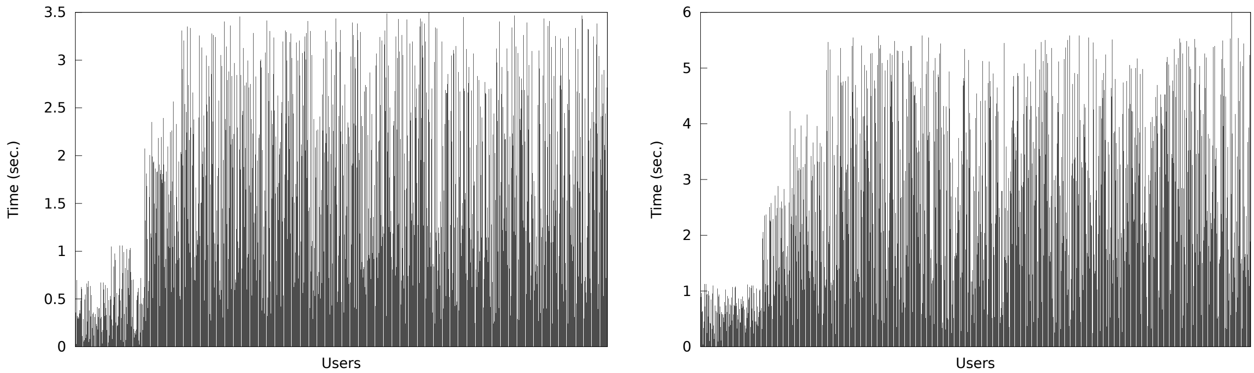
The results of these experiments show that each cloud improves well the features configured by the user. While *Cloud 1* is targeted to improve performance, *Cloud 2* focuses on saving energy. On the one hand, *Cloud 1* requires more energy than *cloud 2*, and also obtain better



(a) Aggregated energy required for *Cloud 1*

(b) Aggregated energy required for *Cloud 2*

Figure 5. Energy consumption



(a) Average response time for each user in *Cloud 1*

(b) Average response time for each user in *Cloud 2*

Figure 6. Average response time

performance results. On the other hand, *Cloud 2* consumes less energy than *Cloud 1*, but the performance results obtained are worse than the results obtained in *Cloud 1*.

Future work will present more features to be improved in the designed cloud systems. Also, different workloads will be analysed and designed in order to compare the trade-offs between the features defined by the user.

ACKNOWLEDGMENT

This research was partially supported by the Spanish MEC projects TESIS (TIN2009-14312- C02-01) and ESTuDIo (TIN2012-36812-C02-01).

REFERENCES

[1] C. Atkins, K. Koyanagi, T. Tsuchiya, T. Miyosawa, H. Hiroseand, and H. Sawano, "A cloud service for end-user participation concerning the internet of things," in *9th International Conference on Signal-Image Technology Internet-Based Systems (SITIS'13)*, Dec 2013, pp. 273–278.

[2] I. T. U. World-Telecommunication. (2011) The World in 2011: ICT Facts and Figures. Http://itu.int/ITU-D/ict/facts/2011 (last accessed 22 September 2014).

[3] M. Satyanarayanan, "Mobile computing: The next decade," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing #38; Services: Social Networks and Beyond (MCS '10)*. ACM, 2010, pp. 5:1–5:6.

[4] L. Mei, W. K. Chan, and T. H. Tse, "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues," in *3rd IEEE Asia-Pacific Services computing conference, APSCC'08*. IEEE Computer Society, 2008, pp. 464–469.

[5] "Google and the wisdom of clouds," Business Week, 2007. Available at http://www.businessweek.com/magazine/ content/07_52/b406404 8925836.htm.

[6] K. A. Delic and M. A. Walker, "Emergence of the academic computing clouds," *Ubiquity*, vol. 2008, no. August, pp. 1–1, Aug. 2008.

[7] "Microsoft plans 'cloud' operating system," New York Times. 2008. Available at http://www.nytimes.com/ /2008/10/28/technology/ 28soft.html.

[8] Amazon Elastic Compute Cloud, Web page at http://aws.amazon.com/ec2/. Date of last access: 10th August, 2013.

[9] P. Mell and T. Grance, "SP 800-145. The NIST Definition of Cloud Computing," National Institute of Standards & Technology, U.S. Department of Commerce, Gaithersburg, MD, United States, Tech. Rep., 2011.

[10] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "iCanCloud: A flexible and scalable cloud infrastructure simulator,"

Journal of Grid Computing, vol. 10, no. 1, pp. 185–209, 2012.

[11] G. Castañé, A. Núñez, P. Llopis, and J. Carretero, "E-mc^2: A formal framework for energy modelling in cloud computing," *Simulation Modelling Practice and Theory*, vol. 39, pp. 56–75, 2013.

[12] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime measurements in the cloud: observing, analyzing, and reducing variance," *Proceedings of the VLDB Endowment*, vol. 3, pp. 460–471, 2010.

[13] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing," in *CloudComp'09*, 2009, pp. 115–131.

[14] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Security Privacy*, vol. 9, no. 2, pp. 50–57, March 2011.

[15] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: Outsourcing computation without outsourcing control," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security (CCSW '09)*, 2009, pp. 85–90.

[16] O. Adinolfi, R. Cristaldi, L. Coppolino, and L. Romano, "QoS-MONaaS: A Portable Architecture for QoS Monitoring in the Cloud," in *8th International Conference on Signal Image Technology and Internet Based Systems (SITIS'12)*, Nov 2012, pp. 527–532.

[17] D. Slamanig and C. Stingl, "Privacy aspects of ehealth," in *Third International Conference on Availability, Reliability and Security (ARES'08)*, March 2008, pp. 1226–1233.

[18] H. Tawfik, O. Anya, and A. K. Nagar, "Understanding clinical work practices for cross-boundary decision support in e-health," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 530–541, Jul 2012.

[19] M. Masud, M. S. Hossain, and A. Alamri, "Data interoperability and multimedia content management in e-health systems," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1015–1023, Nov 2012.

[20] X. Liang, X. Li, Q. Shen, R. Lu, X. Lin, X. Shen, and W. Zhuang, "Exploiting prediction to enable secure and reliable routing in wireless body area networks," in *Proc. IEEE Conf. Comput. Commun.*, 2012, p. 388396.

[21] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable Secure File Sharing on Untrusted Storage," in *Proceedings of the 2Nd USENIX Conference on File and Storage Technologies (FAST '03)*. USENIX Association, 2003, pp. 29–42.

[22] Z. Zhou, E. Wan, and X. Xu, "2006 online game report," Pacific Epoch Red Innovation Report Series, 2006.

[23] V. Nae, R. Prodan, T. Fahringer, and A. Iosup, "The impact of virtualization on the performance of massively multiplayer online games," in *8th Annual Workshop on Network and Systems Support for Games (NetGames)*, Nov 2009, pp. 1–6.

[24] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *7th High Performance Computing and Simulation Conference, HPCS'09*. IEEE Computer Society, 2009, pp. 1–11.

[25] S. U. K. Dzmitry Kliazovich, Pascal Bouvry, "GreenCloud: A packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1263–1283, 2012.

[26] H. Casanova, A. Legrand, and M. Quinson, "SimGrid: A generic framework for large-scale distributed experiments," in *10th Int. Conf. on Computer Modeling and Simulation, UKSIM' 08*, 2008, pp. 126–131.

[27] A. García-Guirado, R. Fernández-Pascual, and J. M. García, "Virtual-gems: An infrastructure to simulate virtual machines," in *Fifth annual workshop on modeling, benchmarking and simulation, MoBS'09*, 2009, pp. 1–10.

[28] J. J. Dongarra, P. Luszczek, and A. Petitet, "The linpack benchmark: Past, present, and future. concurrency and computation: Practice and experience," *Concurrency and Computation: Practice and Experience*, vol. 15, p. 2003, 2003.

[29] H. Wasserman, A. Hoisie, and O. Lubeck, "Performance and scalability analysis of teraflop-scale parallel architectures using multidimensional wavefront applications," *The International Journal of High Performance Computing Applications*, vol. 14, pp. 330–346, 2000.

[30] Y. H. Jung, L. Carloni, and M. Petracca, "Cloud-aided design for distributed embedded systems," *IEEE Design Test*, vol. 31, no. 3, pp. 32–40, June 2014.

[31] S.-H. Hung, C.-H. Chen, and C.-H. Tu, "Performance evaluation of machine-to-machine (m2m) systems with virtual machines," in *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, Sept 2012, pp. 159–163.