Enhancing QoE based on Machine Learning in Cloud infrastructure

Fatma Louati
enicarthage
Oumayma Jouini
École Nationale d'Ingénieurs de Tunis: Ecole Nationale d'Ingenieurs de Tunis
kaouthar sethom (k_sethombr@yahoo.fr)
supcom

Research Article

Keywords: Cloud, QoS, QoE, machine learning

Posted Date: March 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-272172/v1

Abstract

The cloud computing paradigm has recently attracted many industries and academic attention. It provides network access on demand and offers applications, platforms, or access to a shared pool of hardware and software resources. For traditional deployment, the user reserves the most required resources. However, this system does not guarantee an optimal use of resources and is not profitable for users. The characteristic feature of the elasticity of the cloud Computing gives the Cloud the ability to perform an automatic up / down scale resources proportional to demand. However, classical deployment only considers the use of resources based on alarm, and does not consider the quality perceived by the end user. The aim of this paper is to set up a private IAAS Cloud infrastructure and complete it by supervision tools so we could optimize the management of the cloud elasticity based on users' point of view or QoE. We have also used a Machine learning algorithm to predict the load charge of the physical machines of the cloud so that providers could manage efficiently their data centers.

Introduction

With the number of suppliers present on the market, to be competitive, the cloud providers aim to satisfy the users; they must then manage, not only the quality of the service according to a previously established SLA, but must also take into consideration the perception of the user of the service, called Quality of Experience (QoE). On the other hand, with the development of green data center managers, improving energy efficiency and environmental footprint is one of the significant challenging aspect of green cloud. Minimizing energy consumption (computing resources, cooling systems) can significantly reduce the amount of energy bills and then increases the provider's profit. It is within this context that our study belongs, the objective is to deploy a service architecture on an IAAS cloud platform based on OpenStack, to implement the monitoring module which is based on the Ceilometer telemetry module and other supervision tools such as SNMP and PRTG to manage the different physical resources used and monitor them. This will optimize the use of resources by avoiding the under / over use of resources in the Cloud by respecting some QoE on one hand and put in place a mechanism allowing the cloud provider to minimize its energy consumption while respecting a certain SLA on the other hand. In this work, we propose to study the automatic scaling by respecting some QoE and the prediction of the load of a cloud to optimize the distribution of the servers to minimize the energy consumption.

Various research works have been carried out to propose approaches that enables a cloud infrastructure to automatically and dynamically scale-up or scale-down. Some of them deal with resource allocation problem and virtual machine (VM) management to achieve cost savings and that is from better utilization to achieve cost savings and that is from better utilization of computing resources. In [1], cloud resource allocation is considered alongside VM's placement and migration based on VM's CPU, memory, storage, network bandwidth along with resource contention. In [2], Relevant threshold values of resource usage are considered to trigger scaling. In [3], The number of concurrent users and the number of active connections are used to implement the concept of dynamic scaling of resources. The research work described in [4] proposes an auto-scaling mechanism based on budget constraints and job execution

deadline. in [5], A pattern-based prediction algorithm that handle sudden appearance/ disappearance of traffic is used to introduce an auto-scaling mechanism. In the above-mentioned research work, resources are allocated to achieve optimum results towards improving Quality of Service. But no one has focus on QoE.

Qoe And Cloud Computing

A. QoS versus QoE

The QoS was defined by the ITU as [6] "*Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service*". Another definition is that QoS is a set of techniques that offers to applications the service they need, from end to end. The goal of quality of service is to provide priority to networks, including dedicated bandwidth, controlled jitter, low latency and improved loss characteristics. So that service providers could offer the best possible service to their clients.

In other words, Quality of Service, is the prioritization of some services using the network, such as VoIP telephony, messaging, video conferencing or video surveillance. It allows to classify the different types of applications according to their importance, in order to assign more or less bandwidth, and thus optimize the network.

To evaluate the quality on offered service, QoS was the only metric used in the past. However, since it relies only on technical measures related to network performance, it doesn't really reflect users' assessments. For the same service two users could have different appreciations. This is due to the performance of the used device, the expectations or the feeling of the users at that moment, their social and intellectual environment and their emotional state. Thus, a new metric appears named QoE or Quality of experience.

The International Telecommunication Union (ITU) defines the quality of experience in ITU-T P10/G100 [7] "*the overall acceptability of an application or service, as perceived subjectively by the end-user*". The European Telecommunications Standards Institute (ETSI) [8] defines the QoE as "*the performance of a user when using what is presented by a communication service or application user interface*".

By definition, quality of service and quality of experience are two performance indicators for a service but from different ways. For QoS – Quality of Service it takes into account the network characteristics/behavior Performance guarantees given by network provider based on measurements. Regarding QoE – Quality of Experience: it considers the impact of network behavior on end users as some imperfections may go unnoticed and may render application useless. It is not captured by network measurements. The table 1 presents a comparison between QoS and QoE.

Table 1: QoS and QoE comparision

All the definitions presented above consider that QoE is a subjective measurement provided by the end user that reflects the degree of satisfaction of the used service, While QoS is an objective measure provided by clear measurement methods based on indicators. This type of evaluation incorporates the end-to-end system and especially the user's appreciation. This makes its meaning more complete but also exposes it to several factors that may affect the results.

B. QoE measurment factors

QoE is a multidimensional measurement which can be affected by a variety of factors. By definition, the factors that can affect the QoE are: "Any characteristic of a user, system, service, application, or context whose actual state or setting may have influence on the Quality of Experience for the user". Services and applications, the human experience can be influenced by various factors that have an impact on QoE. In this part, we define the general factors that can alter the QoE and the specific factors to our context among them. Factors that may have an impact on QoE can be classified into 4 categories. They are shown is the figure 1:

2.2.3.1 The context factor

As mentioned in the previous section, QoE represents the end-user's perception of quality. For example, quality perception of a multimedia service depends strongly on the viewing context in which consumption takes place. The viewing environment (physical setting) has a considerable influence as it determines lighting conditions, viewing distances, screen quality…

Six categories from different contexts are defined:

- Physical context (location and space)
- Temporal context (time of day, frequency of use, . . .)
- Social context (inter-personal relations during experience)
- Economic context
- Task context (multitasking, interruptions, task type)
- Technical and information context (relationship between systems)

According to K. Brunnström et al. "*Context Influence Factors (CIFs) are factors that embrace any situational property to describe the user's environment*".[10]

2.2.3.2 The user or human factor

A Human Influence Factor (HIF) is any variant or invariant property or characteristic of a human user [10]. The characteristic can describe the demographic and socioeconomic back-ground, the physical and mental constitution, or the user's emotional state. "QoE is how the user feels about how an application or service was delivered, relative to their requirements" [11]. This is strongly influenced by the user's internal

states and predispositions. Common examples of human factors include not only gender, age, level of expertise, but also the psychological situation when using the service. Indeed, the properties related to the emotional and mental constitution of the user can play a major role in the final assessment of the user. Because of their complexity and lack of empirical evidence, we still do not know how human factors affect QoE.

2.2.3.3 The system factor

The properties of the technical system directly influence QoE. The term refers to both the entire chain of communication between the service provider and the end-user (eg network, terminal equipment), and technical characteristics of the service provided.

We can therefore classify them as sub-factors of system as follows:

- Media factors: When dealing with multimedia content, the configuration of the source of media, such as encoding and compression settings, the rate sampling, the resolution of the scene, the frequency of the images, has a high impact on perceived overall quality.
- Network factors: These factors refer to the transmission of data over a network and are closely related to network QoS parameters, including packet loss, delay, jitter, bandwidth, and error rate. The effect of these parameters on the perceived quality depends mainly on the type of multimedia application but evolves with time and / or with the location of the user.
- Device factor: User device performance may affect the whole user experience. These factors include, for example, the resolution display, colors, brightness. For example, if a high-quality image and high resolution is displayed on a low-resolution screen with few colors, most of the original intent of the image may be lost.

2.2.3.4 The content factor

For different types of content, there are different requirements system. For video or gaming, for example, the amount of movement and the bandwidth audio can influence the overall QoE. The content itself and its type of influence strongly the overall QoE of the system because for different content characteristics, different system properties are needed.

C. QoE evaluation models

There are various approaches to quantifying the QoE of a provided service. These approaches are classified according to the perceived quality assessment, if evaluated directly by humans or automatically by technical factors. In the first case, specific evaluation processes are used, called subjective tests, while in the second case, mathematical formulas or algorithms are exploited, called models goals. There is a third category of evaluation of QoE called “hybrid method”; based on the use of

automatic goal estimator, relying however on the subjective tests available. The next figure 2 presents the main classification of the QoE models:

Subjective methods Subjective tests are usually based on controlled experiments with human participants who directly evaluate their experience with an application or service. Different techniques can be used for subjective evaluation. For example, users can rate their experience using an absolute rating scale or they can compare images and / or videos by specifying which is better. In all cases, the results are based on users' opinions, past experiences, expectations, user perception, judgment and description skills, etc. One of the most popular subjective assessment methods is "Mean Opinion Score "(MOS) [13]. This method is based on laboratory tests under good conditions. specific, detailed by the ITU in [14]. The quality is then evaluated by the users based on feedback surveys of the experience lived on a qualitative scale (bad, poor, fair, good and excellent) numbered from (1) to (5) as shown in figure 3. Then, the MOS is calculated as the arithmetic mean of all the individual scores mentioned by the test subjects. The QoE is then attributed to this statistical value.

2.3.2 Objective methods

According to ITU, the principle of objective quality assessment is the estimation of subjective quality only from the measurement of objective quality or indices. Depending on the type of input data used for quality of experience assessment the objective method was classified:

- Media layer models: These models use the multimedia signal to calculate the quality of experience (QoE), following comparisons and do not require any information about the system being tested.
- Packet layer models: These models predict QoE only from packet header information and do not have access to the multimedia signals.
- Parametric planning models: These models use parameters of quality planning for networks and terminals to predict QoE. They require prior knowledge of the tested system.

Objective measures can be classified according to the availability of the original signal. Three major model approaches have been identified:

- Completed reference The QoE Estimation Algorithm requires access to both the reference input data and the degraded output data
- No reference The QoE estimation algorithm requires only access to the degraded output data.
- Reduced reference The QoE estimation algorithm requires access to degraded output data and some features from the original signal that the quality assessment system will use as secondary information to help evaluating the quality of the degraded output data.

2.3.3 Hybrid methods

The third type of QoE assessment method is a hybrid model that is located between the two categories subjective and objective. It works as a quality estimator automatic and objective, relying however on the scores resulting from the subjective tests carried out previously. These hybrid methods are based on learning tools called Machine Learning (ML), and they use subjective test scores like input parameters to form a QoE model.

This model then matches the network parameters (for example, packet loss rate, delay, jitter, etc.) to MOS score values. This model offers the possibility of predicting and / or estimating quality in time real.

This type of solution, lying between the subjective and objective methods, presents a significant advantage in the field of prediction and estimation of quality experience but remains very complex to implement and the learning stage is very long and hardworking. Furthermore, the learning stage of the neural network needs a huge amount of data, but in our case, we can't use hybrid models because of lack of data.

D. QoE in the Cloud

A cloud computing environment must be elastically scalable; in other word it must have the ability to flexibly expand as the offered load and the business demands change. However, this feature requires the development of a diverse set of algorithms, like those outlined below. The study of Elastic Scalability and QoE Assessment for Cloud Services are prerequisites for the construction of an intelligent QoE Management and Control mechanism for the cloud resources. Various research works have been carried out dealing with the resource allocation problem and VM management to achieve better utilization of computing resources while avoiding overload situations considering QoE. Many research works concentrated on measuring the performance of cloud computing through measuring parameters such as availability, reliability, scalability, response time. Table 2 shows a literature review of different metrics related to IAAS cloud services.

In [15], cloud resource allocation is considered alongside dynamic resource provisioning using feedback control mechanism on the infrastructure level performance metrics. In [16], the proposed approach considers dynamic service level agreement (SLA). To improve users' perceived QoE, various studies have come up with metrics, which are directly related to the performance of services such as video streaming. Mean Opinion Score (MOS) that evaluate the user QoE is calculated in [13] as metrics consisting of network bandwidth,video-bit rate, Round Trip Time (RTT), page load times and video interruptions.[17]

S. Dutta et al. inn [18] a novel scaling method that closely considers users' QoE, in fact the solution has considered QoE's feedback as a criterion to scale up/down cloud resources. the proposed solution is to build a QoE-aware resource management of virtual instances; to automatically provision and scale network services (NS) in an elastic way. H. Qian et al. in [19] firstly identifies interactions among the cloud entities and afterwards evaluates the QoE for the End-Users in this complicated environment. Work in [20] studied host reliability issues, from the perspective of the End-Users. IN [21] proposes a three-step

approach to map SLA and QoS requirements of business processes to cloud infrastructures. [22] considers QoE in the cloud with power management issues, since it studies a service cloud environment with mobile devices. Emmanouil Kafetzakis et al. in [23], proposes a unified QoE-aware management framework, directly targeting to cloud computing environments. In [24] authors, proposes a methodology to estimates the QoE from the end-to-end response time and adjusts the estimated score according to the evaluation context. Sunny Dutta et al. proposes in [25] an approach that enables a cloud-infrastructure to automatically and dynamically scale-up or scale-down resources of a virtualized environment aiming for efficient resource utilization and improved quality of experience within the ETSI NFV MANO framework for cloud-based 5G mobile systems. Some companies like Infovista [26] or Compuware [27] offer proprietary, closed assessment solutions for monitoring quality at an IaaS level. W. Cai et al. [28] studied the popular cloud vendors for gaming applications. Though gaming applications involve video streaming, the Quality of Experience (QoE) for gaming are very different from the QoE of video streaming service. Besides, both works focus on the infrastructure services such as computing, storage and networking.

Proposed Architecture

Since in this work, we aim to optimize the use of resources by respecting a certain QoE. Our optimization should take into consideration two point of views, the first is related to the cloud users and his allocated resources, the second is in relation with the cloud provider's servers (suspending or using all servers or allocating resources from another competitor if necessary).

We started with conducting subjective tests, then the deployment of the Cloud OpenStack platform and finally, we measured the current instances used and available resources then we try to scale up or down the system according to an optimization algorithm.

The architecture consists of an Openstack plateform with two key added entities that are the the QoE estimation entity and the orchestration entity. The first one is responsible on evaluating automatically current user QoE so that the orchestrator ca scale up or down the user's instance to better deserve him. The cloud platform we have chosen is an IAAS, that's why the orchestrator can only manipulate parameters related to the physical characteristics of the instances. It can only scale up/down the flavors (VCPU, RAM, Disk).

The architecture shown in Figure 4 illustrates the depoyed plateform.

A. The subjective tests

The type of application will impact the users' MOS, because gamer and developer will need instances with higher configurations. And a standard user may rate a minimalist configuration as excellent, while it will be rated as bad by a gamer.

In our work, we will consider 4 type of users profiles:

- C1 : Gamer
- C2 : Developer
- C3 : Video 4K/HD
- C4 : Standards

In order to realize the survey which allowed us to determine the score MOS given by the users, we used Google Forms. Various configurations were presented to different categories

of users and they rated the related instance according to their appreciation.

Given the different answers, for every user profile category, we calculated the average MOS. We have established rules to associate a MOS to a certain configuration knowing the category to which the user belong. We used Python 3 to implement the MOS calculation algorithm.

1) Gamer

According to [29] There are three classes of gamers: “Omnipresent” (e.g. real-time strategy games), “Third-Person Avatar” (e.g. role-play games), and “First Person Avatar” (e.g. First-Person Shooters).[30]

Table 3 : Games requirements

The table 3 shows 3 different games each one belongs to a different class. Then we asked some users to rate their appreciation of these games using different configuration of the cloud instances. Table 4 summarizes the average MOS scores assigned by users to games for different configurations.

Table 4: MOS survey gamer

2) Developer

In this part, we asked developers to rate their assessments of instances with different configurations for their daily needs. Table 5 summarizes the average MOS scores assigned by developers.

Table 5: MOS scoring by developer

3) Video 4K/HD

In this section, the third category is considered. The users that use 4k/HD Streaming video rated the instances, we will not consider the used device, we assumed that all the used devices have high-resolution screens. The table 5 summarizes MOS scores assigned by users.

Table 6: MOS survey video users

4) Standard users

Finally, simple users will rate the used instances for navigation, word processing. . . The results are shown in table 6.

Table 7: MOS survey standard users

In our case, we will just consider the working days during the winter, but for future work we will add: Working days/ Weekend and Seasons (summer / autumn / spring / winter).

Table 6: Percentage usage per contract

The table 6 resumes, the mean percentage presence of each category of contract and the mean resources usage of the cloud during the day.

B. The QoE estimator

In the Classic Cloud deployment, users will reserve certain resource pools to be allocated to their instances. However, the resources reserved will not always be used at their fair value. Indeed, the resources will either be over used or under-used.

Our goal is to develop a system for estimating the quality of experience for cloud users to better deserve them and optimize resources usage.

We used Ceilometer for Cloud ressources measurment. Ceilometer is the monitoring module that applies to the platform generated by OpenStack. The collected data can be sent to different targets:

1. Gnocchi: is developed to capture measurement data in a format time series to optimize storage and queries.
2. Aodh: is the alarm service that sends alerts when the rules set by the user are not respected.
3. Panko: is used for event storage designed to capture document data such as logs and
4. system event actions.

We hereafter propose a method for estimating the quality of experience (QoE) by specifying the most influential parameters. This step began with a selection of the system factors most affecting the QoE perceived by the users. Then a survey was realized by varying these parameters to have a real score. These parameters are subsequently used to estimate the QoE without using the ratings assigned by users each time.

To estimate user QoE, our algorithm will consider various parameters as shown in Figure 5 :

- User profile : Gamer, developper, video or standard

- Reserved and used cloud instance resources collected by Celiometer.

A matching is then made with subjective tests results (Table 4 to 7) to estimate current user QoE.

C. Elasticity management by the orchestrator

The capability of a system to automatically scale up or down in proportionate with demand is known as autoscaling. Autoscaling is essential for availability and optimal usage of

resources. Cloud service specify metrics to be observed, their threshold value and alarms.

Whenever observed metric value crosses a threshold value, alarms are raised and either new resources are provisioned (scale up) or currently provisioned resources are released (scale down) based on scaling policy.

In our case, we add a new alarm called MOS for autoscaling process (Figure 8). The MOS alarm ensures that we can automatically scale when the estimated QoE is under or over a certain thershold compared to user's need (Figure 9). Moreover, the algorithm of scaling up or down will notice the Orchestrator Heat when it's it necessary to do a scaling.

We decide to use recurrent neural networks (RNN), to forecast the load of the cloud (Total ressource usage CPU) so it could manage its resources. Thanks to our additional tools (PRTG and SNMP), we have the statistical usage of our physical machine. We extracted a CSV file that contain the load change of our cloud usage for two months. Our data are time series data, that is why we orient our choice for the machine learning algorithm to Long Short Term Memory "LSTM". It is a variant of the famous Recurrent Neural Networks (RNN). LSTM was designed to model temporal sequences more accurately than conventional RNNs.

The LSTM learns to keep only pertinent information to make predictions. This is achieved during the retropropagation (training phase). Figure 7 represents the structure of LSTM blocks. We put a sequence of several cells to form a chain. The number of cells in a network depends on the input complexity. Typically, for our time series, 4 to 6 cells are sufficient to give a good performance score.

The key to LSTM is the cell state (Ct), which enables the information to flow along it unchanged. The cell state is regulated by three gates to optionally let information through. The first gate is called the forget gate, controlling which elements of the cell state vector Ct−1 will be forgotten. Following that, the input gate decides which value to be updated Finally, the output gate decides which to be output by a sigmoid layer.

A data set containing 2 months of history of load is used with a step of 1 hour, to predict the load of the infrastructure for the next day to determine what action to take and when. Then we divided our sets in

two: a learning set (5/6) and a test set (1/6). The RNN used the learning set to learn how the total charge of the cloud evolve over the time, and the last set to validate its predictions.

Before starting the learning phase of the LSTM, we have to pass through a phase of data exploration. The goal is to become familiar with the used variables. What are the variables ? The different values that it can takes? Dealing with the case of missing values. Deleting with duplicates. Treat numeric values on one side and categorical values from another. For our variables, we only have numerical data that resumes the total cloud usage per day. Our files are with a step of one hour.

We used for that a open source implementation of LSTM using Anaconda navigator as python distribution, Jupyter notebook for the code. Several libraries have been used as

Keras, Panda. And python 3 for the implemented code.

The next figure 8 shows the input/output of the elasticity algorithm implemented inside the orchestrator.

The figure 9 shows a part of the scaling algorithm.

Performance Evaluation

A. Evaluation of estimated MOS

For the MOS calculation, in addition to agreement, we must consider for the Gamer user , the class of the game (Table 3). In the example above, we just consider the first agreement with the second game. To validate our approach, we compared the MOS with the estimated MOS. The figure 10 bellow represents both the subjective MOS of a gamer who plays a game of the second class and the estimated MOS corresponding.

B. Evaluation of prediction

The figure 11 below shows the results of the validation step, the predicted values are in orange and the real one are in blue. We noticed that the predicted values are very close to the real values. This figure shows the load evolution for 11 days.

We can see the prediction of the evolution of the load for one day. The following figure 12 shows it.

In this way and given the past evolution of a cloud provider and depending on the number of users and their profiles, the cloud provider can estimate the load of its data center and optimize it by minimizing for example the energy cost, while guaranteeing QoS according to SLA and QOE.

Conclusion

Cloud providers face new challenges to provide a higher level of SLA (Availability, Security, performance ...) respecting the quality perceived by the end user “QoE”. In this paper, we proposed a solution to maintain the user experience at the level predefined by the SLA. This solution detects the degradation of the QoE, and tries to correct it by performing a scale up. It also detects over dimensioning of instances and makes it possible to optimize the allocation of resources by performing a scale down. The prediction of the load of the network was realized based on the measurements returned by SNMP and PRTG by training a neuronal network on these data. The presented work opens several prospects for improvement and extension. Indeed, the measurement of the proposed QoE only includes the system factors, without considering the human and context factor.

Declarations

*The authors did not receive support from any organization for the submitted work.

* The authors declare that they have no conflicts of interest.

* The datasets generated and/or analysed during the current study are available from the corresponding author on reasonable request.

*The code generated during the current study is available from the corresponding author on request.

References

1. "Cloud Computing", Wikipédia http://en.wikipedia.org/wiki/Cloudcomputing:(10/08/2020)
2. C. Lim, S. Babu, J. S. Chase, and S. S. Parekh, “Automated Control in Cloud Computing: Challenges and Opportunities,” in Proc. 1st workshop on Automated Control for Datacenters and Clouds, Chicago, Illinois, Jun. 2009
3. C. Chieu, A. Mohindra, A. A. Karve, A. Segal, “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment,” in Proc. IEEE ICEBE Int’lConf. e-business Eng. , Macau, China, Oct. 2009
4. Mao, J. Li, M. Humphrey, “Cloud Auto-scaling with Deadline and Budget Constraints,” in Proc. 11th IEEE/ACM Int’l Conf. Grid Computing, Brussels, Belgium, Oct. 2010
5. Bao, Z. Lu, J. Wu, S. Zhang, Y. Zhong, “Implementing a Novel Load-aware Auto Scale Scheme for Private Cloud Resource Management Platform”, in Proc. IEEE Network Operations and Manage. Symp.(NOMS), Krakow, Poland, May 2014.
6. https://www.itu.int/rec/T-REC-E.800-200809-I/fr.(17/10/2020)
7. https://www.itu.int/rec/T-REC-P.10-200607-I/en (18/09/2020)
8. http://standards.globalspec.com/std/1028189/etsi-eg-202-534. (18/09/2020)
9. Michal Ries et al.,” QoE Evaluation of High-Definition IPTV services”, FTW Forschungszentrum Telekommunikation Wien GmbH Donau-City-Straße 1/3, A-1220 Vienna, Austria,20127

10. http://www.qualinet.eu/index.php?option=comcontent(22/09/2018)
11. Corrie, H. Wong, T. Zimmerman, S. Marsh, A.S. Patrick, J. Singer, B. Emond, S. Noël, “Towardsquality of experience in advanced collaborative environments,” in Proc. of 3rd Annual Workshop on Advanced Collaborative Environments, Seattle, USA, Jun., 2003
12. Michal Ries1, Peter Froehlich1, Raimund Schatz, “ QoE Evaluation of High-Definition IPTV services”in FTW Forschungszentrum Telekommunikation Wien GmbH Donau-City-Straße 1/3, A-1220 Vienna, Austria
13. ITU-T Recommendation P.800.1, “Mean Opinion Score (MOS) terminology,” 2003.
14. Somani, P. Khandelwal, and K. Phatnani, “VUPIC Virtual Machine Usage Based Placement in IaaS Cloud,” arXiv preprint arXiv:1212.0085 (2012).
15. Cai et al., “A survey on cloud gaming: Future of computer games,” IEEE Access, 2016.
16. Kalyanakrishnan, R. K. Iyer, and J. Patel, “Reliability of InternetHosts A Case Study from the End User’s Perspective,” in Proceedingsof the 6th International Conference on Computer Communications andNetworks, 1997.
17. Garg, S.K., S. Versteeg, and R. Buyya, A framework for ranking of cloud computing services. Future Generation Computer Systems, 2013. 29(4): p. 1012-1023.
18. Bruneo, D., A stochastic model to investigate data center performance and qos in iaas cloud computing systems. IEEE Transactions on Parallel and distributed Systems, 2014. 25.
19. Saravanan, M.K. and M.L. Kantham, An enhanced QoS Architecture based Framework for Ranking of Cloud Services. 2013
20. Shawky, D.M. and A.F. Ali. Defining a measure of cloud computing elasticity in Systems and Computer Science (ICSCS), 2012 1st International Conference on. 2012. IEEE.
21. Islam, S., et al. How a consumer can measure elasticity for cloud platforms. WOSP/SIPEW international conference on Performance Engineering. 2012. ACM.
22. http://citeseerx.ist.psu.edu/viewdoc/downloadtype=pdf (25/11/2020)
23. Emmanouil Kafetzakis, et al., "QoE4CLOUD: A QoE-driven Multidimensional Framework for Cloud Environments", 2016
24. Sunny Dutta1, Tarik Taleb, “QoE-aware Elasticity Support in Cloud-Native 5G Systems” ,IEEE ICC 2016
25. Jarschel, D. Schlosser, S. Scheuring, and T. Hossfeld, “An Evaluationof QoE in Cloud Gaming Based on Subjective Tests,” in InnovativeMobile and Internet Services in Ubiquitous Computing (IMIS), July 22011, pp. 330 –335
26. http://www.infovista.com/solutions (8/11/2020
27. http://www.compuware.com/application-performance-management/cloud-computingsolutions.html(8/11/2020)
28. Jiang, S. Sun, V. Sekar, and H. Zhang, “Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation.” in ACM NSDI, 2017.

29. Islam, J. Keung, K. Lee, and A. Liu, “Empirical prediction models for adaptive resource provisioning in the cloud,” Future Generation Computer Systems, vol. 28,no. 1, pp. 155–162, 2012
30. Claypool, K. Claypool Latency and player actions in online gamesCommunications of the ACM, 49 (11) (2006), p. 45

Tables

Due to technical limitations, tables are only available within the manuscript file.

Figures

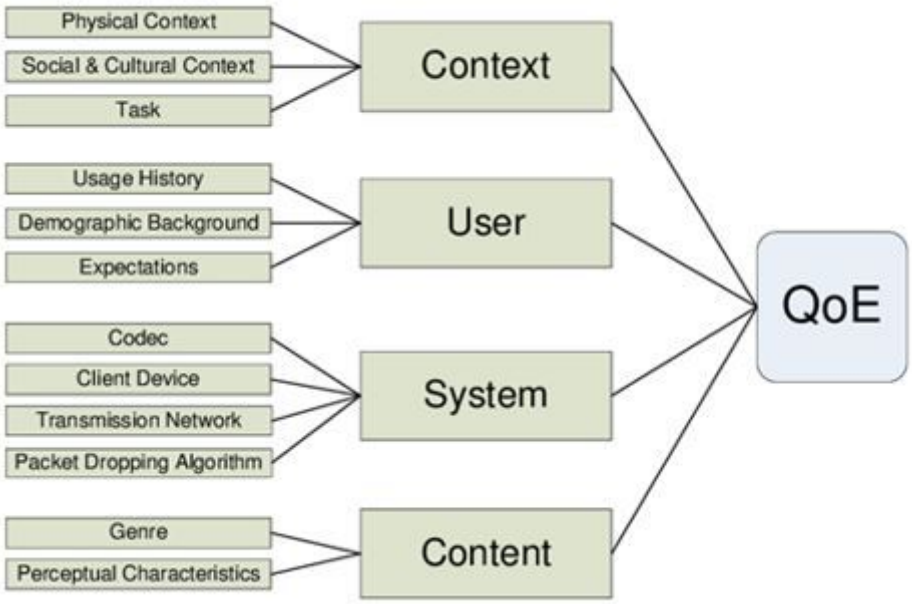

Figure 1

QoE influence factors [9]

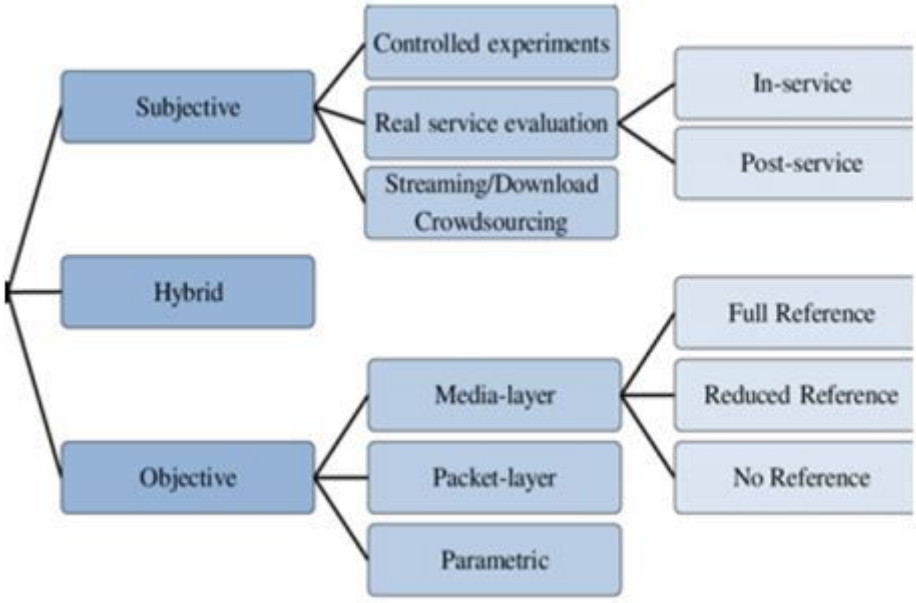

Figure 2

Classification of QoE modeling approaches [12]

Figure 3

MOS score

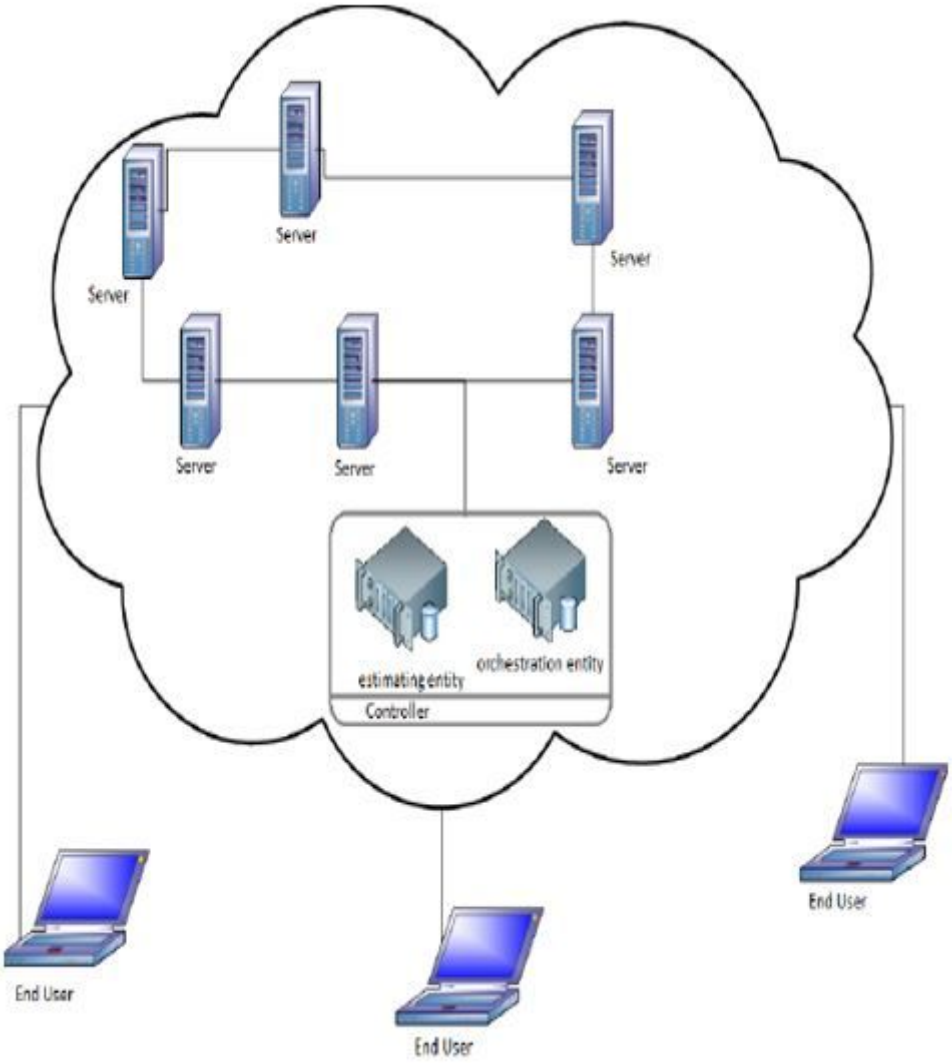

Figure 4

Architecture of estimating QoE in the cloud

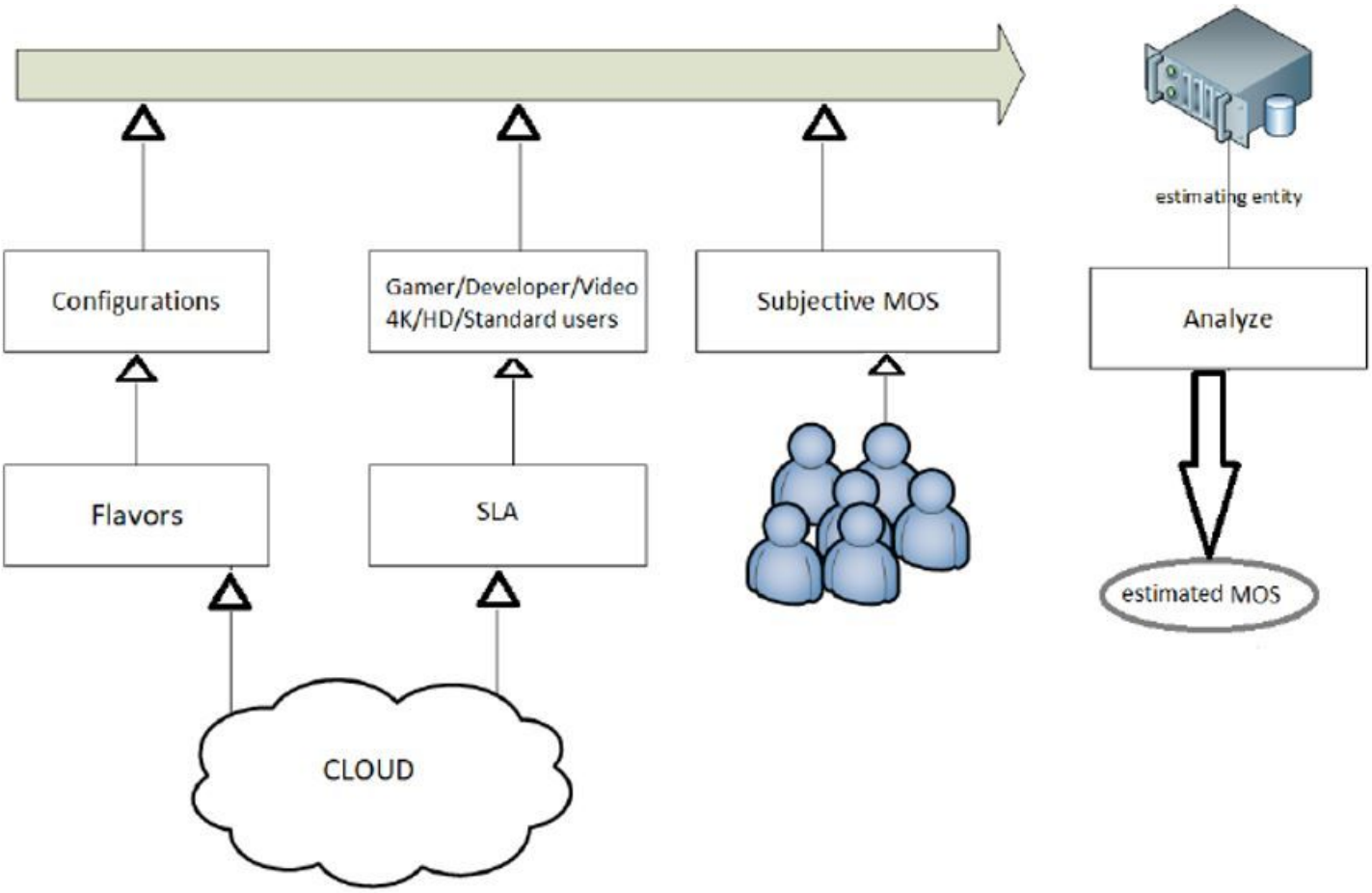

Figure 5

Collecting information process

```
If C1 {
   If (VCPU==1 & RAM <=4 & Disk <= 40 & G2):
      MOS = 1
   Elif (VCPU>=1 &VCPU<4 & RAM <=8 & Disk <= 120 & G2):
      MOS = 3
   Elif (VCPU==4 & (RAM >=4 | RAM <=6) & Disk <= 256 & G2):
      MOS = 4
   Elif (VCPU>=4 & RAM > 6 & Disk <= 256 & G2):
      MOS = 5
   Else:
      MOS = 2.5

}
Else if C2
....
```

Figure 6

MOS calculation algorithm

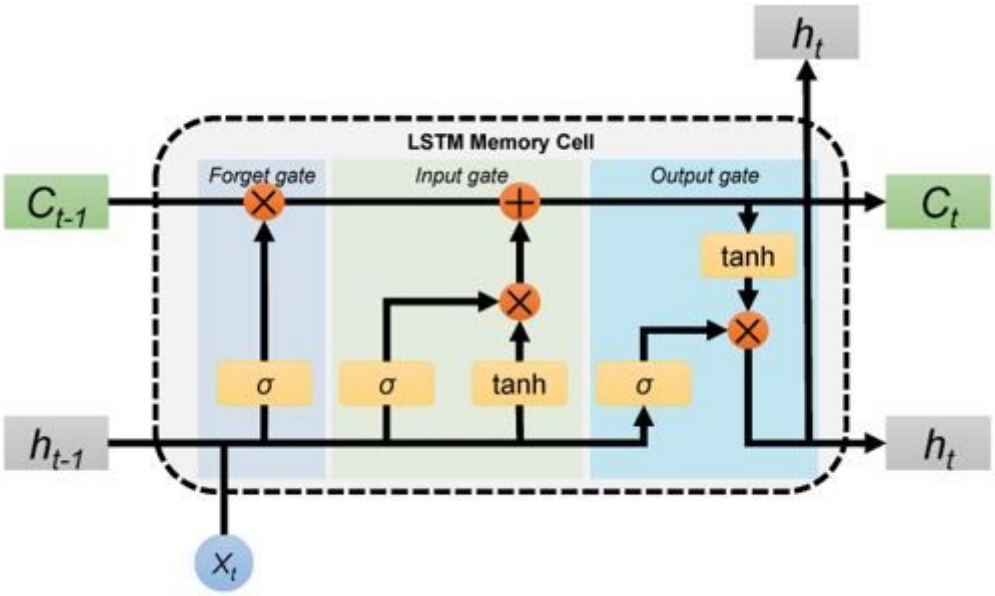

Figure 7

The architecture of LSTM cell

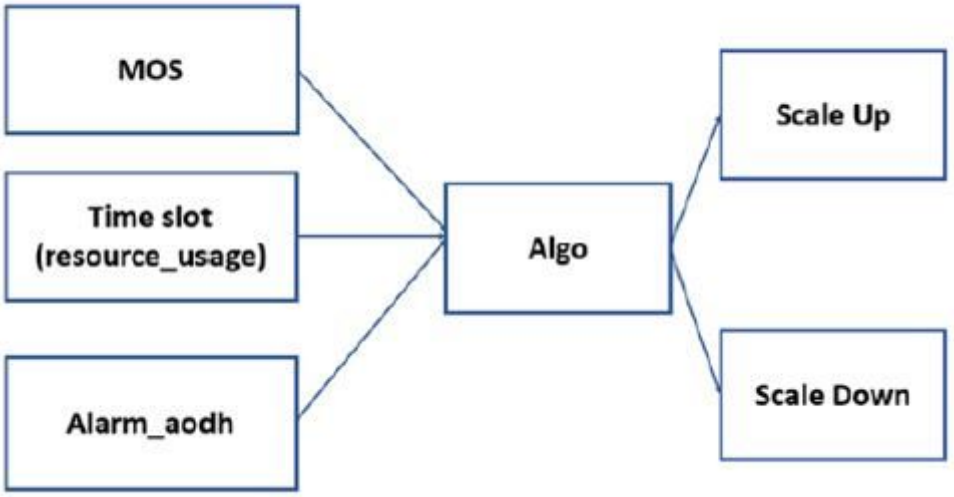

Figure 8

Input output elasticity algorithm

```
Scale Up/Scale Down
   If (MOS < 3):
      Scale_Up()
   Elif (MOS == 3 & res_usg >= 70):
      Scale_Up()
   Elif (MOS >3 && res_usg <= 70 && Alarm_aodh):
       Scale_Up()
   Elif (MOS >3 && res_usg <70):
      Scale_Down()
   Elif (Alarm_aodh):
      Scale_Up()
....
```

Figure 9

Scaling algorithm

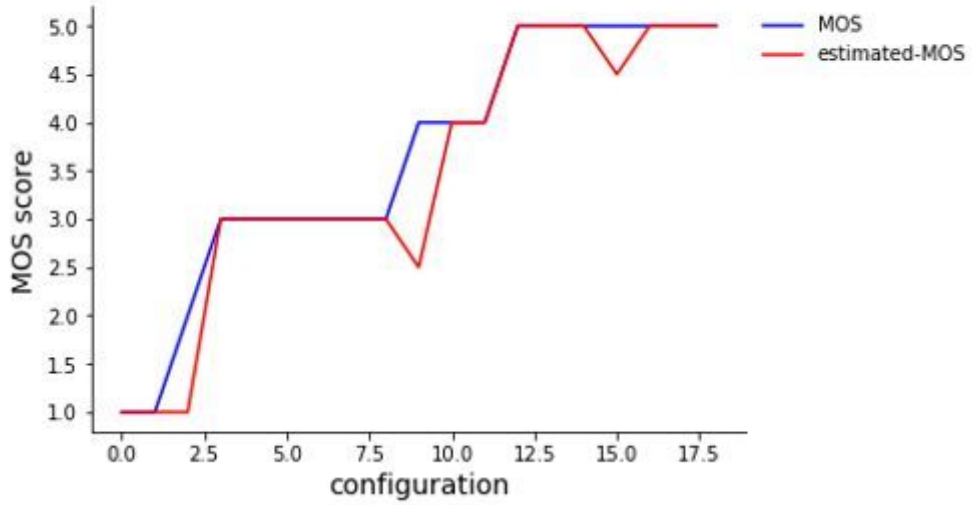

Figure 10

Comparison between estimated MOS and subjective MOS.

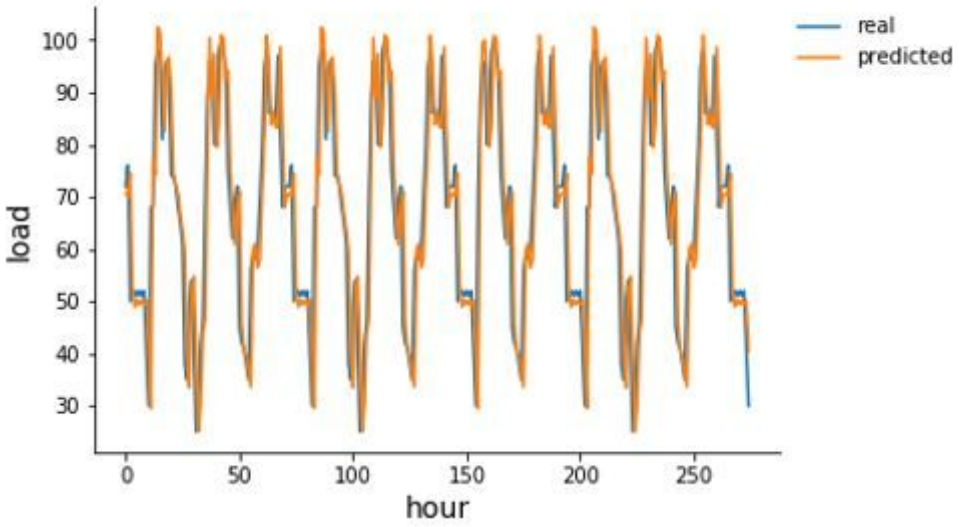

Figure 11

Predicted VS real values 11 days

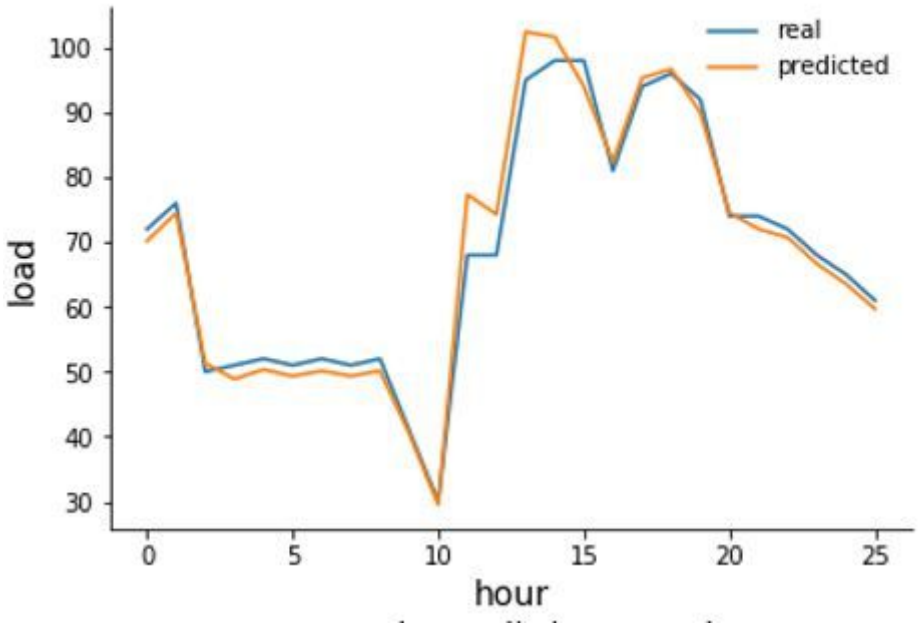

Figure 12

1 day prediction VS real