Cloud-based Simulation for Education: An Illustrative Scenario

Rafael Cano-Parra
rafcano@gsic.uva.es

Eduardo Gómez-Sánchez
edugom@tel.uva.es

Miguel L. Bote-Lorenzo
migbot@tel.uva.es

José Antonio González-Martínez
jgonmar@gsic.uva.es

GSIC/EMIC research group, E.T.S. Ingenieros de Telecomunicación, Universidad de Valladolid
Campus Miguel Delibes – Paseo de Belén, 15
Valladolid, Spain

ABSTRACT

Computer simulation is a technique often used in education to study and understand the behavior and characteristics of a real system, applications or platforms. However, in order to achieve reasonable response times for an educational setting, many computational resources are demanded, which often leads to an oversized infrastructure. This paper discusses the use of cloud computing as an enabler of new learning scenarios through the use of Software as a Service (SaaS) applications deployed on top of an Infrastructure as a Service (IaaS) cloud. The Cloud-based Distributed Network Simulation Environment (CloudDNSE) is the proposed SaaS, designed following Service-Oriented Architecture (SOA) and Representational State Transfer (REST) principles, and used in a real educational scenario of Computer Networks. Benefits as scalability, on-demand provisioning or pay-per-use billing model allow to improve the learning design and process, overcoming technological limitations such as lack of computational resources.

Categories and Subject Descriptors

C.2.4 [**Computer-communication Networks**]: Distributed Systems—*cloud computing, distributed applications*; D.2.11 [**Software Engineering**]: Software Architectures—*domain specific architectures, service-oriented architecture (SOA)*; D.2.13 [**Software Engineering**]: Reusable Software—*reusable libraries*; I.6.7 [**Simulation and Modeling**]: Simulation Support Systems—*environments*; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*distributed, parallel*; J.2 [**Physical Sciences and Engineering**]: Computer Applications; K.3.1 [**Computers and Education**]: Computer Uses in Education—*collaborative learning, distance learning*

General Terms

Design, Management, Performance

Keywords

CloudDNSE, cloud computing, education, IaaS, REST, SaaS, simulation, SOA

1. INTRODUCTION

Computer simulation is a technique widely used in education to study the behavior and characteristics of a real model (e.g. a component of a machine or the weather forecast for the next days) [8]. Students can design and run simulations to understand concepts, by modifying parameters to obtain new simulation scenarios. Simulations allow learners to carry out experiments that cannot be performed in a real system, because it is expensive, fragile, unreachable, or simply because it does not exist (e.g. performing a simulation before prototyping an engineering artifact) [7].

Most educational institutions in scientific or engineering disciplines use computer simulations in their laboratory sessions to carry out learning activities. A typical laboratory classroom is composed of a limited number of computers with simulation tools installed, so that a student or a group of them can perform isolated simulations in their assigned computer. Normally, they run several simulations with varying parameters, in order to understand the behavior of the model under study [2].

However, high response times of simulation runs is a well-known problem in education. Simulation times should be reduced so that enough experiments can be run in one laboratory session (e.g. two hours) [2]. Although some simulation models allow for coarse-grain parallelization (i.e. splitting one simulation task in several threads or processes), a reduction of the response time sometimes can only be achieved if enough computational resources (e.g. more cores or processors) are available [7]. Thus, institutions could buy this additional infrastructure so that students enjoy reduced simulation times. This approach, however, has a high cost of ownership of overprovisioned resources that, most likely, will be underused. Further, it may well happen that these new resources are still not enough for other simulation models or environments demanded occasionally [5].

A feasible solution to solve these limitations is to use an existing Software as a Service (SaaS) application in the cloud that enables running simulations with response time constraints, thanks to the promise of virtually infinite resources from the cloud [3, 6]. However, to the best of our knowledge, there is not such a SaaS application. Nevertheless, a new custom SaaS can be deployed on top of an Infrastructure as a Service (IaaS) cloud, as other SaaS applications have done (e.g. Twitter or Dropbox on Amazon's IaaS). Such simulations SaaS could run parallel simulation jobs [5] in multiple virtual machines of the IaaS provider, automatically requesting resource provisioning or releasing as needed to reduce response times while minimizing infrastructure costs. Thanks to the on-demand provisioning and pay-per-use billing model of cloud computing, the educational institution would be able to support its simulation-based learning scenarios without incurring in high ownership costs [3, 9, 11].

In order to illustrate how cloud computing can enable learning activities that make use of simulations with reasonable times and costs, this paper discusses the design and use of a Cloud-based Distributed Network Simulation Environment (CloudDNSE) in a real educational scenario in a course on Computer Networks. The proposed scenario involves a parameter sweep study of a network model, in which many simulations of the same network model must be run with different parameter values in order to understand the dependencies among parameters. The whole simulation task should have a low response time, so that students can design and run the experiment and reflect on the results within the two-hour laboratory session.

The rest of the paper is organized as follows. Section 2 describes the educational context, design and possible implementation technologies of the CloudDNSE application. Then, a discussion of the advantages and limitations of cloud computing related to the proposed CloudDNSE, and its application in other similar scenarios, is presented in Section 3. Finally, Section 4 gives some conclusions and future work.

2. THE CloudDNSE SaaS APPLICATION

The Cloud-based Distributed Network Simulation Environment (CloudDNSE) is the proposed SaaS application to show how cloud computing can achieve reasonable times and costs when it is used in a real scenario of Computer Networks for four two-hour laboratory sessions. This section first describes the CloudDNSE context, pointing out the educational requirements for the application. Then, the design and architecture of the application is explained, and lastly possible selection of technologies for implementation are detailed.

2.1 Application Context

The CloudDNSE application enables the simulation of multiple different Computer Networks scenarios. This section describes a scenario proposed for the Teletraffic & Management course of the Telematics Engineering Degree program at the Universidad de Valladolid. It contains a practical exercise of Computer Networks that is carried out by students in four two-hour laboratory sessions running several simulations to study the behavior and characteristics of a given network scenario.

The proposed learning scenario deals with a basic computer network (a traffic source and a traffic sink, connected by two routers and three links) in which students must analyze the throughput of a TCP connection varying the delay of the central trunk network link and the bitrate of their other links. It requires running a parameter sweep simulation [2] where different values of delay and bitrate parameters are assigned to study its evolution (in particular, the teacher suggests 6 values for the delay and 15 values for the bitrate parameters, yielding a parameter sweep composed of 90 individual simulations).

Students usually work with a laboratory computer[1] to carry out the practical exercise. Each individual simulation uses completely a core of the processor while running, so that all individual simulations must be executed sequentially in each core. If the response time of a typical individual simulation is three minutes, the whole parameter sweep simulation in a two-core processor lasts about 135 minutes, longer than the two hours of one laboratory session. Then, with the current computational resources, the teacher must change the assignment to one that fits in these two hours.

One feasible solution would be to run all the individual simulations in parallel using the resources provided by a cloud infrastructure. Then, various instances of a same simulation tool [5] can run concurrently on different virtual machines of the computational cloud, so that the response time of a parameter sweep simulation can be closer to the response time of a single individual simulation. Even so, a parameter sweep simulation can be run concurrently by several students, requiring many more resources at a time. Therefore, there is a need to balance resource provisioning to reduce response times with cost constrains.

2.2 Application Design

The design of the application followed the Service-Oriented Architecture (SOA) principles, so that high level elements can be reused from or in other designs. However, the architecture of the CloudDNSE (represented in Fig. 1) was proposed considering all the necessary functionality to satisfy the above scenario. Thus, it includes some generic issues like user account management and storage that will not be further commented here. Moreover, each service of the CloudDNSE is designed following the Representational State Transfer (REST) [4] architectural style, offering resources through a web-based unified Application Programming Interface (API) that can be managed basically by four elementary methods (i.e., GET, POST, PUT and DELETE).

The core of the application deals with the management of the parameter sweep simulations. To implement it, a parameter sweep is described in what is termed a *simulation package*, which describes the network model and the possible values of the parameters.

An *orchestration service* receives the request from the user interface to run one such simulation package. It must then generate all the individual simulation scripts, run the individual simulations, and collect and merge the results. Since each individual simulation is independent from the rest, parallelization is achieved by placing them in a job *queue service*. Existing but idle *simulation services* will then take a job from the queue.

Scalability is achieved by defining some rules in a *scalability service*. These rules are expressed in terms of variables measured by a *monitoring service*, and can trigger actions on the cloud infrastructure, like launching or stopping new virtual machines that run a *simulation service* (e.g. "IF

[1]Having a 2007 Intel Xeon two-core processor at 2.8 GHz, and 3 GB of RAM and 6 GB of SWAP.

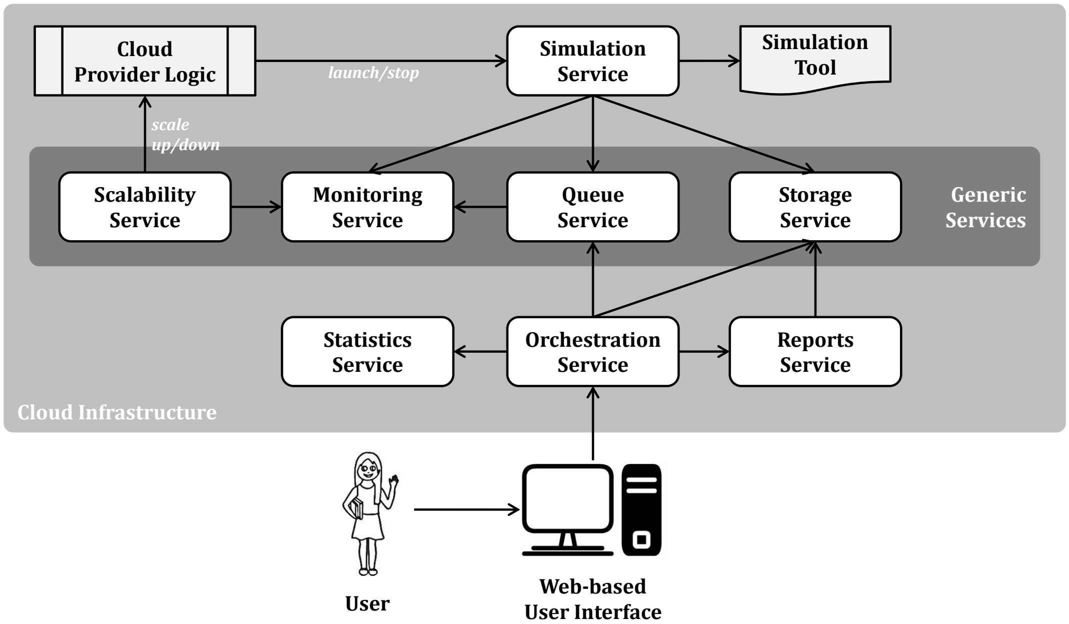

Figure 1: The CloudDNSE architecture, where each service may be executed in one or more virtual machines on the cloud infrastructure, except for the simulation service and tool that need to be run independently in one or more instances of the same virtual machine.

service HAS *metric* GREATER THAN / LESS THAN / EQUAL TO *value* THEN START / STOP *simulation service* WATCHED BY *monitoring service*").

When each *simulation service* ends an individual simulation, it writes the results in the space assigned to the package simulation in a *storage service*, and requests a new individual simulation job. Students can see the results of a simulation and generate different types of graphs. Moreover, they can calculate statistics from parameter values using a *statistics service*. Finally, a *reports service* handles the generation of a package simulation report, which uses all data available in a *storage service*. The resulting report is stored in the previous service.

All these services, and their relationships, are depicted in Fig. 1. The business logic of some of them is specific to the functionality of the CloudDNSE. For example, the *orchestration service* is responsible for managing the simulation environment workflow, and needs services as *statistics*, *reports* and *simulation* to carry out some high computational costly functions in separated virtual machines to achieve better performance (e.g. parallelization of *simulation services*).

On the other hand, the rest of services offer a generic functionality, so that they can be reused in other applications. For example, the *monitoring service* can control other services or the *scalability service* can start or stop other virtual machines that contain a particular service. Also, the *queue service* can receive any computational job, and the *storage service* can contain any file or directory hierarchy.

Some cloud computing providers offer generic services as those described above. An example is Amazon Web Services (AWS) that provides Simple Storage Service (S3)[2] for storage, Simple Queue Service (SQS)[3] to manage queues, CloudWatch[4] to monitoring services, and Auto Scaling[5] to achieve an automatic dynamic scalability. The shaded services of Fig. 1 can be replaced in the CloudDNSE architecture and it would maintain the same functionality. Hence, new applications only require the implementation of dedicated application services.

2.3 Technologies for Implementation

Several technological choices are possible to implement the CloudDNSE. Regarding the deployment in a cloud infrastructure, AWS is the most popular provider if choosing a public cloud. In turn, OpenStack[6] and Eucalyptus[7] are the most probable open-source platforms that could be used to implement a private cloud [10], because they are compatible with the AWS API that enable an eventual transition to a hybrid cloud [12].

As shown in Table 1, both AWS and OpenStack offer their own services that can replace the generic services of the CloudDNSE. Eucalyptus instead offers an API compatible with AWS Auto Scaling and CloudWatch services, but Walrus[8] as its own additional storage service, equivalent to AWS S3. On the contrary, some OpenStack services like Ceilometer[9] and Marconi[10] do not support an equivalent API to the AWS services (in comparison with Heat[11] and

[2]`http://aws.amazon.com/en/s3/`
[3]`http://aws.amazon.com/en/sqs/`

[4]`http://aws.amazon.com/en/cloudwatch/`
[5]`http://aws.amazon.com/en/autoscaling/`
[6]`http://www.openstack.org/`
[7]`http://www.eucalyptus.com/`
[8]`http://www.eucalyptus.com/eucalyptus-cloud/iaas/architecture`
[9]`https://wiki.openstack.org/wiki/Ceilometer`
[10]`https://wiki.openstack.org/wiki/Marconi`
[11]`https://wiki.openstack.org/wiki/Heat`

Features	Amazon Web Services Cloud	Eucalyptus Cloud Middleware	OpenStack Cloud Middleware
Automatic Scalability	AWS Auto Scaling	Integrates the AWS Auto Scaling API	OpenStack Heat (Auto Scaling API)
Control and Management of Metrics	AWS CloudWatch	Integrates the AWS CloudWatch API	OpenStack Ceilometer
Job Queue Management	AWS Simple Queue Service (SQS)	—	OpenStack Marconi
Storage Availability	AWS Simple Storage Service (S3)	Eucalyptus Walrus (S3 API)	OpenStack Object Storage (S3 API)

Table 1: Some services of the most relevant deployments of public and private clouds.

Object Storage[12]), and Eucalyptus does not offer any service or compatible API to be used as an equivalent of AWS SQS.

Communication between services is done using unified APIs designed following the REST principles. A reasonable technology to implement these APIs is RESTlet[13], a REST-based development framework written in Java. Interestingly, the aforementioned cloud infrastructures contain public APIs based on REST that allow the communication between services and the cloud infrastructure.

Finally, ns-3[14] is the simulation tool chosen to be used in simulation services, because it is the preferred simulator by the teacher of the proposed scenario, and allows the description of a simulation job in a script.

3. DISCUSSION

The use of cloud computing has enabled the creation of a SaaS application to solve a new learning scenario that was not feasible before. This way, the CloudDNSE application is viewed by teachers and students as a SaaS application offered by the Universidad de Valladolid, even though the provider itself is offering it on top of an IaaS. Educational institutions could also offer other SaaS such as e-mail or VLEs following a similar approach with the economic appeal of requiring low investments in infrastructure [1].

For example, the application presented here can be deployed completely using AWS services. Under the assumption of 20 groups of two students each one and 90 individual simulations of each parameter sweep simulation with an expected response time of six minutes, the estimated cost for four two-hour sessions would be between 32 € and 210 €, depending on the type of the AWS instance used by simulation services[15]. Furthermore, if a private cloud of an educational institution is used in conjunction with the AWS public cloud in a hybrid cloud, these prices would be reduced. This provides a general idea of how the pay-per-use model can provide significant cost savings, though a more thorough study would include a better estimation of the aggregated computational demands and its time span[16].

Another approach to leverage cloud computing affordances would be setting up a private cloud by an educational institution to be used in courses with occasional load peaks, but with an aggregated demand sustained over time. Cost savings in hardware could be obtained through the consolidation of resources in the private cloud. In summary, public, private or hybrid clouds can bring cost savings depending on whether computational demands are occasional, constant, or sustained with excess peaks.

Additionally, cloud computing brings other benefits such as a high availability, an ubiquitous access over the Internet, or cost savings in power consumption or system management [3]. These benefits may motivate educational institutions to move their infrastructures and services to the cloud or to deploy new ones that enable new learning scenarios with a low cost.

Concerning the design of these new services, the use of SOA and REST principles to design the CloudDNSE application shows the possibility of reusing some services to create new SaaS applications that support similar learning scenarios. In particular for this application, services can be classified in several layers according to their potential reutilization: *application layer* that contains services as orchestration, reports or simulation, which are not easily reusable since they implement the business logic of the particular application; *utility layer* with some generic services as statistics; *persistence layer* integrating storage services; and *scalability layer* that includes services as monitoring, scalability or queue. Then, any application programmer can easily manage the provision of computational resources to meet some performance target, by defining rules in the scalability layer without having to write the code for this. Similary, data can be saved on a virtually infinite storage space thanks to the persistence layer.

Beyond simulation scenarios, cloud computing and an application architecture similar to that of CloudDNSE can enable at a low cost many other educational settings where there are significant computational demands. One example in an engineering or science course would be a MATLAB-based application to calculate some complex equations. The scalability layer can be reused to obtain parallel computational resources, and only the application layer must be

[12] `http://www.openstack.org/software/openstack-storage/`

[13] `http://restlet.org/`

[14] `http://www.nsnam.org/`

[15] This calculation assumes the use of `m1.xlarge` Linux virtual machines for all services, with a minimum price of 0.04 €/hour for spot instances and a maximum price of 0.37 €/hour, the same as on-demand instances. General services are matched with the AWS equivalent ones with an estimated cost of 10.50 €/month, for 100 GB of storage and four custom metrics, four alarms and 2000 requests for monitoring.

[16] Note that Amazon charges by the hour, so costs are minimized if committed resources are intensively used along that hour.

implemented with a new orchestration service and a MATLAB service that will be executed in multiple virtual machines in the cloud infrastructure. Another example in architecture courses would be an AutoCAD-based application to quickly render a 3D building model. It requires both significant processing to do the rendering, and a somewhat large amount of storage to save the plots and the rendered model. Then, both the scalability and persistence layer can be reused, while the application layer must be implemented as the scenario above.

In summary, while services of the application layer usually need to be re-implemented, other services in layers such as scalability or persistence can be reused. Moreover, many of these services can also be reused from the existing cloud infrastructure, either in public clouds (e.g. Amazon S3, SQS, CloudWatch or Auto Scaling) or in private clouds middlewares (e.g. OpenStack Heat, Ceilometer or Marconi).

4. CONCLUSIONS AND FUTURE WORK

Cloud computing enables an easy creation of new educational SaaS to achieve reasonable response times and costs thanks to dynamic and automatic resource scalability. Using a design based on SOA and REST principles allows other applications to reuse some generic services as those related with scalability or persistence, and only a few particular services related with the business logic must be newly reimplemented.

Future work includes the deployment of the CloudDNSE in a hybrid cloud, taking advantage of the API compatibility between OpenStack or Eucalyptus and AWS to get more computing resources at a low cost. Also, a better performance in the dynamic and automatic scalability is needed, for instance using images that are previously configured or using a cloud infrastructure that contains automatic mechanisms to carry out an horizontal and vertical scalability.

5. ACKNOWLEDGMENTS

This research has been partially funded by the Spanish Ministry of Economy and Competitiveness Project TIN2011-28308-C03-02, the Autonomous Government of Castilla and León Project VA293A11-2, and the METIS European LLP Project (531262-LLP-2012-ES-KA3-KA3MP), as well as by the Universidad de Valladolid predoctoral fellowship FPI-UVa 2012. The authors thank as well the rest of the GSIC/EMIC research team for their contributions.

6. REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, Apr. 2010.

[2] M. L. Bote-Lorenzo, J. I. Asensio-Pérez, E. Gómez-Sánchez, G. Vega-Gorgojo, and C. Alario-Hoyos. A Grid Service-Based Distributed Network Simulation Environment for Computer Networks Education. *Computer Applications in Engineering Education*, 20(4):654–665, Dec. 2012.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. 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, June 2009.

[4] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures.* PhD thesis, University of California, Irvine, Irvine, CA, USA, 2000.

[5] R. M. Fujimoto, K. Perumalla, A. Park, H. Wu, M. H. Ammar, and G. F. Riley. Large-Scale Network Simulation: How Big? How Fast? In *Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS 2003)*, pages 116–123, Orlando, FL, USA, Oct. 2003. IEEE Press.

[6] P. Mell and T. Grance. The NIST Definition of Cloud Computing. Technical Report NIST Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, MD, USA, Sept. 2011.

[7] H. Rajaei and J. Wappelhorst. Clouds & Grids: A Network and Simulation Perspective. In *Proceedings of the 14th Communications and Networking Symposium (CNS 2011)*, pages 143–150, Boston, MA, USA, Apr. 2011. Society for Computer Simulation International.

[8] N. Rutten, W. R. van Joolingen, and J. T. van der Veen. The learning effects of computer simulations in science education. *Computers & Education*, 58(1):136–153, Jan. 2012.

[9] L. M. Vaquero. EduCloud: PaaS versus IaaS Cloud Usage for an Advanced Computer Science Course. *IEEE Transactions on Education*, 54(4):590–598, Nov. 2011.

[10] G. von Laszewski, J. Diaz, F. Wang, and G. C. Fox. Comparison of Multiple Cloud Frameworks. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing (CLOUD 2012)*, pages 734–741, Honolulu, Hawaii, June 2012. IEEE Press.

[11] W. Voorsluys, J. Broberg, and R. Buyya. Introduction to Cloud Computing. In R. Buyya, J. Broberg, and A. Goscinski, editors, *Cloud Computing: Principles and Paradigms*, chapter 1, pages 3–37. John Wiley & Sons, Inc., Hoboken, NJ, USA, Jan. 2011.

[12] Z. Zhang, C. Wu, and D. W. Cheung. A Survey on Cloud Interoperability: Taxonomies, Standards, and Practice. *ACM SIGMETRICS Performance Evaluation Review*, 40(4):13–22, Apr. 2013.